ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

---------- & ----------

ĐỀ CƯƠNG LUẬN VĂN THẠC SĨ

PHÁT HIỆN CONCEPT DRIFT SỬ DỤNG SHAPE DRIFT DETECTOR TRONG GIAI ĐOẠN XỬ LÝ DỮ LIỆU

Ngành: KHOA HỌC MÁY TÍNH

GVHD: PGS.TS Thoại Nam

—o0o—

HVTH: Lê Phúc Đức – MSSV: 2370116

Thành phố Hồ Chí Minh, tháng 5/2025

Tóm tắt nội dung

Đề tài này tập trung tìm hiểu về việc nghiêng cứu giải thuật ShapeDD dùng để phát hiện sự trôi dạt khái niệm (concept drift), điều thường xảy ra nhiều trong các ứng dụng học máy trong cuộc sống và công nghiệp cùng với những vấn đề gặp phải trong thực tế của các mô hình học máy khi gặp phải concept drift – yếu tố làm ảnh hưởng đến độ chính xác và hiệu năng của các mạng nơ-ron khi áp dụng trong môi trường có dữ liệu biến đổi liên tục về thời gian.

Mục tiêu của đề tài là ứng dụng phương pháp ShapeDD để phát hiện trôi dạt, tìm hiểu cơ sở lý thuyết, cách hoạt động cũng như ứng dụng lên một số tập dữ liệu bao gồm cả thực tế và dữ liệu synthetic để đánh giá về mức độ hiệu quả và tính ứng dụng của giải thuật.

Mục lục

Danh sách hình vẽ

Chương 1

Giới thiệu chung

Trong những năm gần đây, lĩnh vực trí tuệ nhân tạo ngày càng phát triển nhanh chóng. Việc ứng dụng thành quả của trí tuệ nhân tạo ngày càng được phổ biến rộng rãi, không chỉ trong đời sống hằng ngày mà cả trong công việc. Khi các ứng dụng học máy không còn bị giới hạn trong phòng thí nghiệm nữa để được ứng dụng vào trong đời sống trong các lĩnh vực sản xuất như bảo trì thông minh và kiểm soát chất lượng. Khi đó, các câu hỏi liên quan đến độ tin cậy và độ bền liên tục của chúng nảy sinh.

Các tập dữ liệu tĩnh được sử dụng để huấn luyện các mô hình học máy chỉ có thể nắm bắt được một phần nhỏ các điều kiện có thể xảy ra trong thế giới thực. Các trường hợp trôi dạt khái niệm (concept drift), chẳng hạn như thay đổi điều kiện môi trường, thiết bị và vận hành có thể, theo thời gian, làm giảm đáng kể hiệu suất của các mô hình học máy, gây ảnh hưởng đến sự an toàn, độ tin cậy của mô hình và kinh tế nếu không được giải quyết đúng cách. Nội dung của đề cương này được trình bày như sau:

Chương 2 trình bày về concept drift trong học máy, về định nghĩa, phân loại và sự ảnh hưởng của điều đó đến hệ thống thực tế, từ đó nêu ra được tầm quan trọng của các drift detector trong các ứng dụng giám sát.

Chương 3 trình bày chi tiết về phương pháp được nghiêng cứu và nền tảng lý thuyết.

Chương 4 trình bày kết quả sơ bộ đã thực nghiệm đối với dữ liệu tổng hợp (synthetic) cho các trường hợp concept drift giống như trong thực tế.

Chương 5 và cũng là chương cuối cùng – tổng kết lại những công việc mà đề cương đã làm được, những hạn chế của phương pháp và phương hướng phát triển sắp tới trong giai đoạn luận văn.

Chương 2

Concept drift trong học máy

Học máy đã có những đóng góp đáng kể cho sự phát triển của các hệ thống giám sát tình trạng và quy trình tiên tiến trong quá trình sản xuất, cho phép giám sát và phân tích các thiết bị và quy trình sản xuất theo thời gian thực để phát hiện các sai lệch so với hoạt động bình thường và dự đoán các lỗi tiềm ẩn [4, 5].

