ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Khánh Tùng

NGHIÊN CỨU PHÁT TRIỂN KỸ THUẬT HỌC MÁY PHÁT HIỆN VÀ PHÂN LỚP TRÔI KHÁI NIỆM

Ngành đào tạo : Hệ thống thông tin
Mã số : 9480104

TÓM TẮT LUẬN ÁN TIẾN SĨ HỆ THỐNG THÔNG TIN

HÀ NỘI - 2025

Công trình được hoàn thành tại: Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

Người hướng dẫn khoa học: PGS.TS Hà Quang Thụy và PGS.TS Phan Xuân Hiếu

Phản biện: PGS.TS. Nguyễn Đức Dũng - Viện CNTT, Viện Hàn lâm KH&CN

Phản biện: PGS.TS Bùi Thu Lâm - HV Kỹ thuật mật mã, Ban Cơ yếu chính phủ

Phản biện: PGS.TS. Lê Hồng Phương - Trường ĐH Khoa học Tự nhiên, ĐHQGHN

Luận án sẽ được bảo vệ trước Hội đồng cấp Đại học Quốc gia chấm luận án tiến sĩ họp tại .....

vào hồi giờ ngày tháng năm

Có thể tìm hiểu luận án tại:

- Thư viện Quốc gia Việt Nam
- Trung tâm Thông tin - Thư viện, Đại học Quốc gia Hà Nội

Mở đầu

Trong một thế giới thực luôn động, các khái niệm thường không ổn định mà thay đổi theo thời gian nhưng hầu hết các mô hình học máy được xây dựng dựa trên các tập dữ liệu quá khứ đều tĩnh. Vì vậy, việc triển khai trực tuyến ngày càng nhiều các mô hình học máy tĩnh đã làm gia tăng tính cấp thiết cho việc phát triển các cơ chế hiệu quả cao và hiệu suất cao để giải quyết vấn đề học máy đối với phân phối dữ liệu không tĩnh, đặc biệt trong bối cảnh trực tuyến. Đồng thời, các phương pháp học máy trực tuyến phải đối phó được với các thách thức như dung lượng bộ nhớ hạn chế, các yêu cầu thời gian thực và trôi khái niệm (concept drift). Hơn nữa, các phương pháp học máy trực tuyến phải cung cấp các dự đoán có chất lượng cao, ổn định khi có nhiễu đầu vào và khả năng diễn giải tốt để có thể sử dụng đáng tin cậy trong thực tế. Trôi khái niệm ([43], [59], [19], [49], [28], [36]) mô tả những thay đổi không lường trước được phân phối dữ liệu được ghi nhận theo thời gian và các mô hình học máy cần có khả năng tự động thích nghi được với những thay đổi theo thời gian như vậy. Trôi khái niệm là một vấn đề nổi bật trong khai phá dữ liệu, đề cập tới "hiện tượng mà các tính chất thống kê của biến mục tiêu thay đổi theo thời gian một cách tuỳ ý không biết trước" [28] hoặc "mối quan hệ giữa dữ liệu đầu vào và biến mục tiêu có sự thay đổi theo thời gian" [36]. Các phương pháp tiếp cận khác nhau để phát hiện và xử lý trôi khái niệm đã được đề xuất và nhiều phương pháp trong số đó đã minh chứng được tiềm năng của chúng trong nhiều lĩnh vực ứng dụng, chẳng hạn như, phát hiện gian lận, kiểm soát hệ thống thích nghi, mô hình hóa người dùng, truy xuất thông tin, khai phá văn bản, y sinh học, chăm sóc sức khỏe, giáo dục, tài chính hoặc quảng cáo ([43], [59], [19], [49])

Nghiên cứu về trôi khái niệm liên quan đến việc phát triển các phương pháp luận và kỹ thuật giải quyết ba bài toán trôi khái niệm chính là Phát hiện trôi khái niệm, Hiểu trôi khái niệm và Thích nghi với trôi khái niệm. Tồn tại ba nhóm phương pháp và thuật toán phát hiện trôi khái niệm là: dựa trên tỷ lệ lỗi, dựa trên phân phối dữ liệu và dựa trên kiểm thử đa giả thuyết. Khung tổng quát Phát hiện trôi khái niệm bao gồm bốn giai đoạn. Tìm kiếm dữ liệu (Data Retrieval) nhằm thu được các đoạn dữ liệu từ các luồng dữ liệu (data streams). Mô hình hóa dữ liệu (Data Modeling) nhằm bao gói các đoạn dữ liệu thu được và trích chọn thành các đặc trưng chứa đựng các thông tin có tính nhận dạng tốt, hay nói cách khác là bước này sẽ chọn ra các đặc trưng có ý nghĩa trong việc đánh giá hệ thống khi nó thay đổi. Tính toán kiểm thử thống kê (Test Statistics Calculation) là bước tính toán khoảng cách để xác định có sự thay đổi hay không. Bước này là bước khó khăn nhất trong tất cả các bước. Kiểm thử giả thuyết (Hypothesis Test) tiến hành một kiểu kiểm thử giả thuyết để kiểm thử một giả thuyết cụ thể để đánh giá tính thống kê các thay đổi được xác định ở bước Tính toán kiểm thử thống kê. Lưu ý rằng, phát hiện trôi khái niệm là một quá trình phản ứng bị động mà không phải là quá trình phòng ngừa chủ động [36].

Ba bài toán hiểu trôi khái niệm là Phát hiện điểm trôi (Change Point Detection) xác định thời điểm xảy ra trôi khái niệm, Mức nghiêm trọng của trôi khái niệm và Vùng trôi khái niệm, trong đó, phát hiện điểm trôi là bài toán được quan tâm nhiều nhất. Hiểu biết về trôi khái niệm cần dự báo được thời điểm bắt đầu, giai đoạn thay đổi và thời điểm kết thúc của trôi khái niệm. Mức nghiêm trọng của trôi khái niệm đề cập đến việc sử dụng giá trị định lượng để đo mức độ tương đồng giữa khái niệm mới và khái niệm trước đó. Vùng trôi của khái niệm trôi là vùng xung đột giữa khái niệm mới và khái niệm trước đó, được định vị bằng cách tìm vùng trong không gian đặc trưng dữ liệu mà tại đó các phân phối dữ liệu có khác biệt đáng kể. Các kỹ thuật xác định vùng trôi phụ thuộc rất nhiều vào mô hình dữ liệu được sử dụng trong các phương pháp/thuật toán phát hiện trôi khái niệm. Thích nghi trôi đề cập tới việc cập nhật các mô hình học máy hiện có để phù hợp với trôi khái

niệm. Ba tiếp cận chính cho thích nghi trôi là Huấn luyện mô hình mới cho trôi toàn cục, Mô hình kết hợp cho trôi tái biễn và Điều chỉnh các mô hình hiện có theo trôi theo vùng.

Các xu hướng nghiên cứu điển hình để giải quyết các thách thức đối với xử lý trôi khái niệm là ([28], ([28], [2], [9], [29]): (1) Cung cấp thông tin về mức nghiêm trọng, các vùng trôi cùng với điểm trôi; (2) Học máy không giám sát và bán giám sát để phát hiện nhanh trôi, hiểu và điều chỉnh trôi khái niệm, đặc biệt trong xử lý trực tuyến; (3) Độ đo, khung và phương pháp đánh giá hiệu năng thuật toán học xử lý trôi khái niệm; (4) Học kết hợp các kỹ thuật xử lý trôi khái niệm; (5) Phát hiện trôi trên luồng dữ liệu nhiều chiều, đa dạng đặc trưng (số, phân lớp, đa phân lớp, thời gian, nhị phân và độ lệch) và dung lượng lớn; (6) Mô hình kết hợp phát hiện trôi khái niệm đa kiểu (đột ngột, dần dần, gia tăng); (7) Trôi khái niệm trong khai phá quy trình.

Định hướng tới các chủ đề nghiên cứu quan trọng theo xu hướng xử lý trôi khái niệm như đã được đề cập, **nghiên cứu của luận án** hướng tới **ba câu hỏi nghiên cứu với mục tiêu nghiên cứu** sau đây.

- *Câu hỏi nghiên cứu đầu tiên* là về khung và mô hình học kết hợp xử lý trôi khái niệm cũng như cơ chế quyết định trong học kết hợp, tương ứng với xu hướng nghiên cứu về Tổng hợp các kỹ thuật xử lý trôi khái niệm (Xu hướng 4). Mục tiêu nghiên cứu tương ứng là đề xuất các mô hình phát hiện trôi khái niệm dựa trên học tổng hợp và tối ưu hóa siêu tham số mô hình.
- *Câu hỏi nghiên cứu thứ hai* là về cách thức sử dụng mô hình cửa sổ linh hoạt để phân lớp trôi khái niệm đa kiểu, tương ứng với xu hướng nghiên cứu về mô hình kết hợp phát hiện trôi khái niệm đa kiểu (đột ngột, gia tăng, dần) (Xu hướng 6). Mục tiêu nghiên cứu tương ứng là đề xuất các mô hình phân lớp trôi khái niệm dựa trên mạng nơron nguyên mẫu (Prototypical Network) xử lý cửa sổ phát hiện trôi khái niệm đa kiểu.
- *Câu hỏi nghiên cứu thứ ba* là về nâng cao khả năng phân biệt của mô hình, phù hợp với xu hướng Học máy không giám sát và bán giám sát để phát hiện nhanh trôi, đặc biệt trong môi trường trực tuyến với số lượng mẫu có nhãn rất ít (Xu hướng 2). Mục tiêu nghiên cứu là đề xuất mô hình phân lớp trôi khái niệm dựa trên phân tích và tối ưu không gian biểu diễn của mạng nguyên mẫu.

Đối tượng nghiên cứu của luận án là các mô hình xử lý trôi khái niệm và các kỹ thuật được áp dụng trong các mô hình đó.

Phạm vi nghiên cứu của luận án tập trung tới các kỹ thuật và giải pháp cải tiến được áp dụng trong các mô hình xử lý trôi khái niệm dựa trên học máy kết hợp và học xử lý trôi khái niệm đa kiểu.

Luận án sử dụng **phương pháp nghiên cứu kết hợp** phương pháp nghiên cứu định tính và phương pháp nghiên cứu định lượng.

Tham gia vào dòng nghiên cứu trên thế giới theo các xu hướng xử lý trôi khái niệm như đã được đề cập, luận án có **ba nhóm đóng góp** chính sau đây:

- Luận án đề xuất một mô hình phát hiện trôi khái niệm gồm hai pha nối tiếp nhau, là pha tối ưu hóa siêu tham số mô hình và pha học kết hợp. Trong pha học kết hợp xây dựng hai mô hình: E-ERICS và ERICS+3. Mô hình E-ERICS (Ensemble-ERICS) được thiết kế bằng cách kết hợp bốn mô hình cơ sở tốt nhất sau pha tối ưu hóa siêu tham số. Mô hình sử dụng cơ chế bỏ phiếu để đưa ra quyết định về việc một mẫu dữ liệu trong luồng có phải là điểm trôi khái niệm đột ngột hoặc trôi gia tăng hay không. Trong khi đó, mô hình ERICS+3 được xây dựng bằng cách kết hợp ba mô hình cơ sở tốt nhất nhằm phát hiện các điểm trôi dần dần trên từng trường hợp dữ liệu trong luồng. Các kết quả nghiên cứu tương ứng với hai mô hình này đã được công bố trong các công trình [TungNK1] đối với mô hình E-ERICS và [TungNK2] đối với mô hình ERICS+3.