Trong lĩnh vực quản lý chất lượng, các thuật toán học máy có thể phân tích dữ liệu điều khiển máy móc và cảm biến để phát hiện sớm các lỗi trong quy trình sản xuất, do đó giảm hoặc ngăn ngừa việc sản xuất ra các sản phẩm lỗi. Các nghiên cứu gần đây đã chỉ ra rằng các ứng dụng học máy trong sản xuất không còn giới hạn trong phạm vi học thuật nữa mà ngày càng được các công ty áp dụng và triển khai trong các tình huống thực tế [6, 7].

Khi các ứng dụng học máy vượt ra ngoài phạm vi sử dụng thực tế, các câu hỏi liên quan đến độ tin cậy và hiệu suất lâu dài của chúng nảy sinh. Trong ứng dụng sử dụng cho việc giám sát quy trình, tập dữ liệu huấn luyện của mô hình học máy sẽ bị giới hạn ở một trạng thái nhất định của quy trình sản xuất theo thời gian.

Tuy nhiên, trong quá trình sử dụng mô hình, môi trường sản xuất có thể sẽ gặp phải những thay đổi như hao mòn dụng cụ và máy móc cảm biến hoặc thay đổi trong cách bố trí nhà máy và vị trí đặt máy [8, 9] không được ghi lại trong tập dữ liệu huấn luyện,

một kịch bản được gọi là sự trôi dạt khái niệm (concept drift) [7].

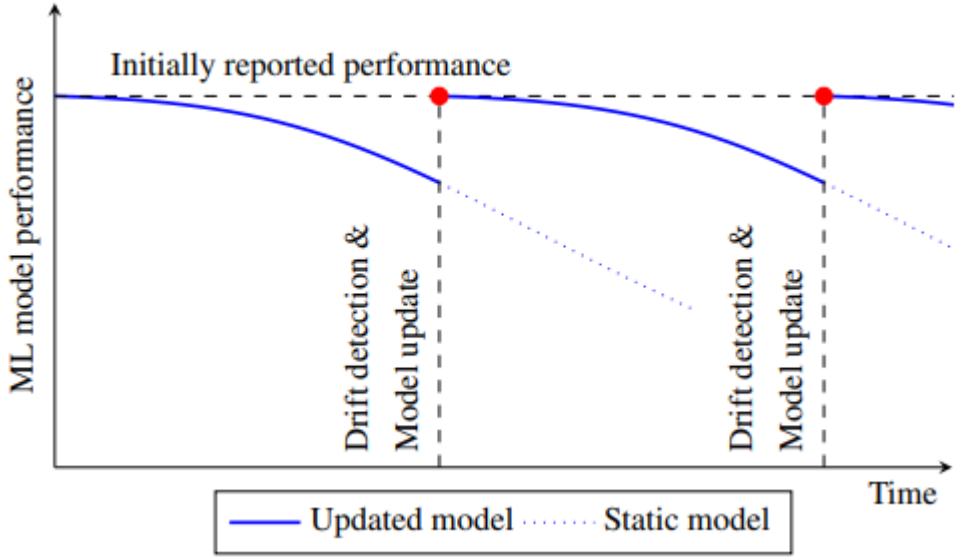

Figure 1 Sự suy giảm về độ chính xác của mô hình khi gặp phải hiện tượng trôi dạt

2.1. Sự ảnh hưởng của concept drift

2.1.1. Khái niệm về concept drift và phân loại

Concept drift, hay còn gọi là sự trôi dạt khái niệm, đề cập đến những thay đổi trong phân phối dữ liệu được tạo ra theo thời gian, đặc biệt là trong môi trường động và thay đổi theo thời gian, chẳng hạn như trong ứng dụng về IoT [12]. Cụ thể hơn, sự trôi dạt khái niệm là một vấn đề trong đó các mối quan hệ thống kê giữa các giá trị đầu vào và giá trị mục tiêu bị thay đổi theo thời gian theo cách không thể dự đoán được [13].

Có nhiều loại trôi dạt khác nhau, tùy thuộc vào các yếu tố dữ liệu đang thay đổi. Các loại chính của sự trôi dạt khái niệm bao gồm:

Sự trôi dạt ảo (Virtual Drift): Còn được gọi là sự dịch chuyển biến phụ, đề cập đến tình huống mà sự thay đổi xảy ra trong phân phối các trường hợp đầu vào, trong khi xác suất sau của các giá trị mục tiêu vẫn không đổi [1].

Sự trôi dạt thực (Real Drift): Sự thay đổi trong xác suất sau của các giá trị mục tiêu (tức là các lớp) được gọi là sự trôi dạt thực. Sự trôi dạt thực có thể không ảnh hưởng đến sự phân phối các trường hợp đầu vào. Ví dụ, người ta có thể đề cập đến sự thay đổi trong sở thích của người dùng khi họ theo dõi các kênh tin tức phát trực tuyến, trong khi sự phân phối các mục tin tức nhận được thường không thay đổi [1].

Sự trôi dạt dần dần và đột ngột (Gradual and abrupt drift): Xét về tốc độ thay đổi, sự trôi dạt có thể được phân loại thành sự trôi dạt dần dần và đột ngột. Sự trôi dạt dần dần biểu thị trường hợp khi sự phân phối dữ liệu thay đổi dần dần theo thời gian, trong khi sự trôi dạt đột ngột có thể xảy ra khi sự thay đổi trong phân phối dữ liệu xảy ra đột ngột [1].

Sự trôi tăng dần (Increamental Drift):, thể hiện cho sự tiến hóa dần dần của phân phối dữ liệu. Ví dụ như sự tiến hóa dần dần của hệ thống đề xuất người dùng ngày càng tiến hóa và nhiều hơn dựa trên sự thay đổi của người dùng.[14]

Sự trôi dạt lặp lại (Recurrent Drift): Trôi dạt lặp lại là khi dữ liệu quay trở lại trạng thái cũ sau một thời gian, hoặc lặp lại theo chu kỳ. Những thay đổi trong dữ liệu không phải mới mà đã từng xảy ra trước đó. Ví dụ như xu hướng thời trang thay đổi theo mùa, tuần hoàn theo từng năm. [14]