- Luận án đề xuất hai mô hình cải tiến cho nhiệm vụ phân lớp trôi khái niệm, bao gồm VAR-WIND và MetaLDD-Finetune. Cả hai mô hình đều được phát triển dựa trên mô hình phân lớp trôi Meta-ADD [55], đồng thời kế thừa một số thành phần từ mô hình Type-LDD [56] – phiên bản mở rộng của Meta-ADD. Mô hình VAR-WIND cải tiến giai đoạn trích xuất đặc trưng của Meta-ADD bằng cách áp dụng các chiến lược cửa sổ khác nhau phù hợp với từng kiểu trôi: sử dụng cửa sổ lật cho trôi đột ngột, cửa sổ trượt cho trôi dần và cửa sổ mở rộng cho trôi tăng dần. Mỗi chiến lược được lựa chọn dựa trên đặc điểm của từng kiểu trôi, từ đó giúp mô hình học được các biểu diễn đặc trưng phù hợp hơn cho từng loại trôi khái niệm. Mô hình MetaLDD-Finetune được xây dựng bằng cách bổ sung một bước tinh chỉnh (fine-tuning) sau giai đoạn tiền huấn luyện, nhằm cải thiện khả năng thích ứng của bộ phân lớp với phân phối dữ liệu mới. Bước tinh chỉnh này giúp tối ưu hóa hiệu quả phân lớp cho các kiểu trôi khái niệm trong luồng dữ liệu. Các kết quả nghiên cứu liên quan đến hai mô hình đề xuất đã được công bố trong các công trình: [TungNK5] cho VAR-WIND và [TungNK3] cho MetaLDD-Finetune.
- Từ nền tảng của mô hình MetaLDD-Finetune, luận án phát triển một mô hình phân lớp trôi khái niệm mới có tên là CS&AM_SC (Cosine Similarity and Angular Margin based Softmax Classifier), nhằm nâng cao độ chính xác trong phân loại kiểu trôi thông qua việc phân tích và tối ưu hóa không gian biểu diễn đặc trưng trong mạng nguyên mẫu. Cụ thể, mô hình CS&AM_SC sử dụng ba kỹ thuật chính: (i) thay thế phép nhân vô hướng bằng độ tương đồng Cosine trong bộ phân lớp Softmax, (ii) tích hợp biên góc (Angular Margin) để tăng cường khả năng phân tách giữa các lớp, và (iii) sử dụng hàm mất mát trung tâm (Center Loss) nhằm giảm biến thiên nội lớp. Sự kết hợp các kỹ thuật này giúp tối ưu hóa không gian biểu diễn theo hướng vừa đảm bảo tính gắn kết trong lớp, vừa tăng cường khả năng phân tách giữa các lớp, từ đó cải thiện hiệu quả phân lớp kiểu trôi. Kết quả nghiên cứu thuộc nhóm đóng góp này của luận án đã được công bố trong [TungNK4].

Bố cục của luận án gồm Mở đầu và bốn chương nội dung, Kết luận và Danh mục các tài liệu tham khảo.

Chương 1. Trôi khái niệm, xử lý trôi khái niệm và liên hệ với luận án

1.1 Trôi khái niệm

1.1.1 Định nghĩa về trôi khái niệm

Cho một khoảng thời gian $[0, t]$, một tập các mẫu, $S_{0,t} = d_0, \dots, d_t$ trong đó $d_i = (X_i, y_i)$ là một quan sát (hoặc một thể hiện) với X_i là vector đặc trưng, y_i là nhãn, và $S_{0,t}$ tuân theo một phân phối định rõ $F_{0,t}(X, y)$.

Định nghĩa 1.3 (Trôi khái niệm) [28]

Trôi khái niệm xuất hiện tại thời điểm $t + 1$ nếu:

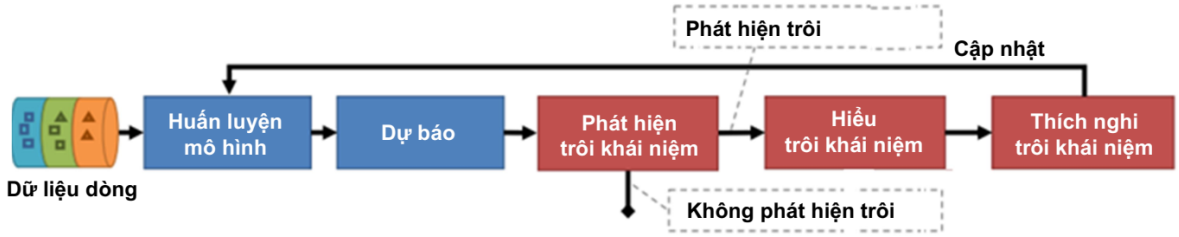
$$F_{0,t}(X, y) \neq F_{0,t+1}(X, y), \text{ được ký hiệu là } \exists t: P_t(X, y) \neq P_{t+1}(X, y). \quad (1.4)$$

1.1.2 Các kiểu trôi khái niệm

Trôi khái niệm được phân thành bốn kiểu là đột ngột (*sudden*), dần dần (*gradual*), gia tăng (*incremental*), tái diễn (*reoccuring*).

1.1.3 Xử lý trôi khái niệm

Một khung quy trình chung xử lý trôi khái niệm được minh họa tại Hình 1.4.



Hình 1.4. Khung xử lý trôi khái niệm [28]

1.2 Phát hiện trôi khái niệm

Khung tổng quát phát hiện trôi khái niệm bao gồm bốn giai đoạn là Truy hồi dữ liệu, Mô hình hóa, Tính toàn kiểm thử thống kê, Kiểm thử giả thuyết.

1.2.1 Dựa trên tỉ lệ lỗi

Trong phương pháp này, các đại lượng thống kê liên quan đến tỷ lệ lỗi của bộ phân lớp được khai thác để xác định sự xuất hiện của trôi khái niệm. Cụ thể, khi trôi khái niệm xảy ra, tỷ lệ lỗi của bộ phân lớp có xu hướng tăng đột ngột hoặc biến đổi dần theo thời gian.

1.2.2 Dựa trên phân phối dữ liệu

Nguyên lý cốt lõi của phương pháp này là: khi trôi khái niệm xảy ra, phân phối của các điểm dữ liệu có xu hướng thay đổi đáng kể. Do đó, bằng cách so sánh phân phối của dữ liệu hiện tại với phân phối trong quá khứ, ta có thể phát hiện những khác biệt có ý nghĩa thống kê, từ đó xác định sự xuất hiện của trôi khái niệm [59].

1.2.3 Kiểm thử đa giả thuyết

Phát hiện trôi khái niệm dựa trên kiểm định đa giả thuyết cũng sử dụng các kỹ thuật phát hiện trôi giống như hai phương pháp đã đề cập ở trên. Sự khác biệt đó là phương pháp này thực hiện đồng thời nhiều phép kiểm định thống kê trên các cửa sổ dữ liệu nhằm phát hiện trôi khái niệm

1.2.4 Học kết hợp phát hiện trôi khái niệm

Bên cạnh ba hướng nghiên cứu truyền thống đã nêu ở trên, học kết hợp (ensemble learning) là một hướng tiếp cận triển vọng trong phát hiện trôi khái niệm. Đây là một kỹ thuật mạnh trong học máy, cho phép kết hợp nhiều mô hình nhằm cải thiện hiệu năng dự đoán, tăng tính ổn định và khả năng tổng quát hóa [Du14].

1.3 Hiểu trôi khái niệm

1.3.1 Thời gian trôi khái niệm

Thời gian trôi bao gồm các mốc thời gian như thời điểm bắt đầu trôi (điểm trôi), quãng thời gian thay đổi khái niệm, thời điểm kết thúc trôi.

1.3.2 Mức độ nghiêm trọng của trôi khái niệm

Mức độ nghiêm trọng của trôi khái niệm, được định nghĩa là một đại lượng để đo lường mức độ khác biệt giữa khái niệm mới và khái niệm trước đó [19].

1.3.3 Vùng trôi khái niệm

Vùng trôi trong hiện tượng trôi khái niệm được hiểu là khu vực trong không gian đặc trưng dữ liệu nơi tồn tại sự xung đột giữa khái niệm hiện tại và khái niệm trước đó [Webb16].

1.3.4 Phân lớp trôi khái niệm

Phân lớp trôi khái niệm đóng vai trò thiết yếu trong việc hỗ trợ tối ưu hóa các chiến lược thích ứng, quản lý hiệu quả tài nguyên tính toán, và nâng cao hiệu suất giám sát của mô hình học máy [19]. Việc nhận

diện đúng kiểu trôi (đột ngột, dần dần, gia tăng...) cho phép hệ thống lựa chọn phương án xử lý phù hợp, từ đó cải thiện độ chính xác và độ ổn định của mô hình theo thời gian.

Có hai phương pháp chính để thực hiện phân lớp trôi khái niệm, đó là:

(i) Dựa trên phân phối dữ liệu và

(ii) Dựa trên học cách học (Meta learning).

Các nghiên cứu của phương pháp phân lớp kiểu trôi *dựa trên phân phối dữ liệu* có đặc điểm chung là định lượng quãng thời gian thay đổi khái niệm để xác định kiểu trôi.

Phương pháp học cách học có thể học cách đánh giá độ tương đồng giữa các mẫu dữ liệu để áp dụng cho nhiều tác vụ phân lớp khác nhau mà không cần phải huấn luyện lại hoàn toàn. Như vậy có thể áp dụng phương pháp học cách học để phân lớp các kiểu trôi dựa trên việc đánh giá độ tương đồng giữa các mẫu dữ liệu trôi mới với các kiểu đã học trong quá khứ.

1.4 Thích nghi trôi khái niệm

Vấn đề thích ứng với trôi khái niệm được giải quyết theo ba phương pháp phổ biến đó là: (1) Huấn luyện lại mô hình; (2) Điều chỉnh mô hình và (3) Sử dụng tập hợp các mô hình.

1.5 Một số bộ dữ liệu phổ biến

1.5.1 Năm bộ dữ liệu tổng hợp

Năm bộ dữ liệu SEA, Agrawal, và Hyperplane, Stagger, Sine

1.5.2 Bốn bộ dữ liệu thực tế

Spambase, Adult, KDD 1999, Dota2.

1.6 Đánh giá hiệu năng phát hiện trôi khái niệm

Độ chính xác, điểm F1, độ trễ phát hiện.

1.7 Xu hướng nghiên cứu xử lý trôi khái niệm

- Cung cấp thông tin về mức nghiêm trọng, các vùng trôi cùng với điểm trôi.
- Học máy không giám sát và bán giám sát để phát hiện nhanh trôi, hiểu và điều chỉnh trôi khái niệm, đặc biệt trong xử lý trực tuyến.
- Độ đo, khung và phương pháp đánh giá hiệu năng thuật toán học xử lý trôi khái niệm.
- Tích hợp các kỹ thuật xử lý trôi khái niệm.
- Dữ liệu nhiều chiều, đa dạng đặc trưng (số, phân lớp, đa phân lớp, thời gian, nhị phân và độ lệch) và dung lượng lớn.
- Phát hiện trôi khái niệm không đột ngột và kết hợp phát hiện trôi khái niệm đa kiểu (đột ngột, gia tăng, dần).
- Trôi khái niệm trong khai phá quy trình.

1.8 Liên hệ với nghiên cứu trong luận án

1.8.1 Một số luận án Tiến sỹ liên quan trên thế giới

Mục này giới thiệu một số luận án Tiến sỹ trên thế giới về phát hiện và xử lý trôi khái niệm.

1.8.2. Liên hệ với nghiên cứu của luận án

Mục này chỉ ra mối liên hệ các nghiên cứu của luận án này tới các triển vọng nghiên cứu về phát hiện và phân loại trôi khái niệm

Kết luận Chương 1

Tổng hợp nội dung khảo sát trong chương 1

Chương 2. Mô hình học tổng hợp phân phối tham số phát hiện trôi khái niệm

2.1 Mô hình phát hiện trôi khái niệm ERICS

Haug và cộng sự đề xuất một mô hình phát hiện trôi khái niệm mới, **ERICS** (Effective and Robust Identification of Concept Shift), bằng cách giám sát sự thay đổi trong phân phối tham số của mô hình học dự đoán trong luồng dữ liệu.

Thay vì phân tích trực tiếp dữ liệu, ERICS xem các tham số của mô hình θ là biến ngẫu nhiên với phân phối xác suất $P(\theta; \psi)$. Nếu có trôi khái niệm thực, tức là thay đổi trong phân phối điều kiện $P(Y|X)$, thì sẽ xảy ra sự thay đổi trong phân phối các tham số mô hình → có thể phát hiện bằng cách đo:

- Entropy vi phân (differential entropy)
- Độ lệch KL (KL-divergence) giữa các phân phối tham số tại các thời điểm khác nhau.

Hai đặc tính chính của ERICS:

1. Phát hiện trôi gắn liền với sự thay đổi trong tham số mô hình → tránh báo động giả do nhiễu.
2. Khả năng giải thích được có thể truy vết trôi đến từng tham số → từ đó đến từng đặc trưng đầu vào).