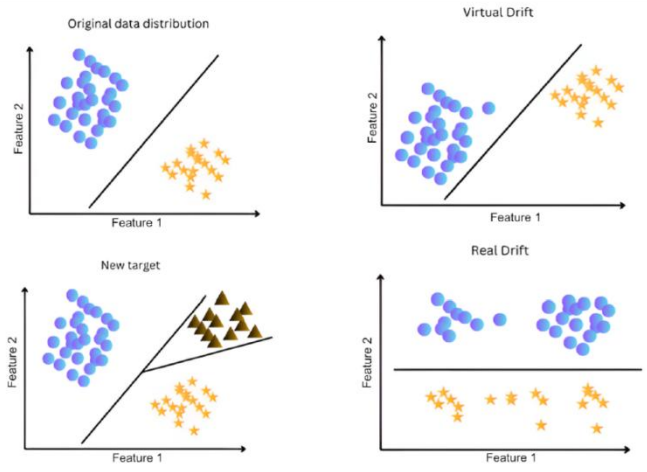

Figure 2 Phân loại các dạng trôi dạt dựa trên sự thay đổi phân phối

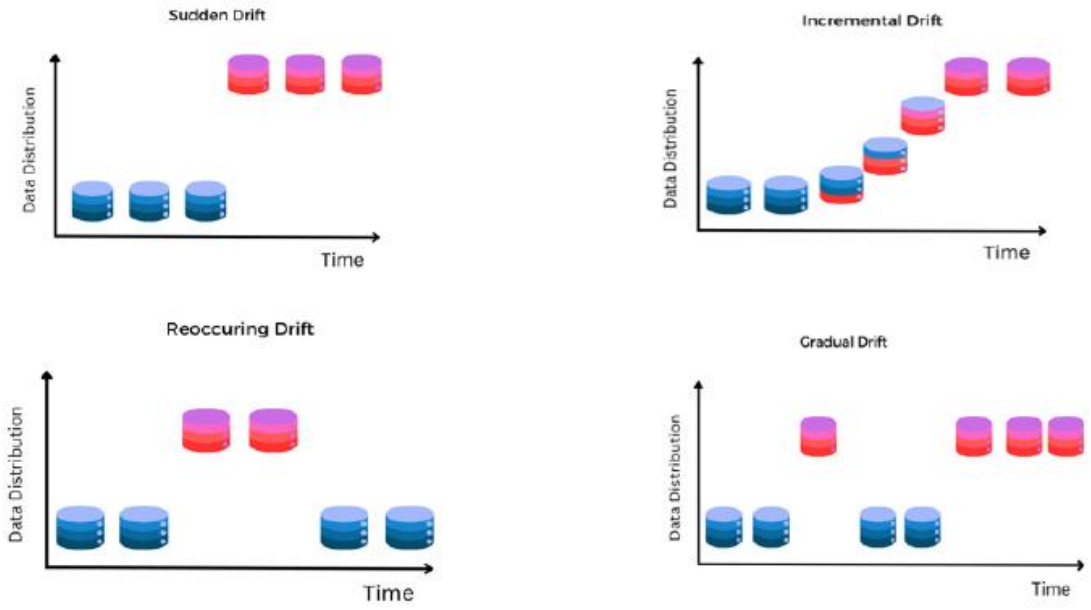

Figure 3 Phân loại cách sự trôi dạt dựa trên sự thay đổi về mẫu dữ liệu

2.1.2. Hậu quả của concept drift và tầm quan trọng của drift detection

Sự trôi dạt khái niệm có thể dẫn đến hiệu suất giảm trong quá trình vận hành thực tế của mô hình học máy, điều này trái ngược với hiệu suất được đánh giá trên tập dữ liệu thử nghiệm tĩnh trong quá trình phát triển.

Sự trôi dạt khái niệm có thể ảnh hưởng lớn đến hiệu suất của mô hình dự đoán, đặc biệt là khi mô hình học từ luồng dữ liệu. Một loạt các dịch vụ/ứng dụng trong bối cảnh hệ thống và mạng truyền thông có thể bị cản trở bởi sự trôi dạt khái niệm như: Hệ thống phát hiện xâm nhập (Intrusion Detection Systems - IDS, hệ thống phân loại và dự đoán lưu lượng và IIoT. Ví dụ, người ta có thể tham khảo các kỹ thuật bảo trì dựa trên tình trạng (Condition-Based Maintenance - CBM) trong IIoT được sử dụng để dự đoán các điều kiện bất thường và thời gian bảo trì thông qua phân tích dữ liệu IIoT. Sự trôi dạt khái niệm ảnh hưởng đáng kể đến hiệu suất của CBM và do đó làm giảm chất lượng sản phẩm.

Điều này có thể được giải thích bởi thực tế là sự phân phối các mẫu lỗi có thể thay đổi theo thời gian do máy móc lão hóa và quy trình bảo trì. Do đó, một kỹ thuật CBM không có khả năng xử lý sự trôi dạt khái niệm sẽ hoạt động kém. Thật vậy, sự trôi dạt khái niệm có thể ảnh hưởng đến hiệu quả và tính mạnh mẽ của phân tích luồng dữ liệu.

Sự trôi dạt khái niệm cũng là vấn đề đối với các ứng dụng IoT khác, chẳng hạn như thành phố thông minh. Trong các thành phố thông minh, dữ liệu có thể được thu thập vì nhiều lý do khác nhau, chẳng hạn như đảm bảo an ninh mạng, dự đoán ô nhiễm không khí, dự đoán giao thông đường bộ và dự báo tải điện. Tuy nhiên, theo thời gian, những thay đổi không lường trước có thể xảy ra trong dữ liệu được thu thập (sự trôi dạt khái niệm) và do đó, nó đặt ra những thách thức nghiêm trọng đối với độ chính xác của các mô hình dự đoán.

Trong môi trường không cố định, có một số cân nhắc mà các mô hình dự đoán phải tính đến để phát hiện và tự thích ứng với sự trôi dạt khái niệm, nếu không, hiệu suất của các mô hình này sẽ giảm sút về độ chính xác và độ mạnh mẽ. Theo thời gian, một mô hình dự đoán có thể cần cập nhật các tham số và cấu trúc của nó bằng cách kết hợp các dữ liệu huấn luyện mới hoặc thay thế hoàn toàn mô hình cũ để xử lý sự trôi dạt khái niệm.