Cơ chế phát hiện trôi:

- Tính trung bình trượt (moving average) của sự thay đổi entropy và KL-divergence qua các bước thời gian.
- Áp dụng ngưỡng động (adaptive threshold) để phát hiện trôi:
    - Cập nhật ngưỡng α theo thời gian với tốc độ β (siêu tham số điều chỉnh độ nhạy).
    - Nếu độ thay đổi vượt ngưỡng → phát hiện có trôi.

2.2 Nhận xét và ý tưởng cải tiến mô hình ERICS

Nhược điểm của mô hình ERICS là nó phát hiện nhiều điểm trôi sai. Độ chính xác của ERICS phụ thuộc nhiều vào lựa chọn bộ siêu tham số (M, W, β) do vậy việc lựa chọn bộ tham số phù hợp vẫn là câu hỏi mở. Do vậy:

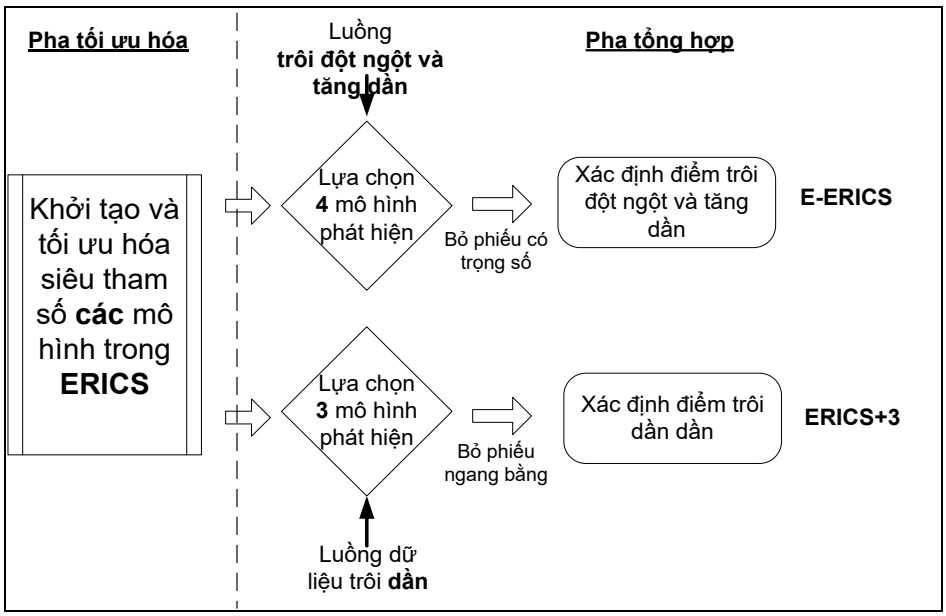
- Thứ nhất, áp dụng phương pháp tối ưu hóa siêu tham số dựa trên kỹ thuật Tối ưu Bayes, nhằm tìm ra bộ tham số tối ưu giúp cải thiện hiệu năng phát hiện trôi của ERICS.
- Thứ hai, áp dụng ý tưởng từ phương pháp học kết hợp trong để kết hợp các mô hình ERICS (đã tối ưu siêu tham số). Kết quả phát hiện điểm trôi từ từng mô hình đơn lẻ này sẽ được tổng hợp bằng một cơ chế bỏ phiếu và đưa ra kết quả cuối cùng đảm bảo tin cậy và giảm thiểu ảnh hưởng của các quyết định sai lệch so với từ một mô hình đơn lẻ.

2.3 Hai mô hình đề xuất E-ERICS và ERICS+3

2.3.1 Mô hình cải tiến đề xuất

Phần này đề xuất mô hình gồm hai pha để cải tiến ERICS, trong đó đề xuất mô hình E-ERICS để phát hiện trôi đột ngột và trôi gia tăng; đề xuất mô hình ERIC+3 để phát hiện trôi dần dần. Mô hình hai pha minh họa tại Hình 2.1 gồm:

- Pha tối ưu siêu tham số mô hình (gọi tắt là pha BO): áp dụng phương pháp tối ưu hóa Bayes để tìm kiếm các siêu tham số tối ưu cho mô hình ERICS. Quá trình này được thực hiện đồng nhất cho các kiểu trôi
- Pha học kết hợp: Đưa ra các phương án lựa chọn mô hình có hiệu năng tốt nhất với cơ chế bỏ phiếu khác nhau, được thiết kế riêng để phù hợp với đặc điểm của từng kiểu trôi khái niệm, giúp nâng cao độ chính xác trong việc phát hiện trôi.



Hình 2.1 Hai mô hình E-ERICS và ERICS+3 đề xuất

2.3.2 Pha tối ưu hóa siêu tham số

Tối ưu hóa siêu tham số đóng vai trò quan trọng trong việc cải thiện hiệu năng của các mô hình học máy. Luận án sử dụng tối ưu hóa Bayes do *phù hợp* cho việc tinh chỉnh siêu tham số trong học máy, nơi việc đánh giá một bộ siêu tham số có thể rất tốn kém.

2.3.3 Pha học kết hợp

Sau khi kết thúc pha tối ưu hóa Bayes, ta có nhiều mô hình đã được huấn luyện với các siêu tham số khác nhau. Pha học kết hợp được thực hiện để xác nhận lại điểm trôi khái niệm một cách chắc chắn hơn bằng cách kết hợp các mô hình tốt nhất.

2.3.3.1 Mô hình E-ERICS

Để xác định các điểm trôi đột ngột hoặc gia tăng một cách chính xác và ổn định hơn, luận án xây dựng cơ chế học kết hợp trên các mô hình đã được tối ưu siêu tham số. Sau pha BO, thu được n mô hình phát hiện trôi và biết được những mô hình nào có điểm F1 cao nhất (sắp xếp theo thứ tự từ cao đến thấp M_0, M_1, M_2, … M_n) trong đó M_0 là mô hình có điểm F1 cao nhất. Luận án chọn 4 mô hình (M_0, M_1, M_2, M_3) vì 4 mô hình cho điểm F1 cao nhất, trong đó M_0 có số phiếu gấp đôi các mô hình còn lại.

2.3.3.2 Mô hình ERICS+3

Đối với việc phát hiện trôi dần dần, sau pha BO, chọn ba mô hình gồm mô hình có điểm F1 cao nhất M_F mô hình có độ hồi tưởng cao nhất M_P, và mô hình có điểm chính xác cao nhất M_R. Sau đó, sử dụng cơ chế bỏ phiếu ngang bằng để xác định điểm trôi.

2.4 Thực nghiệm và đánh giá

2.4.1 Mục tiêu thực nghiệm

Đánh giá hiệu quả của mô hình hai pha trong việc phát hiện điểm trôi đột ngột, trôi gia tăng, trôi dần dần so sánh với mô hình ERICS gốc.

2.3.2 Thiết lập thực nghiệm

- Bộ dữ liệu: gồm các bộ tổng hợp SEA, Stagger, Sine, Agrawal và các bộ thực tế KDD, Spambase, Dota2, Adult.

- Cấu hình mô hình: Khởi tạo m=10 mô hình cơ sở với tập giá trị siêu tham số ngẫn nhiên nằm trong không gian tìm kiếm. Sau đó đề xuất 350 mô hình (đối với E-ERICS) và 250 mô hình (đối với ERICS+3). Kích thước các lô trong E-ERICS theo từng bộ dữ liệu; trong ERICS+3 là 100.
- Chiến lược huấn luyện và đánh giá: Sử dụng cách đánh giá xen kẽ việc kiểm tra với huấn luyện.

2.3.3 Các độ đo: P, R, điểm F1.

2.3.4 Kết quả và đánh giá

- Mô hình E-ERICS

Các độ đo	SEA			Hyperplane		
	ERICS	Pha BO	Pha tổng hợp	ERICS	Pha BO	Pha tổng hợp
R	0.5000	1.0000	**1.0000**	0.1701	**1.0000**	**1.0000**
P	0.3103	0.5306	**0.5778**	1.0000	**1.0000**	**1.0000**
Điểm F1	0.3830	0.6933	**0.7324**	0.1748	**1.0000**	**1.0000**

Bảng 2.3. So sánh hiệu năng trên bộ SEA và Hyperplane

Các độ đo	KDD			Spambase		
	ERICS	Pha BO	Pha tổng hợp	ERICS	Pha BO	Pha tổng hợp
R	0.2500	**1.0000**	**1.0000**	0.7500	0.7500	0.7500
P	0.2667	0.7333	**0.7917**	0.6061	**0.8611**	0.8571
Điểm F1	0.2581	0.8462	**0.8837**	0.6704	**0.8017**	0.8000

Bảng 2.4. So sánh hiệu năng trên bộ KDD và Spambase

Các độ đo	Dota2			Adult		
	ERICS	Pha BO	Pha tổng hợp	ERICS	Pha BO	Pha tổng hợp
R	0.2500	1.0000	**1.0000**	1.0000	1.0000	**1.0000**
P	0.0417	0.3913	**0.4000**	0.2212	0.3929	**0.3934**
Điểm F1	0.0714	0.5625	**0.5714**	0.3623	0.5641	**0.5647**

Bảng 2.5. So sánh hiệu năng trên bộ Dota2 và Adult

Các độ đo	Giá trị trung bình		
	ERICS	Pha BO	Pha tổng hợp
R	0.4867	**0.9583**	**0.9583**
P	0.4077	0.6515	**0.6700**
Điểm F1	0.3200	0.7446	**0.7509**

- **Mô hình ERICS+3**

Bảng 2.7. So sánh hiệu năng giữa các mô hình

Mô hình	Bộ dữ liệu	#TP1	#FP1	P	#TP2	#FN2	R	Điểm F1

ERICS	**SEA**	81	64	0.5586	596	54	0.9169	0.6943
ERICS+3		80	60	**0.5714**	596	54	0.9169	**0.7041**
ERICS	**Stagger**	38	30	0.5588	150	50	0.7500	0.6404
ERICS+3		28	24	0.5385	200	0	**1.0000**	**0.7000**
ERICS	**Sine**	50	47	0.5155	197	3	0.9850	0.6768
ERICS+3		41	38	0.5190	197	3	0.9850	0.6798
ERICS	**Agrawal**	87	92	0.4860	295	5	0.9833	0.6505
ERICS+3		74	63	**0.5401**	295	5	0.9833	**0.6973**

Bảng 2.8. So sánh trung bình hiệu năng giữa các mô hình trên các bộ dữ liệu

Độ đo	ERICS	ERICS+3
P trung bình	0.5297	**0.5423**
R trung bình	0.9088	**0.9713**
Điểm F1 trung bình	0.6655	**0.6953**

Kết quả phát hiện trôi đột ngột và gia tăng của E-ERICS

Kết quả cho thấy rằng, nhờ cải thiện các siêu tham số quan trọng (M, W, β), một số mô hình trong pha BO đã đạt được hiệu năng vượt trội so với ERICS trong tất cả các trường hợp. Mô hình học kết hợp tận dụng ưu điểm của các mô hình đơn lẻ trong việc loại bỏ một số điểm dương tính giả được tìm thấy trong pha BO, đồng thời bổ sung thêm một số điểm dương tính thực sự. Nhờ đó, mô hình tăng điểm F1 khoảng **4%** trên các bộ dữ liệu SEA và KDD, đồng thời đạt kết quả tốt hơn ở hầu hết các tiêu chí đánh giá so với chỉ sử dụng pha BO, đặc biệt là trong phát hiện trôi đột ngột.

Kết quả phát hiện trôi dần của ERICS+3

Kết quả thực nghiệm cho thấy rằng thay đổi trong phương pháp học kết hợp dẫn đến tăng trung bình khoảng 6,25% về độ hồi tưởng, 2,4% về điểm F1, và tăng nhẹ về độ chính xác. Trong một số trường hợp đặc biệt, ERICS+3 đạt hiệu năng vượt trội so với ERICS, cụ thể:

- Trên bộ dữ liệu Agrawal, điểm F1 tăng 4,4% và độ chính xác tăng hơn 5%.
- Trên bộ dữ liệu Stagger, điểm F1 tăng gần 6%, trong khi độ hồi tưởng tăng 25%

2.4 Kết luận Chương 2

Chương này giới thiệu mô hình phát hiện trôi khái niệm gồm hai pha là pha tối ưu siêu tham số và pha học kết hợp, trong đó đề xuất hai mô hình E-ERICS phát hiện trôi đột ngột/gia tăng và mô hình ERICS+3 phát hiện trôi dần dần, cho kết quả chính xác hơn với độ trễ thấp hơn so với mô hình ERICS. Một phần các kết quả nghiên cứu của chương này được công bố trong [TungNK1] (đã nhận được một tham chiếu Scopus từ các tác giả nước ngoài), [TungNK2].