2.2. Tầm quan trọng của Drift Detection

Như đã nhắc đến ở trên, việc trôi dạt xảy ra sẽ ảnh hưởng đến hiệu suất của mô hình. Khi điều đó xảy ra, mô hình ban đầu sẽ cần phải được cập nhật thay đổi để phù hợp với dữ liệu mới, tuy nhiên, điều này cần phải được phát hiện sớm để . Vì vậy, việc phát hiện sớm khi xảy ra hiện tượng trôi dạt sớm sẽ giúp cho việc chuẩn bị cho những phương án tiếp theo trở nên dễ dàng hơn (huấn luyện lại hay cập nhật mô hình với dữ liệu mới).

2.2.1. Các lý thuyết liên quan

Concept Drift có thể được hiểu thông qua các khuôn khổ lý thuyết đã được thiết lập, bao gồm lý thuyết học thống kê (statistical learning theory), suy luận Bayes (Bayesian inference), lý thuyết thông tin (information theory) và lý thuyết học trực tuyến (online learning theory). Các khuôn khổ này cung cấp cơ sở có cấu trúc để hiểu các thách thức và phương pháp liên quan đến phát hiện trôi dạt [14].

Statistical Learning Theory: Lý thuyết học thống kê củng cố khả năng khái quát hóa của các mô hình học máy. Lý thuyết này giả định rằng phân phối xác suất chung P(X,Y) là không thay đổi theo thời gian. Tuy nhiên, sự trôi dạt khái niệm vi phạm giả định này, dẫn đến hiệu suất mô hình bị suy giảm. Sự thích nghi của mô hình là cần thiết để giải quyết những vi phạm này và khôi phục hiệu suất mô hình [14].

Bayesian Inference: Suy luận Bayesian cung cấp một khuôn khổ xác suất để cập nhật mô hình khi dữ liệu mới có sẵn. Sự trôi dạt khái niệm có thể được xem như một quá trình liên tục cập nhật các phân phối trước đó để thích ứng với những sự thay đổi đang phát triển. Ví dụ: Trong dự báo tài chính, các mô hình Bayesian cập nhật động các dự đoán để phản ánh những thay đổi trong điều kiện thị trường, đảm bảo đánh giá rủi ro chính xác hơn.

Information Theory: Các biện pháp dựa trên lý thuyết thông tin, chẳng hạn như entropy và độ phân kỳ Kullback–Leibler (KL), thường được sử dụng để định lượng những thay đổi trong phân phối dữ liệu. Ví dụ: Độ phân kỳ KL có thể được sử dụng để so sánh các đặc tính thống kê của luồng dữ liệu đến với dữ liệu lịch sử, đánh dấu các độ lệch đáng kể là độ trôi tiềm ẩn.

Online Learning Theory: Lý thuyết học trực tuyến liên quan đến các việc cập nhật mô hình liên tục khi dữ liệu mới đến. Điều này nhấn mạnh vào việc cân bằng tính ổn định (bảo tồn dữ liệu trong quá khứ) với tính mềm dẻo (thích ứng với các mẫu mới).Ví dụ: Các mô hình học trực tuyến được sử dụng trong hệ thống đề xuất có thể thích ứng

với các tùy chọn thay đổi của người dùng mà không cần huấn luyện lại toàn bộ hệ thống.

Ý nghĩa thực tiễn của việc tích hợp các khuôn khổ lý thuyết này vào các phương pháp phát hiện trôi dạt khái niệm sẽ tăng cường khả năng thích ứng và hiệu quả của chúng:

- Lý thuyết học thống kê nhấn mạnh sự cần thiết của các mô hình thích ứng để giải quyết các phân phối thay đổi.
- Suy luận Bayesian cung cấp một cơ chế tự nhiên để thích ứng trôi dạt dần dần.
- Các biện pháp lý thuyết thông tin cho phép định lượng chính xác trôi dạt ảo.
- Lý thuyết phát hiện thay đổi cung cấp các công cụ mạnh mẽ để xác định những thay đổi đột ngột.
- Các khuôn khổ học trực tuyến đảm bảo khả năng mở rộng và khả năng thích ứng theo thời gian thực.

Bằng cách đưa phát hiện trôi dạt khái niệm vào các lý thuyết nền tảng này, các nhà nghiên cứu có thể phát triển các mô hình mạnh mẽ, thích ứng được điều chỉnh theo sự phức tạp của các môi trường có nhiều sự biến đổi.