Trong chương sau, luận án sẽ trình bày các đề xuất của luận án về các kỹ thuật học máy phân lớp trôi khái niệm trong dữ liệu luồng.

Chương 3. Phát triển mô hình phân lớp trôi khái niệm dựa trên mạng nguyên mẫu

3.1 Phân lớp trôi khái niệm dựa trên mạng nguyên mẫu

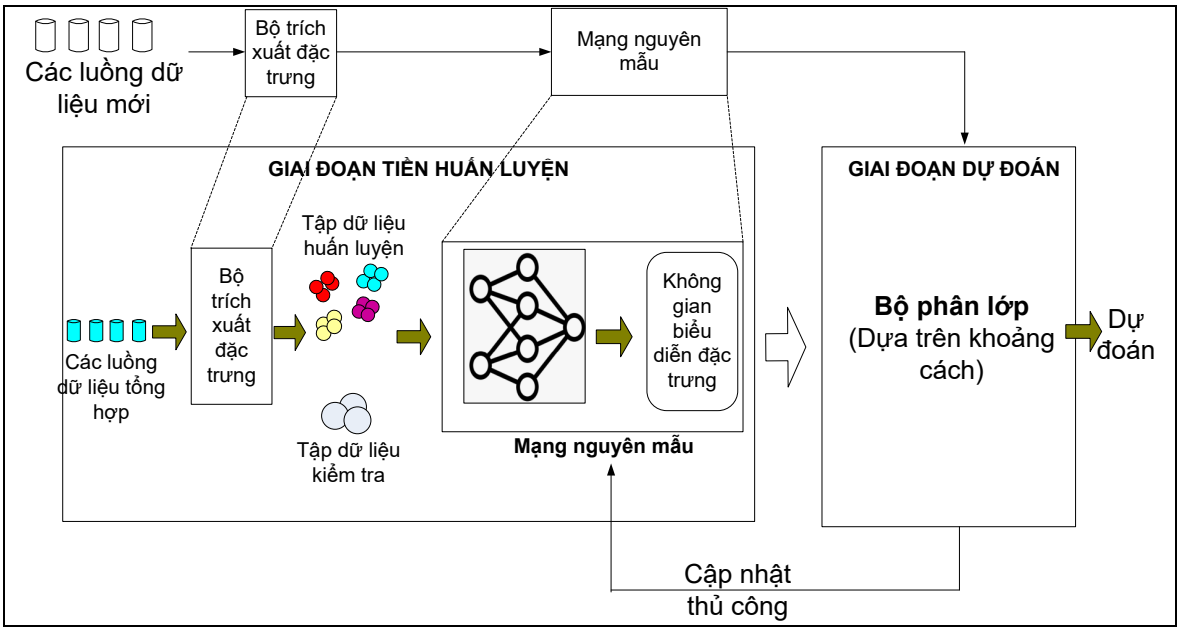
Để giải quyết bài toán phân lớp trôi, luận án tiếp cận theo hướng học ít mẫu thông qua mạng nguyên mẫu, như đã đề cập trong Mục 1.3.4. Phương pháp này cho phép mô hình phân biệt được các kiểu trôi dựa trên đặc trưng biểu hiện của từng loại, ngay cả khi số lượng mẫu cho mỗi loại trôi là rất hạn chế – điều thường gặp trong môi trường dữ liệu luồng thực tế.

Vì vậy Mục đầu tiên của chương này trình bày tiếp cận phân lớp trôi khái niệm dựa trên mạng nguyên mẫu. Hai mục tiếp theo trình bày về đề xuất của luận án về hai mô hình phân lớp trôi khái niệm VAR-WIND và MetaLDD-Finetune, kế thừa phương pháp từ Meta-ADD.3.1.1 Mạng nguyên mẫu học ít mẫu.

Trong bối cảnh phân lớp trôi khái niệm với số lượng mẫu có nhãn hạn chế, mạng nguyên mẫu là một phương pháp đơn giản nhưng hiệu quả, đặc biệt phù hợp với bài toán học từ ít mẫu. Quá trình huấn luyện mạng nguyên mẫu thường tuân theo chiến lược: huấn luyện theo từng nhiệm vụ.

3.1.2 Mô hình phân lớp trôi khái niệm Meta-ADD

Meta-ADD đề xuất cách thức phát hiện và phân lớp trôi khái niệm trong luồng dữ liệu như minh họa trong Hình 3.2.



Hình 3.2. Quy trình phân lớp trôi chi tiết của Meta-ADD [Yu22]

- Trong giai đoạn tiền huấn luyện, thực hiện huấn luyện mạng so khớp theo phương pháp huấn luyện theo tập nhiệm vụ.
- Trong giai đoạn dự đoán kiểu trôi của một luồng dữ liệu, thực hiện dự đoán kiểu trôi và cập nhật mạng nguyên mẫu theo cách thủ công.

Đặc trưng mà Meta-ADD trích xuất là khoảng cách giữa tỉ lệ lỗi trung bình giữa các cửa sổ. Một cửa sổ W_i với độ dài n (mỗi W_i chứa n mẫu) có tỷ lệ lỗi trung bình $\hat{e}_t$:

$$\hat{e}_t = \frac{\sum_{t=1}^{n} e_t}{n} \tag{3.1}$$

Khoảng cách giữa tỷ lệ lỗi trung bình của hai cửa sổ là:

$$gap_i = \hat{e}_{t+1} - \hat{e}_t \tag{3.6}$$

Giả sử mỗi luồng dữ liệu có l cửa sổ thì số gap là $l - 1$. Do vậy một mẫu huấn luyện i là:

$$G_i: \{\hat{X} \times \hat{y}\} \in R^l \quad (3.7)$$

Trong đó: $\hat{X} = \{gap_1, \dots, gap_{l-1}\}$ và $\hat{y} \in \{1, .., k.., 4\}$ là nhãn tương ứng 4 loại trôi. Nếu có N luồng, mỗi luồng có l cửa sổ, mỗi cửa sổ có n mẫu, vậy tổng mỗi luồng có $m = l * n$ mẫu dữ liệu thì ta có tập dữ liệu $g: \{X \times y\}^{N \times m}$ sẽ được ánh xạ vào tập đặc trưng là $G: \hat{X} \times \hat{y} \in R^{N \times l}$. Có thể thấy với đặc trưng G này sẽ độc lập với dữ liệu thực tế của luồng sinh ra.

Các vector đặc trưng được chia làm 2 tập: tập dữ liệu hỗ trợ và tập dữ liệu truy vấn. Các tập dữ liệu này sau đó được sử dụng để để huấn luyện mạng nguyên mẫu theo phương pháp **huấn luyện theo tập nhiệm vụ**, như sau:

- Bước 1 - Lấy dữ liệu: Trong mỗi vòng huấn luyện, một số ít mẫu dữ liệu được chọn làm tập huấn luyện, và một số lượng lớn hơn được chọn làm tập kiểm tra cho từng kiểu trôi khái niệm.
- Bước 2 – Ánh xạ sang không gian nhúng: sử dụng một mạng nơ-ron ánh xạ cả tập huấn luyện và tập kiểm tra vào không gian nhúng thông qua hàm f_ϕ:

  $$f_\phi: R^N \to R^M \quad (3.8)$$

  với ϕ là tham số được học, ánh xạ $\hat{X}_i$ vào không gian nhúng $f_\phi(\hat{X}_i)$,
- Bước 3 – Tính toán các vector tâm cụm của lớp c_k bằng cách lấy trung bình các vector nhúng của *tập huấn luyện* cho từng kiểu trôi khái niệm:

  $$c_k = \frac{1}{|S_k|} \sum_{(X_i, y_i) \in R^{N \times l}} f_\phi(\hat{X}_i) \quad (3.9)$$

  S_k là tập dữ liệu luồng với nhãn k
- Bước 4 – Phân lớp: các mẫu $\hat{x}$ trong tập kiểm tra được phân lớp bằng cách tính toán khoảng cách (sử dụng một hàm đo khoảng cách độ tương đồng cosine) từ mẫu $\hat{x}$ tới tâm cụm của từng lớp trong không gian nhúng. Mẫu $\hat{x}$ sẽ thuộc về lớp có khoảng cách gần nhất.

  $$p_\theta(\hat{y} = k|\hat{x}) = \frac{\exp\left(-d\left(f_\phi(\hat{x}), c_k\right)\right)}{sum_{k'} \exp\left(-d\left(f_\phi(\hat{x}), c_k\right)\right)} \quad (3.10)$$

- Bước 5 – Tính toán hàm mất mát: tính bằng xác suất log âm (negative log-likelihood) của nhãn thực sự của các mẫu trong tập kiểm tra, dựa trên xác suất của lớp được dự đoán theo khoảng cách đến các tâm cụm của lớp.

  $$J(\phi) = -\log p_\phi(y = k|\hat{x}) \quad (3.11)$$

- Bước 6 - Tối ưu hóa: Các tham số của mạng nhúng f_ϕ được cập nhật bằng thuật toán hạ gradient ngẫu nhiên (SGD) để giảm thiểu lỗi phân lớp.

Sau giai đoạn tiền huấn luyện là giai đoạn dự đoán kiểu trôi trên các luồng dữ liệu khác nhau. Vì đặc điểm về trôi khái niệm đối với mỗi loại luồng cũng khác nhau, vì vậy cần cập nhật mạng nguyên mẫu để thích nghi với các đặc điểm trôi của dạng luồng cụ thể

3.1.3 Nhận xét về Meta-ADD

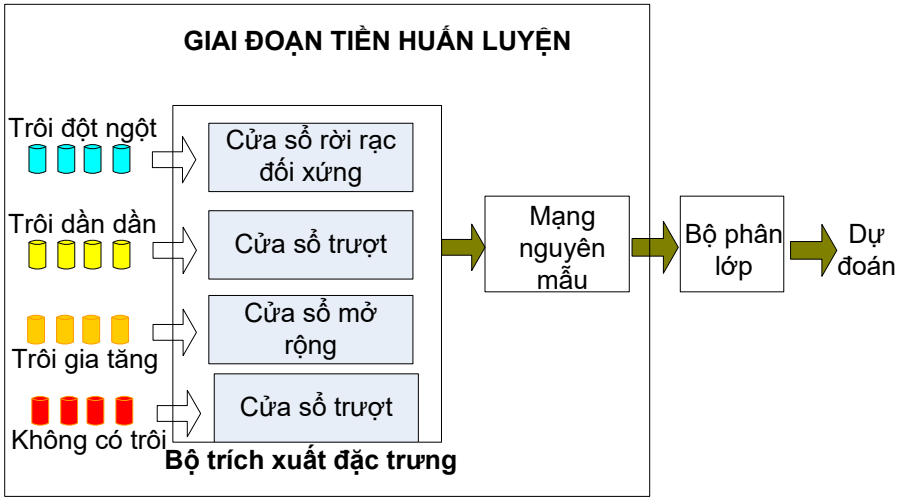
- Nhận xét 1: Trong mô hình Meta-ADD, ở bộ trích xuất đặc trưng, tác giả áp dụng duy nhất chiến lược cửa sổ rời rạc. Tuy nhiên mỗi kiểu trôi có đặc trưng khác nhau nên nếu chỉ sử dụng một chiến lược cửa sổ rời rạc sẽ không trích xuất được các đặc trưng điển hình của từng kiểu. Do vậy đối với mỗi kiểu trôi cần áp dụng một chiến lược cửa sổ phù hợp. Từ nhận xét này, luận án đề xuất mô hình VAR-WIND trong đó bộ trích xuất đặc trưng sẽ áp dụng các chiến lược của sổ linh hoạt đối với mỗi kiểu trôi, được trình bày chi tiết trong Phần 3.2.

- Nhận xét 2: Trong giai đoạn tiền huấn luyện, mạng nguyên mẫu được huấn luyện trên một bộ dữ liệu tổng hợp. Trong giai đoạn dự đoán cần xác định kiểu trôi trên các mẫu dữ liệu mới, chưa từng thấy trước đây. Để đảm bảo tính tổng quát, mô hình này đưa ra phương án cập nhật mạng nguyên mẫu theo cách thủ công. Điều này rất tốn kém. Do vậy luận án đề xuất phương án khác, đó là thêm giai đoạn tinh chỉnh sau giai đoạn tiền huấn luyện, để có mô hình cải tiến MetaLDD-Finetune với kỹ thuật huấn luyện đảm bảo tính tổng quát hóa trên các luồng dữ liệu mới, được trình bày chi tiết trong Phần 3.3.

3.2. Mô hình phân lớp trôi khái niệm VAR-WIND đề xuất

3.2.1 Mô hình VAR-WIND đề xuất

Trong giai đoạn tiền huấn luyện Meta-ADD, ở bước trích xuất đặc trưng, chỉ sử dụng duy nhất một chiến lược cửa sổ rời rạc áp dụng trên luồng dữ liệu để trích xuất các đặc trưng của tất cả các kiểu trôi, tuy nhiên mỗi kiểu trôi có đặc điểm khác nhau, dẫn đến độ chính xác phân lớp kiểu trôi thấp. Vì vậy, trong phần này, luận án đề xuất phương pháp cải tiến như trong Hình 3.4, sử dụng các chiến lược cửa sổ khác nhau cho mỗi kiểu trôi: *cửa sổ rời rạc đối xứng cho trôi đột ngột, cửa sổ trượt cho trôi dần, cửa sổ mở rộng cho trôi gia tăng.* Mỗi chiến lược này có sự phù hợp với đặc điểm của từng kiểu trôi.



Hình 3.4 Mô hình VAR-WIND cải tiến bộ trích xuất đặc trưng

3.2.2. Thực nghiệm và đánh giá

Mục tiêu thực nghiệm nhằm so sánh hiệu năng trong giai đoạn tiền huấn luyện giữa hai mô hình: mô hình Meta-ADD và mô hình VAR-WIND.

3.2.2.2 Thiết lập thực nghiệm

Bộ dữ liệu: SEA, Hyperplane, Agrawal, LED. Mỗi tập dữ liệu được chia thành hai phần 80% dành cho huấn luyện, 20% dành cho kiểm tra.

Bảng 3.1. Bảng tổng hợp trích xuất đặc trưng trên bộ Dataset_50-50-4800

Loại luồng	Chiến lược cửa sổ	Tham số chính
Trôi đột ngột	Rời rạc đối xứng	W = 50
Trôi dần dần	Trượt	W = 50, S = 5
Trôi gia tăng	Mở rộng	Lmin = 25, Lmax= n = 50, Δ = 5
Bình thường	Trượt	W = 50, S = 5

Bảng 3.2 Bảng tổng hợp trích xuất đặc trưng trên bộ Dataset_200-25-3800

Kiểu trôi	Chiến lược cửa sổ	Tham số chính
Đột ngột	Rời rạc đối xứng	W = 25
Dần dần	Cửa sổ trượt	W = 25, S = 5
Gia tăng	Cửa sổ mở rộng	Lmin = 5, Lmax = 25, Δ = 5
Bình thường	Cửa sổ trượt	W = 25, S = 5

3.2.2.3 Độ đo hiệu năng: độ chính xác và điểm F1

3.2.2.4 Kết quả và đánh giá

Loại trôi	Độ chính xác		Điểm F1	
	Meta-ADD	VAR-WIND	Meta-ADD	VAR-WIND
Đột ngột	0.9612	**0.9855**	0.9519	**0.9865**
Dần dần	0.9425	**0.9516**	0.9522	**0.9624**
Gia tăng	0.8911	**0.9860**	0.8904	**0.9862**
Bình thường	0.9478	**0.9707**	0.9433	**0.9649**

Bảng 3.3 So sánh trên bộ dữ liệu 50-50-4800

Loại trôi	Độ chính xác P		Điểm F1	
	Meta-ADD	VAR-WIND	Meta-ADD	VAR-WIND
Đột ngột	0.5698	**0.9103**	0.7043	**0.9324**
Dần dần	0.9496	**0.9632**	0.7867	**0.9514**
Gia tăng	0.9167	**0.9741**	0.9246	**0.9743**
Bình thường	0.9744	**0.9804**	0.9532	**0.9645**

Bảng 3.4 So sánh trên bộ dữ liệu 200-25-3800

Chiến lược cửa sổ linh hoạt cho thấy hiệu suất vượt trội so với chiến lược cửa sổ rời rạc trên tất cả các kiểu trôi khái niệm, xét theo cả độ chính xác và điểm F1. Sự vượt trội này đặc biệt rõ rệt đối với các trường hợp trôi khái niệm dạng tăng dần và đột ngột, nơi khả năng thích ứng với sự thay đổi phân phối dữ liệu đóng vai trò quan trọng.

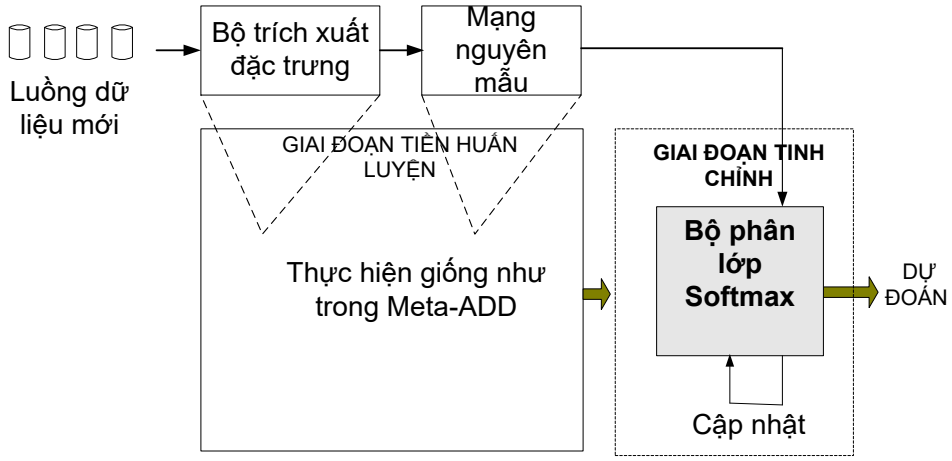
3.3 Mô hình phân lớp trôi khái niệm MetaLDD-Finetune đề xuất

3.3.1 Mô hình MetaLDD-Finetune đề xuất

Trong giai đoạn dự đoán, Meta-ADD đề xuất cập nhật mạng nguyên mẫu bằng cách thủ công tốn kém. Do vậy luận án đề xuất cách giải quyết thay thế bộ phân lớp (dựa trên khoảng cách đến tâm các lớp) của Meta-ADD bằng bộ phân lớp Softmax, sau đó cập nhật bộ phân lớp này, bằng cách huấn luyện trên cả tập dữ liệu hỗ trợ và tập dữ liệu truy vấn của luồng dữ liệu mới, với một hàm mất mát kết hợp, như thể hiện ở Hình 3.8.

Trong giai đoạn tiền huấn luyện, thực hiện trích xuất xác đặc trưng và huấn luyện mạng nguyên mẫu theo phương pháp của Meta-ADD.

Giai đoạn tinh chỉnh: giữ nguyên mạng nguyên mẫu từ gia đoạn tiền huấn luyện, chỉ thay thế bộ phân lớp trong Meta-ADD bằng bộ phân lớp Softmax. Mục tiêu chính của giai đoạn này là cập nhật bộ phân lớp Softmax sao cho nó có thể phân lớp chính xác các lớp mới với chỉ một vài mẫu huấn luyện của luồng dữ liệu mới. Ý tưởng tinh chỉnh như sau:



Hình 3.8 Mô hình MetaLDD-Finetune đề xuất

Khởi tạo trọng số bộ phân lớp:

Ta khởi tạo $b = 0$ và W được khởi tạo như sau: đối với mỗi lớp trôi k, vector trọng số $\boldsymbol{w}_k$ của bộ phân lớp Softmax được khởi tạo bằng trọng số của vector tâm cụm $\boldsymbol{c}_k$ của các mẫu hỗ trợ thuộc lớp đó. Tâm cụm của lớp $\boldsymbol{c}_k$ được tính bằng cách lấy trung bình của tất cả các vector đặc trưng của các mẫu trong tập hỗ trợ S_k của lớp k:

$$\boldsymbol{w}_k = \boldsymbol{c}_k = \frac{1}{|S_k|}\sum_{\boldsymbol{x}_i \in S_k} f_{\phi}(\boldsymbol{x}_i) \quad (3.14)$$

Các vector này trở thành trọng số khởi tạo của lớp phân loại Softmax, giúp mô hình có điểm xuất phát tốt hơn trước khi huấn luyện.

Cập nhật bộ phân lớp Softmax

Bộ phân lớp Softmax sau đó tiếp tục được cập nhật bằng cách huấn luyện theo cách thức *có giám sát truyền thống*, sử dụng cả các mẫu hỗ trợ và mẫu truy vấn trong tập tinh chỉnh, trong đó cần tối thiểu hóa hàm mất mát tổng thể $L_{fine-tune}$, là kết hợp của cả hàm mất mát độ bất định chéo và điều chuẩn độ bất định:

$$L_{fine-tune} = L_{cross-entropy} + \lambda H(p) \quad (3.15)$$

Trong đó:

$L_{cross-entropy}$ là mất mát entropy chéo trung bình của các mẫu **trong tập huấn luyện *S*.**

$$L_{cross-entropy} = (1/N_s)\sum_{(x,y)\in S} -log\left(p_{\phi}(y|x)\right) \quad (3.16)$$

λ là tham số số điều chuẩn độ bất định.

$H(p)$: độ bất định trung bình của phân phối dự đoán trên mẫu **trong tập kiểm tra Q:**

$$H(p) = \left(1/N_Q\right)\sum_{(x,y)\in Q} H\left(p_{\phi}(\cdot\,|x)\right) \quad (3.17)$$

3.3.2 Thực nghiệm và đánh giá

3.3.2.1 Mục tiêu thực nghiệm: so sánh hiệu năng giữa mô hình Meta-ADD và mô hình MetaLDD-Finetune.

3.3.2.2 Thiết lập thực nghiệm

Bộ dữ liệu: SEA, Hyperplane, Agrawal, LED

3.3.2.3 Cấu hình mô hình: Kiến trúc mạng nhúng: FAN. Bộ phân lớp Softmax được huấn luyện theo phương thức học có giám sát bằng giảm Gradient ngẫu nhiên sử dụng tối ưu hóa Adam, với tốc độ học là 0.01, $\lambda = 0.1$.

3.3.2.4 Các độ đo: Hai chỉ số chính được sử dụng là độ chính xác và điểm F1. Ngoài ra, độ chính xác phát hiện trôi và thời gian dự đoán.

Kiểu trôi	Độ chính xác		Điểm F1	
	Meta-ADD	MetaLDD-Finetune	Meta-ADD	MetaLDD-Finetune
Đột ngột	0.8913	**0.9104**	0.8910	**0.8915**
Dần dần	**0.7411**	0.7109	**0.7544**	0.7508
Gia tăng	0.7415	**0.7805**	0.7447	**0.7607**
Bình thường	0.8908	**0.9001**	0.8901	**0.9002**

Bảng 3.8 Độ chính xác và điểm F1 trên bộ dữ liệu 50-15-4800

Kiểu trôi	Độ chính xác		Điểm F1	
	Meta-ADD	MetaLDD-Finetune	Meta-ADD	MetaLDD-Finetune
Đột ngột	**0.7104**	0.6908	**0.8100**	0.8002
Dần dần	0.9211	**0.9310**	0.8320	**0.8433**
Gia tăng	0.9202	**0.9401**	0.9201	**0.9405**
Bình thường	0.9480	**0.9520**	0.9302	**0.9411**

Bảng 3.10 Độ chính xác và điểm F1 trên bộ dữ liệu 200-25-3800

Kết quả thực nghiệm cho thấy sự cải thiện rõ rệt trong hiệu năng phân lớp kiểu trôi và phát hiện trôi. Đối với bộ dữ liệu 1: MetaLDD-Finetune cải thiện độ chính xác và điểm F1 cho các kiểu trôi tăng dần, bình thường và đột ngột, nhưng giảm nhẹ với trôi dần. Trung bình độ chính xác tăng từ 0.87 → 0.88. Đối với bộ dữ liệu 2: MetaLDD-Finetune tăng nhẹ độ chính xác và điểm F1 cho trôi tăng dần, bình thường và dần, nhưng giảm nhẹ với trôi đột ngột. Trung bình độ chính xác tăng từ 0.82 → 0.83.