2.2.2. Các phương pháp phát hiện trôi dạt

Tương tự như các nhiệm vụ học máy chung, phát hiện trôi dạt (drift detection) có thể được xem xét trong các bối cảnh học có giám sát, tức là các bối cảnh liên quan đến phân phối có điều kiện, thường liên quan đến một nhãn hoặc mục tiêu, và học không giám sát, tức là các bối cảnh liên quan đến phân phối chung hoặc phân phối biên. Trong khi ở bối cảnh học có giám sát, cả trôi dạt thực (real drift) và trôi dạt ảo (virtual drift) đều có thể xuất hiện, thì ở bối cảnh học không giám sát, chỉ cần xem xét trôi dạt ảo. Đề cương này chỉ tập trung vào việc phát hiện những trôi dạt ảo ở dữ liệu đầu vào.

Dựa trên cơ chế hoạt động, drift detection trong phát hiện trôi dạt ảo có thể được phân loại theo cơ chế tiếp cận chính, nó có thể là two samples, meta-statistics hoặc là block based [2].

Two – sample analysis based: Cách khai thác phổ biến nhất của các drift detector là đo lường sự khác nhau giữa hai thời điểm mà có thể kiểm thử bằng phương pháp thống kê. Để thực hiện điều đó, mẫu dữ liệu đầu vào S(t) được chia thành thành hai mẫu S_(t) và S+(t).

Meta – statistic based: Phương pháp two-sample ở trên là phương pháp đơn giản nhất vì nó cân nhắc các điểm dữ liệu trong luồng một các độc lập. Điều này có thể dẫn đến các vấn đề như multiple testing problem, sub-optimal sensitivity và high computational complexity. Cách tiếp cận theo meta-statistic cố gắng giải quyết một số vấn đề bằng cách kết hợp nhiều giá trị ước lượng để cho ra kết quả tốt hơn thay vì cân nhắc chúng một cách riêng lẻ nhưng vẫn sử dụng cơ chế cửa sổ trượt.

Block – based: Trái ngược với toàn bộ các phương pháp ở trên, block-based không chia dữ liệu thành hai phần khác nhau mà lấy toàn bộ phân đoạn dữ liệu và phân tích chúng cùng một lúc.

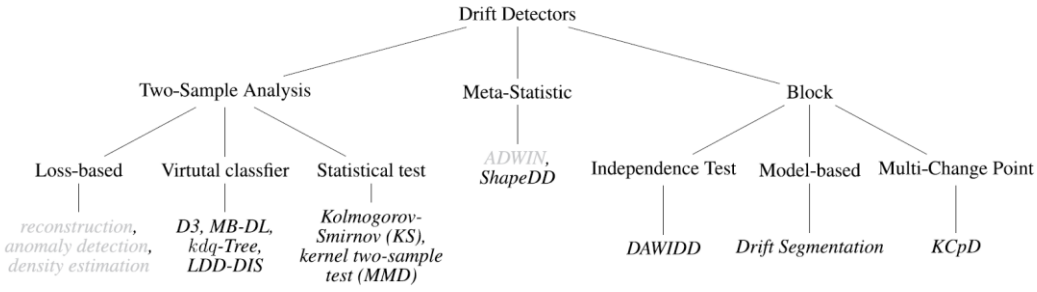

Figure 4 Các phương pháp phát hiện trôi dạt

Như đã đề cập ở trên, khi áp dụng phương pháp two-sample analysis based, chúng ta cần chia mẫu thu được thành hai mẫu con, sau đó được sử dụng cho bài kiểm tra. Bước này rất quan trọng vì khi chia dữ liệu không phù hợp có thể ảnh hưởng lớn đến kết quả. Trong trường hợp nghiêm trọng, việc chọn cách chia không phù hợp có thể

khiến sự trôi dạt biến mất và do đó không thể phát hiện được, vì chúng ta xem xét giá trị trung bình thời gian của các cửa sổ. Khi đó, các phương pháp như Meta – statistic và Block – based được đề xuất hơn trong các ứng dụng này [2].

Tóm lại, việc lựa chọn phương pháp phù hợp với từng loại ứng dụng để đối phó với các trường hợp xảy ra trôi dạt khác nhau là rất quan trọng. Việc lựa chọn phương pháp phù hợp sẽ có thể giúp phát hiện sớm các hiện tượng trôi dạt có thể giúp cho giúp duy trì hiệu suất mô hình, hỗ trợ sửa lỗi mô hình, giảm rủi ro sai sót, và đảm bảo dự đoán đáng tin cậy đồng thời có thể ngăn chặn các quyết định sai lầm trong các ứng dụng quan trọng như y tế hoặc tài chính và đồng thời giúp tiết kiệm chi phí và thời gian bằng cách cập nhật mô hình kịp thời, thay vì phải xây dựng lại hoàn toàn.

Ở chương tiếp theo, đề tài này sẽ trình bày về bài toán nghiên cứu, cũng như chi tiết về phương pháp mà đề tài sư dụng để phát hiện hiện tượng trôi dạt.

Chương 3

Bài toán nghiêng cứu

3.1. Bài toán nghiên cứu

Trong bối cảnh dữ liệu lớn và học máy ngày càng phát triển, các hệ thống học máy thường đối mặt với thách thức khi dữ liệu thay đổi theo thời gian, dẫn đến hiện tượng concept drift. Hiện tượng này xảy ra khi phân phối dữ liệu hoặc mối quan hệ giữa dữ liệu đầu vào và đầu ra thay đổi, làm giảm hiệu suất của các mô hình dự đoán. Việc phát hiện kịp thời và chính xác concept drift đóng vai trò quan trọng trong việc đảm bảo độ tin cậy và hiệu quả của các hệ thống học máy trong các ứng dụng thực tiễn như phân tích tài chính, y tế, và giám sát hệ thống. Đề tài này tập trung vào việc xem xét một phương pháp phát hiện concept drift dựa trên đánh việc đánh giá sự thay đổi phân phôi, đánh giá hiệu quả của nó và đề xuất giải pháp tối ưu nhằm nâng cao khả năng thích ứng của các mô hình học máy trong môi trường dữ liệu động. Về mặt mục tiêu, đề tài này sẽ đề ra 3 mục tiêu cần đạt được:

- Tìm hiểu cơ sở lý thuyết và cách hoạt động của phương pháp
- Thử nghiệm phương pháp với tập dữ liệu synthetic.
- Đánh giá độ chính xác của phương pháp trong các trường hợp trôi dạt khác nhau.

3.1.1. Mục tiêu 1: Tìm hiểu cơ sở lý thuyết và cách hoạt động của phương pháp.

Để xây dựng nền tảng cho đề tài, mục tiêu này tập trung vào việc tìm hiểu và hệ thống hóa các cơ sở lý thuyết liên quan đến hiện tượng concept drift trong học máy. Cụ thể, đề tài này sẽ làm rõ khái niệm concept drift, bao gồm các loại chính như drift đột ngột (sudden drift) và drift gia tăng (incremental drift). Đồng thời, các đặc trưng của dữ liệu dẫn đến concept drift, như thay đổi phân phối dữ liệu đầu vào (covariate shift) sẽ là tập trung của đề tài này.

Về mặt phương pháp, đề tài sẽ dựa trên cơ sở lý thuyết và cách xảy ra của concept drift trong ngữ cảnh không giám sát, từ đó chọn ra phương pháp phù hợp để sử dụng cho đề tài. Phương pháp sẽ được phân tích về cơ sở lý thuyết, cách thức hoạt động bằng ví dụ đơn giản, cũng như ứng dụng lên một tập dữ liệu synthetic với mục đích làm rõ cách các phương pháp này phát hiện sự thay đổi trong dữ liệu thông qua các chỉ số như độ lệch phân phối, sai số dự đoán, hoặc sự thay đổi trong đặc trưng dữ liệu.

Kết quả của mục tiêu này là một báo cáo tổng hợp cung cấp cái nhìn toàn diện về lý thuyết và cơ chế hoạt động của các phương pháp phát hiện concept drift, làm cơ sở cho việc đánh giá và phát triển các giải pháp trong các mục tiêu tiếp theo.

3.1.2. Mục tiêu 2: Thử nghiệm phương pháp với tập dữ liệu synthetic

Mục tiêu của phần này là đánh giá hiệu quả của phương pháp phát hiện concept drift thông qua việc thử nghiệm trên các tập dữ liệu synthetic. Các tập dữ liệu này được tạo ra nhằm mô phỏng các kịch bản concept drift khác nhau, bao gồm drift đột ngột, drift dần dần và drift gia tăng, với các đặc trưng được kiểm soát như kích thước mẫu, mức độ nhiễu và quy mô thay đổi phân phối.