Ngoài ra kết quả còn cho thấy MetaLDD-Finetune không chỉ giữ hoặc nâng cao độ chính xác phát hiện trôi, mà còn giảm rõ rệt thời gian dự đoán – minh chứng cho khả năng áp dụng trong các hệ thống thời gian thực. Như vậy có thể thấy MetaLDD-Finetune giúp cải thiện đáng kể độ chính xác và giảm thời gian xử lý trong cả hai bộ dữ liệu, đặc biệt hiệu quả cho trôi tăng dần và bình thường.

3.4 Kết luận chương 3

Chương này đã trình bày nhóm đóng góp thứ hai của luận án thông qua việc đề xuất hai mô hình phân lớp trôi khái niệm nhằm nâng cao hiệu quả phân lớp các dạng trôi trong bối cảnh học ít mẫu. Các mô hình được phát triển dựa trên mạng nguyên mẫu. Kết quả nghiên cứu theo nhóm đóng góp này của luận án được công bố trong [TungNK5] cho VAR-WIND, [TungNK3] cho MetaLDD-Finetune.

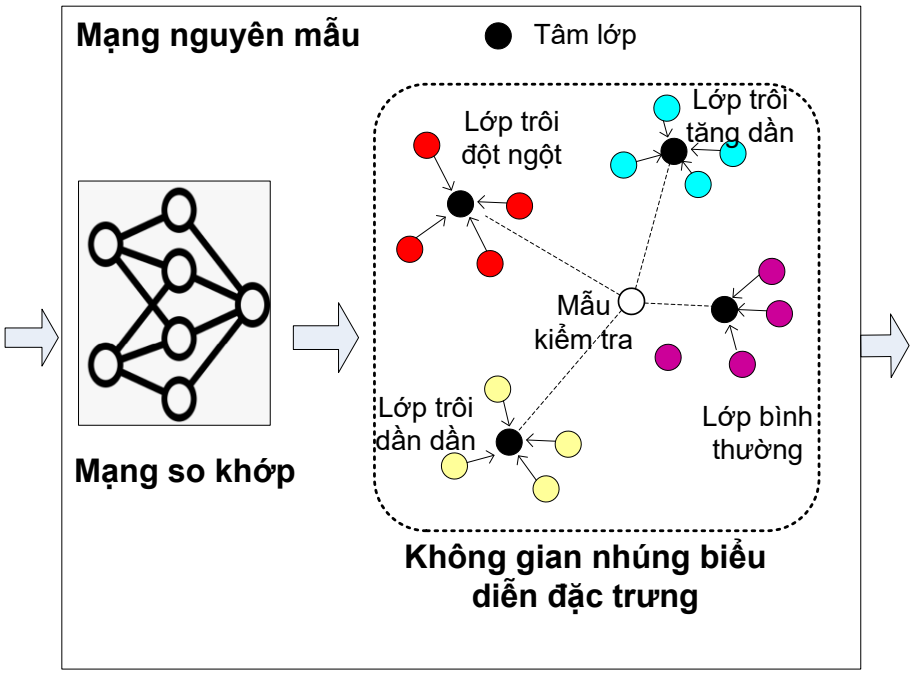
Trong chương tiếp theo, luận án đề cập đến vấn đề tối ưu không gian biểu diễn để từ đó đề xuất các kỹ thuật phân lớp kiểu trôi hiệu quả.

Chương 4. Phát triển mô hình phân lớp trôi khái niệm dựa trên phân tích và tối ưu không gian biểu diễn đặc trưng

Chương này trình bày mô hình đề xuất CS&AM_SC (Cosine Similarity and Angular Margin based Softmax Classifier), như một cải tiến của mô hình MetaLDD-Finetune nhằm nâng cao chất lượng biểu diễn đặc trưng.

4.1 Không gian biểu diễn đặc trưng

Trong các hệ thống học máy áp dụng cho luồng dữ liệu động, đặc biệt là trong bài toán phát hiện và phân lớp trôi khái niệm, việc xây dựng một không gian biểu diễn đặc trưng hiệu quả là yếu tố rất quan trọng để đảm bảo hiệu năng phân lớp lâu dài và ổn định [Yu22]. Một trong những hướng tiếp cận mang tính nền tảng để giải quyết vấn đề trên là học một không gian biểu diễn đặc trưng sao cho các mẫu dữ liệu thuộc cùng một lớp nằm gần nhau, đồng thời mẫu dữ liệu tương ứng với các lớp khác nhau phải được phân tách rõ ràng, được minh họa ở Hình 4.1.



Hình 4.1 Không gian biểu diễn đặc trưng của mạng nguyên mẫu

Mô hình MetaLDD-Finetune tập trung vào việc tinh chỉnh bộ phân lớp Softmax mà *chưa* đề cập đến việc *tối ưu không gian biểu diễn đặc trưng*, tức là làm thế nào để tối ưu cả mạng nhúng và cả bộ phân lớp Softmax trong giai đoạn tinh chỉnh. Vì vậy chương này tập trung làm rõ hai vấn đề then chốt có ảnh hưởng trực tiếp đến chất lượng của không gian biểu diễn đặc trưng trong bài toán phân lớp trôi khái niệm:

i. Độ biến thiên trong lớp
ii. Mức độ phân tách giữa các lớp.

Trong mô hình CS&AM_SC đề xuất, như minh họa tại Hình 4.2, quá trình huấn luyện được chia làm hai giai đoạn:

(1) Giai đoạn tiền huấn luyện:

Ở giai đoạn này, mạng so khớp được huấn luyện trên tập dữ liệu có nhãn (gồm các luồng dữ liệu được gán kiểu trôi). Chiến lược huấn luyện theo tập nhiệm vụ được sử dụng để học biểu diễn đặc trưng phân biệt

giữa các kiểu trôi. Tại thời điểm này, mô hình chưa áp dụng các kỹ thuật độ tương đồng Cosin, hàm mất mát trung tâm, biên góc. Mục tiêu giai đoạn này là huấn luyện một mạng so khớp có khả năng khái quát tốt, tạo tiền đề cho giai đoạn tinh chỉnh.

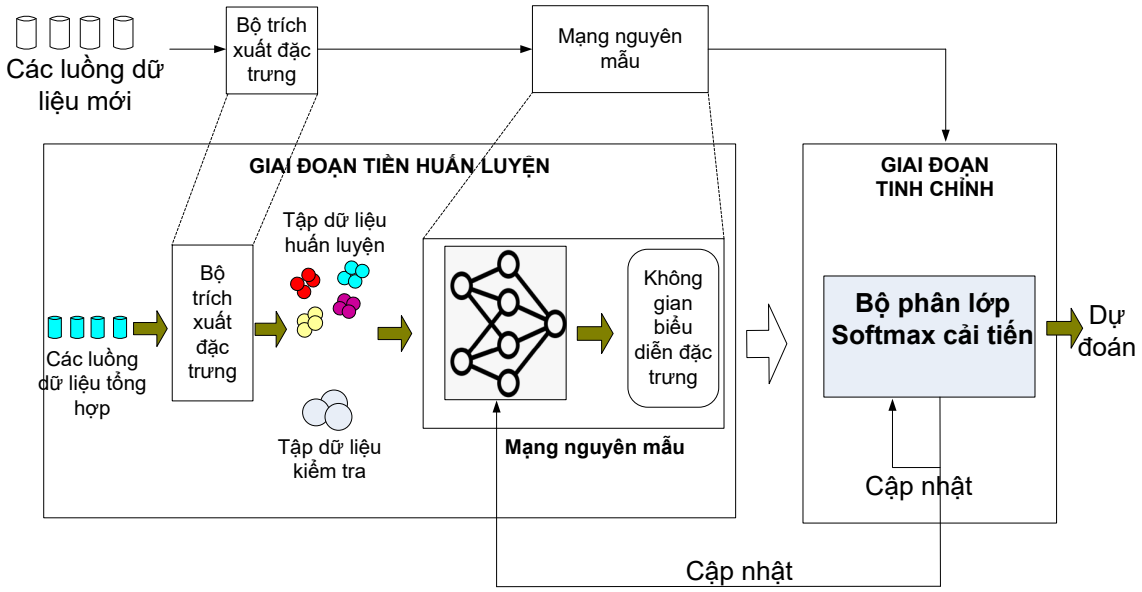
(2) Giai đoạn tinh chỉnh (Fine-tuning):

Sau khi kết thúc tiền huấn luyện, mạng so khớp được giữ nguyên tham số và được sử dụng làm khởi tạo ban đầu cho giai đoạn tinh chỉnh. Giai đoạn này được thực hiện trên các tập nhiệm vụ mới đại diện cho các luồng dữ liệu chưa từng thấy. Khác với MetaLDD-Finetune – vốn chỉ cập nhật bộ phân lớp Softmax – mô hình CS&AM_SC thực hiện cập nhật đồng thời cả mạng so khớp và bộ phân lớp Softmax bằng một hàm mất mát tích hợp. Cụ thể, hàm mất mát tổng được xây dựng từ ba thành phần:

i. Hàm Softmax dựa trên độ tương đồng Cosine và biên góc.

ii. Hàm mất mát trung tâm.

iii. Giữ lại thành phần điều chuẩn độ bất định trong MetaLDD-Finetune để tăng độ chắc chắn của dự đoán.



Hình 4.2 Mô hình CS&AM_SC đề xuất

4.2 Biến thiên trong lớp và giải pháp làm giảm

Giả sử ta có một tập dữ liệu $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ trong đó $\boldsymbol{x}_i$ là mẫu đầu vào và $y_i \in \{1,2,\dots,K\}$ là nhãn lớp tương ứng. Một hàm f_ϕ ánh xạ $\boldsymbol{x}_i$ thành một vector trong không gian đặc trưng $f_\phi(\boldsymbol{x}_i) \in R^d$ (gọi là vector đặc trưng), với d là số chiều của không gian biểu diễn, ϕ là tham số.

Độ biến thiên trong lớp của lớp k được định nghĩa:

$$\mathrm{Var}_{\mathrm{intra}}(k) = \frac{1}{|S_k|}\sum_{\boldsymbol{x}_i \in S_k} ||f_\phi(\boldsymbol{x}_i) - \boldsymbol{c}_k||_2^2 \qquad (4.1)$$

Trong đó:

- $S_k \subset S$ là tập các mẫu có nhãn $y_i = k$,
- $\boldsymbol{c}_k = \frac{1}{|S_k|}\sum_{\boldsymbol{x}_i \in S_k} f_\phi(\boldsymbol{x}_i)$ là tâm của lớp k.

Nếu $\mathrm{Var}_{\mathrm{intra}}(k)$ lớn, điều đó cho thấy các mẫu trong lớp k phân tán mạnh trong không gian đặc trưng.

4.2.1 Thay thế tích vô hướng trong Softmax bằng độ tương đồng Cosine

Độ tương đồng Cosin là một chỉ số đo mức độ tương đồng về hướng giữa hai vector, độc lập với độ lớn của chúng. Với hai vector $\boldsymbol{f}_{\phi}(\boldsymbol{x}),\ \boldsymbol{c}_k \in R^d$ độ tương đồng Cosin được định nghĩa như sau:

$$\cos(\theta_k) = \frac{\boldsymbol{f}_{\phi}(\boldsymbol{x}) \cdot \boldsymbol{c}_k}{||\boldsymbol{f}_{\phi}(\boldsymbol{x})|| \cdot ||\boldsymbol{c}_k||} \tag{4.2}$$

Trong đó:

- $\boldsymbol{f}_{\phi}(\boldsymbol{x}) \cdot \boldsymbol{c}_k$ là tích vô hướng giữa hai vector,
- $\|\cdot\|$ là chuẩn Euclid bậc hai,
- θ_k: là góc giữa hai vector $\boldsymbol{f}_{\phi}(\boldsymbol{x})$ và $\boldsymbol{c}_k$,
- $\cos(\theta_k)$: độ tương đồng Cosin giữa $\boldsymbol{f}_{\phi}(\boldsymbol{x})$ và $\boldsymbol{c}_k$.

Khi sử dụng độ tương đồng Cosin làm cơ sở phân lớp thay vì tích vô hướng, mục tiêu tối ưu của mô hình là tăng độ tương đồng về hướng giữa vector đặc trưng và tâm lớp — đồng nghĩa với việc các mẫu cùng lớp được học để có hướng gần nhau, bất kể độ lớn ban đầu của chúng.

Khi tích hợp độ tương đồng Cosin vào bộ phân lớp Softmax, ta thay thế tích vô hướng trong hàm Softmax bằng độ tương đồng Cosin:

$$p_k = \frac{\exp(\tau \cdot \cos(\theta_k))}{\sum_{j=1}^{K} \exp(\tau \cdot \cos(\theta_j))} \tag{4.3}$$

Trong đó:

- $\cos(\theta_k) = \frac{\boldsymbol{f}_{\phi}(\boldsymbol{x}) \cdot \boldsymbol{c}_k}{||\boldsymbol{f}_{\phi}(\boldsymbol{x})|| \cdot ||\boldsymbol{c}_k||}$ là độ tương đồng Cosin giữa vector đặc trưng và tâm cụm lớp k, với θ_k là góc giữa hai vector.
- $\tau \in R^+$ là hệ số điều chỉnh nhằm khuếch đại sự phân biệt giữa các lớp,
- K là số lượng lớp.

4.2.2 Sử dụng hàm mất mát trung tâm

Độ tương đồng Cosin chỉ điều chỉnh góc giữa các vector đặc trưng, trong khi lại không trực tiếp kiểm soát khoảng cách tuyệt đối giữa các mẫu trong cùng một lớp. Để giải quyết vấn đề trên, một giải pháp bổ sung có tính trực tiếp và rõ ràng là sử dụng một hàm mất mát được thiết kế để tối thiểu hóa khoảng cách hình học giữa các mẫu và tâm lớp tương ứng.

Giả sử hàm nhúng của mô hình là $f_{\phi}(x) \in R^d$, ánh xạ mẫu đầu vào $\boldsymbol{x}$ vào không gian đặc trưng R^d. Với mỗi lớp $k \in 1,2,\dots,K$, ta định nghĩa một vector trung tâm lớp $\boldsymbol{\mu}_k \in R^d$, được cập nhật động trong quá trình huấn luyện.

Hàm mất mát trung tâm được định nghĩa như sau:

$$\mathcal{L}_{\text{center-loss}} = \frac{1}{2Z} \sum_{i=1}^{Z} ||\boldsymbol{f}_{\phi}(\boldsymbol{x}_i) - \boldsymbol{\mu}_{y_i}||_2^2 \tag{4.4}$$

Trong đó:

Trong đó:

- Z là số lượng mẫu trong tập tập huấn luyện và tập kiểm tra,
- $\boldsymbol{x}_i$ là mẫu đầu vào thứ i, y_i là nhãn lớp tương ứng,
- $\boldsymbol{\mu}_{yi}$ là vector trung tâm lớp tương ứng với lớp y_i,

Hàm mất mát này phạt các giá trị mã hóa nằm xa tâm lớp của chúng, đồng thời thúc đẩy tất cả các mẫu trong cùng một lớp hội tụ về một vị trí trung tâm trong không gian biểu diễn.

Các vector $\boldsymbol{\mu}_k$ là tham số có thể học được, và được cập nhật sau mỗi vòng huấn luyện bằng cách lấy trung bình giá trị nhúng của các mẫu thuộc lớp đó. Cụ thể:

$$\mu_k^{(t+1)} = \mu_k^{(t)} - \alpha \cdot \Delta\mu_k \quad (4.5)$$

Cơ chế cập nhật này đảm bảo rằng tâm lớp không bị cập nhật đột ngột mà được điều chỉnh dần dần, từ đó tạo ra một tâm lớp ổn định, đại diện tốt cho toàn bộ phân phối lớp trong không gian biểu diễn.

4.3 Chồng lấn giữa các lớp và giải pháp biên góc

Biên góc được đưa vào công thức (4.3) bằng cách thêm một giá trị m vào góc θ như sau:

$$p_k = \frac{\exp(\tau \cdot \cos(\theta_k + m))}{\sum_{j=1}^{K} \exp(\tau \cdot \cos(\theta_j))} \quad (4.6)$$

Với $m > 0$, giá trị cosine sẽ giảm đi vì $cos(\theta + m) < cos(\theta)$. Điều này buộc mô hình phải giảm góc giữa $\boldsymbol{f}_{\phi}(\boldsymbol{x})$ và $\mathbf{c}_k$ xuống hơn nữa để yêu cầu mô hình phải hội tụ chặt hơn mới đạt được xác suất cao.

4.4. Mô hình CS&AM_SC đề xuất

4.4.1 Hàm mất mát kết hợp

Trong MetaLDD-Finetune, giai đoạn tinh chỉnh chỉ tối ưu bộ phân lớp trong khi giữ cố định mạng so khớp f_{ϕ}. Mặc dù điều này giúp giảm chi phí huấn luyện và tránh quá khớp, nhưng lại hạn chế khả năng điều chỉnh không gian biểu diễn đặc trưng theo các yêu cầu phân loại mới, tức là phải giảm biến thiên trong cùng một lớp và tăng được sự phân tách giữa các lớp.

Để khắc phục hạn chế này và thực sự tối ưu hóa không gian biểu diễn đặc trưng, luận án đề xuất một phương pháp tinh chỉnh trong CS&AM_SC: đồng thời cập nhật cả mạng so khớp f_{ϕ} và bộ phân lớp Softmax. Hàm mất mát tổng hợp được thiết kế để tối ưu đồng thời cả hai mục tiêu, có dạng:

$$\mathcal{L}_{total} = \mathcal{L}_{cos-softmax+margin} + \lambda \cdot \mathcal{L}_{\text{center-loss}} \quad (4.7)$$

Trong đó:

- $\mathcal{L}_{cos-softmax+margin}$: hàm mất mát sử dụng độ tương đồng Cosin và biên góc, mở rộng phân tách lớp;

$$\mathcal{L}_{cos-softmax+margin} = -\log\left(\frac{\exp\left(\tau \cdot \cos\left(\theta_{y_i} + m\right)\right)}{\sum_{j=1}^{K} \exp(\tau \cdot \cos(\theta_j))}\right) \quad (4.8)$$

- $\mathcal{L}_{\text{center-loss}}$: hàm mất mát trung tâm, được thể hiện ở công thức (4.4) giảm khoảng cách giữa các mẫu và tâm lớp → giảm biến thiên và tăng gắn kết;
- λ: hệ số điều chỉnh ảnh hưởng của mất mát trung tâm;

4.4.2 Chiến lược huấn luyện

Phần này mô tả chi tiết quá trình tinh chỉnh, được thực hiện sau khi hoàn tất tiền huấn luyện mạng so khớp.

Đầu vào: • Tập huấn luyện *D_train* gồm nhiều kiểu trôi có nhãn. • Số lượng vòng huấn luyện T. • Các siêu tham số: τ, m, λ, η, α
Đầu ra: • Mạng so khớp đã tinh chỉnh f_{ϕ} • Bộ phân lớp Cosine Softmax đã huấn luyện
Begin

1. Khởi tạo tham số mạng so khớp f_ϕ thông qua tiền huấn luyện trên *D_train*

2. Khởi tạo vector trung tâm lớp:

3. $\forall k \in \{1, ..., K\}$, đặt $\boldsymbol{\mu}_k \leftarrow 0$

3. **for** t = 1 → T **do**

4. Sinh một tập huấn luyện gồm:

- Tập huấn luyện S_k
- Tập kiểm tra Q_k
- $D\_batch \leftarrow S_k \cup Q_k$

5. Tính vector đặc trưng cho tất cả mẫu:

$\forall x_i \in D\_batch$: Chuẩn hóa $f_\phi(\boldsymbol{x}_i)$

6. Tính tâm $\boldsymbol{c}_k$ của mỗi lớp:

$$\forall k, \ \mathbf{c}_k \leftarrow \frac{1}{|S_k|}\sum_{\boldsymbol{x}_i \in S_k} f_\phi(\boldsymbol{x}_i)$$

7. Khởi tạo trọng số Softmax:

$\forall k, \ \boldsymbol{w}_k \leftarrow \boldsymbol{c}_k$

8. Tính loss cosine softmax có margin: $\mathcal{L}_{cos-softmax+margin}$

$\forall x_i \in D\_batch$:

$\theta_j = góc\ giữa\ (f_\phi(\boldsymbol{x}_i), \boldsymbol{w}_j)$

$$\mathcal{L}_{cos-softmax+margin} \leftarrow -\log\left(\frac{\exp\left(\tau\cdot\cos\left(\theta_{y_i}+m\right)\right)}{\sum_{j=1}^{K}\exp(\tau\cdot\cos(\theta_j))}\right)$$

9. Tính loss trung tâm:

$$\mathcal{L}_{\text{center-loss}} \leftarrow \frac{1}{2Z}\sum_{i=1}^{Z}||f_\phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_{y_i}||_2^2$$

10. Tổng hợp hàm mất mát:

$$\mathcal{L}_{total} \leftarrow \mathcal{L}_{cos-softmax+margin} + \lambda \cdot \mathcal{L}_{\text{center-loss}}$$

11. Lan truyền ngược:

- Cập nhật tham số của mạng so khớp f_ϕ theo $\nabla_\phi \mathcal{L}_{total}$
- Cập nhật trọng số w_k của bộ phân lớp Softmax

12. Cập nhật vector trung tâm lớp:

$\forall k$:

$$\overline{f_k} \leftarrow \frac{1}{N_k}\sum_{i:\, y_i=k} f_\phi(\boldsymbol{x}_i)$$

$$\Delta\mu_k \leftarrow \frac{1}{1+N_k}\left(\boldsymbol{\mu}_k - \overline{\boldsymbol{f}_k}\right)$$

$$\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k - \alpha \cdot \Delta\boldsymbol{\mu}_k$$

14. **end for**

15. Trả về f_ϕ, bộ phân lớp Softmax

End

Thực nghiệm và đánh giá

4.5.1 Mục tiêu và kịch bản thực nghiệm:

Mục tiêu của thực nghiệm: đánh giá hiệu năng giữa mô hình MetaLDD-Finetune và mô hình CS&AM_SC.

Kịch bản thực nghiệm: luận án sử dụng hai bộ dữ liệu để tiến hành thực nghiệm. Trên mỗi bộ dữ liệu, tiến hành chạy thử nghiệm cả hai mô hình với các kiến trúc mạng so khớp khác nhau (FCN, FAN, FNN) để kiểm chứng khả năng tối ưu hóa không gian biểu diễn đặc trưng trong CS&AM_SC, so sánh với MetaLDD-Finetune.

4.5.2 Thiết lập thực nghiệm

- Phần cứng: máy tính Dell PowerEdge T40, CPU Intel Xeon E-2224G 3.5 Ghz, 32GB RAM, 2TB HDD.
- Phần mềm: hệ điều hành Windows 10, Pycharm IDE Community, Python 3.11.
- **Bộ dữ liệu**
  - Bộ dữ liệu thứ nhất: Dataset_200-25-3800
  - Bộ dữ liệu thứ hai: Dataset_50-50-4800

  Mỗi bộ dữ liệu được chia thành ba phần: tập huấn luyện, tập tinh chỉnh, và tập kiểm tra theo tỷ lệ 60–20–20.

  Ba kiến trúc mạng so khớp gồm FNN (Feedforward Neural Network), FCN (Fully Convolutional Network) và FAN (Feature Attention Network) được lựa chọn cho quá trình đánh giá

4.5.3 Các độ đo hiệu năng:

Độ chính xác và điểm F1 và thời gian dự đoán.

4.5.4 Trình bày và phân tích kết quả

Bảng 4.4. Độ chính xác phân lớp theo từng kiểu trôi

Kiểu trôi	Cấu trúc mạng so khớp	Dataset_200-25-3800		Dataset_50-50-4800	
		MetaLDD-Finetune	CS&AM_SC	MetaLDD-Finetune	CS&AM_SC
Đột ngột	FCN	0.6233	**0.6547**	0.9510	**0.9513**
	FAN	0.6819	**0.6929**	**0.9504**	0.9213
	FNN	0.6833	**0.6858**	0.9221	**0.9239**
Dần dần	FCN	0.9571	**0.9588**	**0.6808**	0.6643
	FAN	0.9345	**0.9448**	0.9102	**0.9215**
	FNN	0.9242	**0.9455**	0.9243	**0.9334**
Gia tăng	FCN	0.8987	**0.9044**	0.2330	**0.4431**
	FAN	0.9173	**0.9256**	0.8765	**0.9063**
	FNN	0.9402	**0.9534**	0.8812	**0.9067**
Bình thường	FCN	0.9723	**0.9769**	0.9523	**0.9549**
	FAN	0.9514	**0.9656**	0.9475	**0.9488**
	FNN	0.9678	**0.9690**	0.9625	**0.9653**

Bảng 4.5 Độ chính xác trung bình và thời gian dự đoán trên bộ dữ liệu 200-25-3800

Cấu trúc	Trung bình của độ chính xác phân lớp	Điểm F1	Thời gian dự đoán (ms)

	MetaLDD-Finetune	CS&AM_SC	MetaLDD-Finetune	CS&AM_SC	MetaLDD-Finetune	CS&AM_SC
FCN	0.8511	**0.8623**	0.8512	**0.8619**	1510	**1014**
FAN	0.8713	**0.8737**	0.8766	**0.8843**	793	873
FNN	0.8742	**0.8869**	0.8781	**0.8894**	1120	**726**

Bảng 4.6. Độ chính xác trung bình và thời gian dự đoán trên bộ dữ liệu 50-50-4800

Cấu trúc	**Trung bình của độ chính xác phân lớp**		**Điểm F1**		**Thời gian dự đoán (ms)**	
	MetaLDD-Finetune	CS&AM_SC	MetaLDD-Finetune	CS&AM_SC	MetaLDD-Finetune	CS&AM_SC
FCN	0.7012	**0.7367**	0.6928	**0.7332**	**597**	739
FAN	0.9159	**0.9189**	0.9198	**0.9205**	**182**	920
FNN	0.9223	**0.9276**	0.9223	**0.9251**	508	**436**

Ngoài ra CS&AM_SC đạt được độ chính xác và điểm F1 cao hơn trong tất cả các cấu hình mạng so khớp. Mặc dù thời gian dự đoán dao động tuỳ theo cấu trúc mạng, phương pháp vẫn cho thấy hiệu năng chấp nhận được hoặc vượt trội của CS&AM_SC so với MetaLDD-Finetune. Điều này cho thấy hiệu quả của mô hình mới trong việc cải thiện độ chính xác phân lớp kiểu trôi trong khi vẫn duy trì hoặc cải thiện tốc độ dự đoán.

4.6 Kết luận chương 4

Chương này đã phân tích chi tiết ba yếu tố cốt lõi trong việc xây dựng không gian biểu diễn đặc trưng hiệu quả cho bài toán phân lớp kiểu trôi: giảm độ biến thiên trong lớp và mở rộng phân tách giữa các lớp. Thông qua việc kết hợp các kỹ thuật như thay thế tích vô hướng bằng độ tương đồng Cosin trong hàm Softmax, hàm mất mát trung tâm và biên góc, mô hình có thể học được ranh giới các lớp rõ ràng, ổn định và dễ phân biệt trong không gian đặc trưng

Kết luận

Luận án này hướng tới việc phát triển những phương pháp nhằm nâng cao độ tin cậy trong việc phát hiện và phân lớp trôi khái niệm trên luồng dữ liệu. Các kết quả thực nghiệm và phân tích cho thấy tính hiệu quả của các phương pháp đề xuất.

Các kết quả đạt được

Luận án đã đóng góp các hướng cho lĩnh vực phát hiện và phân lớp trôi khái niệm trong dữ liệu luồng như được mô tả sau đây:

Về phát hiện trôi khái niệm

Mô hình học kết hợp **E-ERICS** tận dụng ưu điểm của các mô hình đơn lẻ trong việc giảm các trường hợp dương tính giả, đồng thời bổ sung thêm các điểm dương tính thực. Nhờ đó, mô hình này cải thiện điểm F1 trung bình khoảng **4%** trên các bộ dữ liệu SEA và KDD, và đạt kết quả tốt hơn trong hầu hết các tiêu chí đánh giá so với chỉ sử dụng pha BO, đặc biệt trong việc phát hiện trôi khái niệm đột ngột.

Mô hình **ERICS+3** cũng cho thấy sự cải thiện đáng kể, với mức tăng trung bình khoảng **6,25% về độ hồi tưởng**, **2,4% về điểm F1**, và cải thiện nhẹ về độ chính xác trong việc phát hiện trôi dần dần. Trong một số trường hợp cụ thể, ERICS+3 vượt trội hơn cả E-ERICS và ERICS.

Về phân lớp kiểu trôi khái niệm

Mô hình **VAR-WIND** cho thấy chiến lược cửa sổ linh hoạt luôn đạt hiệu năng cao hơn so với chiến lược cửa sổ rời rạc, khẳng định khả năng thích ứng và phản ứng nhanh với sự thay đổi của luồng dữ liệu. Điều này góp phần nâng cao độ chính xác trong cả phân lớp và phát hiện trôi khái niệm. Mô hình **MetaLDD-Finetune**, bổ sung kỹ thuật tinh chỉnh, không chỉ duy trì hoặc nâng cao độ chính xác trong phát hiện trôi, mà còn giúp giảm đáng kể thời gian dự đoán. Bên cạnh đó, mô hình **CS_AM&SC** cho thấy hiệu năng vượt trội về độ chính xác và điểm F1 trên tất cả các cấu hình mạng so khớp. Mặc dù thời gian dự đoán có dao động tuỳ theo kiến trúc mạng, phương pháp vẫn duy trì hiệu năng cạnh tranh hoặc cao hơn so với MetaLDD-Finetune, cho thấy tính hiệu quả trong việc nâng cao phân loại kiểu trôi đồng thời tối ưu tốc độ.

Hạn chế

Mặc dù luận án đã đề xuất và đánh giá một số phương pháp hiệu quả cho bài toán phát hiện và phân loại loại trôi khái niệm, vẫn còn tồn tại một số hạn chế:

i) Thứ nhất, trong mô hình E-ERICS và ERICS+3, mặc dù luận án đã sử dụng nhiều bộ dữ liệu khác nhau để kiểm chứng phương pháp, các thử nghiệm vẫn chưa bao phủ đầy đủ các tình huống khó, đặc biệt là trong các môi trường có nhiễu cao (ví dụ: dữ liệu chứa nhiều ngoại lai, hoặc các hiện tượng trôi mờ nhạt và không rõ ràng). Đây là những điều kiện thường gặp trong thực tế và có thể ảnh hưởng đáng kể đến hiệu năng của hệ thống, do đó cần được đưa vào phạm vi đánh giá trong các nghiên cứu tiếp theo.

ii) Thứ hai, phương pháp ở mô hình CS&AM_SC sử dụng kiến trúc mạng nguyên mẫu để thực hiện phân lớp kiểu trôi, mà chưa tiến hành đánh giá so sánh với các kiến trúc học từ ít mẫu phổ biến khác như Matching Networks, Siamese Networks hoặc Relation Networks. Việc mở rộng so sánh thực nghiệm với các kiến trúc này có thể giúp làm rõ hiệu quả tương đối của mô hình được đề xuất.

iii) Các mô hình đề xuất trong luận án đều là các đề xuất mang tính kỹ thuật, kết thừa từ các mô hình sẵn có mà chưa đề xuất được một mô hình khái quát dựa trên một khung kỹ thuật chung với các lý giải chặt chẽ về mặt lý thuyết để áp dụng vào các bài toán phát hiện và phân lớp trôi khái niệm.

Định hướng nghiên cứu tiếp theo

Nghiên cứu sinh đang tiến hành các nghiên cứu mở rộng và hoàn thiện các mô hình đã được đề xuất trong luận án, với trọng tâm là khắc phục các hạn chế đã chỉ ra. Cụ thể, hai hướng nghiên cứu đang được triển khai như sau:

Thứ nhất, nhằm nâng cao hiệu quả phát hiện trôi khái niệm trong các điều kiện nhiễu cao và không rõ ràng – một thách thức đối với mô hình E-ERICS và ERICS+3 – nghiên cứu sinh đang phân tích phương pháp DriftLens của Greco và cộng sự. Phương pháp này sử dụng biểu diễn học sâu kết hợp với kỹ thuật phát hiện trôi không giám sát, cho phép nhận diện các thay đổi nhỏ trong phân phối dữ liệu mà không cần nhãn. Dựa trên phương pháp này, nghiên cứu sinh dự kiến áp dụng vào quy trình phát hiện trôi trong E-ERICS/ERICS+3, đồng thời tiến hành thử nghiệm trên các tập dữ liệu có chứa nhiễu mạnh và các dạng trôi mờ nhạt. Kết quả dự kiến sẽ giúp nâng cao độ nhạy và tính ổn định của mô hình trong các môi trường dữ liệu phức tạp hơn so với các thử nghiệm trước đây.

Thứ hai, để cải thiện hiệu năng phân lớp kiểu trôi khái niệm trong điều kiện học từ ít mẫu, nghiên cứu sinh đang tìm hiểu các kiến trúc học từ ít mẫu nền tảng, cụ thể là Matching Networks, Siamese Networks và Relation Networks. Nghiên cứu sinh hiện đang triển khai thực nghiệm so sánh trực tiếp với mạng nguyên mẫu trong cùng điều kiện và tập dữ liệu, từ đó đánh giá hiệu quả tương đối của từng kiến trúc. Kết quả dự kiến sẽ giúp xác định mô hình tối ưu hơn cho bài toán phân lớp kiểu trôi, hoặc gợi mở các hướng kết hợp kiến trúc để nâng cao hiệu năng phân lớp trong điều kiện dữ liệu thiếu nhãn và phân bố thay đổi.

Hai hướng nghiên cứu này sẽ tiếp tục được hoàn thiện với hy vọng đóng góp vào việc phát triển một hệ thống phát hiện và phân lớp trôi khái niệm có độ chính xác cao, khả năng thích nghi tốt và phù hợp với môi trường ứng dụng thực tiễn.

Danh mục công trình khoa học
của tác giả liên quan tới luận án

1. [TungNK1]. Nguyen, K. T., Tran, T., Nguyen, A. D., Phan, X. H., & Ha, Q. T. (2022, November). *Parameter distribution ensemble learning for sudden concept drift detection.* In Asian Conference on Intelligent Information and Database Systems, pp. 192-203. Cham: Springer Nature Switzerland. **Scopus**, **DBLP**, *01 trích dẫn Scopus*.
2. [TungNK2]. Nguyen, K. T., Tran, T., Nguyen, A. D., Phan, X. H., & Ha, Q. T. (2023, November). *Concept Drift Detection in Data Stream: Ensemble Learning Method for Detecting Gradual Instances*. In 2023 Asia Meeting on Environment and Electrical Engineering (EEE-AM), pp. 01-05. **IEEE**[1] .
3. [TungNK3]. K. -T. Nguyen, Q. -T. Ha, X. -H. Phan and Q. -N. N. Han. *A Fine-Tuning Approach to Improve Concept Drift Type Classification Accuracy*, 2024 16th International Conference on Knowledge and System Engineering (KSE), Kuala Lumpur, Malaysia, 2024, pp. 01-05, doi: 10.1109/KSE63888.2024.11063551. **Scopus**, **DBLP**, **IEEE**.
4. [TungNK4]. Nguyen, K. T., Ha, Q. T., & Phan, X. H. (2024, December). *Enhancing Drift Type Classification Through Intra-class Variation Reduction.* In International Conference on Applied Mathematics and Computer Science, pp. 168-177. Cham: Springer Nature Switzerland. (First Online: 27 August 2025). **Scopus**, **DBLP**.
5. [TungNK5]. Nguyen, K. T., Ha, Q. T., & Phan, X. H. (2024, December) *Dynamic Windowing Strategies for Concept Drift Type Classification in Data Streams*. In 2024 IEEE International Conference on Progress in Informatics and Computing (PIC) (pp. 70-74). **Scopus**, **IEEE**, *01 trích dẫn Scopus.*

[1] https://ieeexplore.ieee.org/author/697470114198698