Cụ thể, các tập dữ liệu synthetic sẽ được tạo bằng các công cụ như Scikit-multiflow hoặc MOA, với các tham số được thiết lập để mô phỏng các tình huống thực tế, nhưng tập trung vào việc thay đổi phân phối dữ liệu đầu vào. Phương pháp phát hiện concept drift đã được nghiên cứu trong mục trước sẽ được áp dụng trên các tập dữ liệu này để đánh giá độ chính xác trong việc phát hiện ra các sự kiện trôi dạt.

Kết quả của phần thử nghiệm này là nền tảng để cung cấp kết quả để đánh giá kết quả, ưu và nhược điểm của phương pháp, từ đó định hướng cho việc cải tiến hoặc phát triển phương pháp mới trong các giai đoạn tiếp theo của đề tài.

3.1.3. Mục tiêu 3: Đánh giá độ chính xác của phương pháp trong các trường hợp trôi dạt khác nhau

Mục tiêu này tập trung vào việc đánh giá độ chính xác của phương pháp phát hiện được sử dụng trong các kịch bản trôi dạt khác nhau như trôi đột ngột và trôi tăng dần. Thí nghiệm sẽ được thực hiện trên các tập dữ liệu synthetic với các kịch bản được thiết kế để tái hiện các tình huống thực tế, chẳng hạn như thay đổi phân phối đặc trưng (covariate shift). Kết quả sẽ được phân tích và so sánh thông qua các bảng số liệu và biểu đồ, minh họa hiệu suất của từng phương pháp trong từng kịch bản trôi dạt. Phân tích định tính cũng sẽ được thực hiện để đánh giá tính ổn định và độ nhạy của các phương pháp khi đối mặt với các mức độ nhiễu hoặc thay đổi dữ liệu khác nhau.

Kết quả của mục tiêu này sẽ cung cấp cái nhìn sâu sắc về hiệu quả của các phương pháp phát hiện concept drift, từ đó xác định phương pháp tối ưu hoặc định hướng cải tiến cho các kịch bản cụ thể trong giai đoạn tiếp theo.

3.2. Cơ sở lý thuyết

Nền tảng của concept drift trong dữ liệu, đặc biệt là sự trôi dạt ảo (virtual drift) – tập trung vào sự thay đổi phân phối dữ liệu đầu vào. Sự thay đổi này có thể làm giảm độ chính xác và độ tin cậy của các mô hình dự đoán, gây ra những hậu quả nghiêm trọng trong các ứng dụng thực tiễn như tài chính, y tế hoặc giám sát hệ thống. Để giải quyết vấn đề này, việc phát hiện sớm concept drift là cần thiết. Việc phát hiện ra sự thay đổi phân phối của dữ liệu đầu vào có thể được giải quyết sử dụng Maximum Mean Discrepancy (MMD), đó là một thước đo thống kê mạnh mẽ để so sánh sự khác biệt giữa hai phân phối dữ liệu. MMD hoạt động bằng cách tính toán khoảng cách giữa các biểu diễn trung bình của hai tập dữ liệu trong không gian Hilbert tái tạo hạt nhân, cho phép phát hiện các thay đổi trong phân phối mà không cần giả định về hình dạng phân phối của chúng.

Trong đề tài này, đề tài sẽ khám phá cách MMD được áp dụng để phát hiện concept drift, phân tích nền tảng lý thuyết. Từ đó tiến đến việc phân tích và tìm hiểu về Shape Drift Detector – một phương pháp dựa trên MMD để xác định sự khác biệt về phân phối của hai cửa sổ dữ liệu, từ đó triển khai phương pháp đối với tập dữ liệu ví dụ và

thực hiện với tập dữ liệu synthetic nhằm đảm bảo hiệu suất của phương pháp ổn định trong môi trường dữ liệu động.

3.2.1. Maximum Mean Discrepancy (MMD)

Maximum Mean Discrepancy (MMD) [3] là một thước đo thống kê dùng để so sánh hai phân phối xác suất P và Q. Ý tưởng chính là ánh xạ dữ liệu từ không gian ban đầu vào một Reproducing Kernel Hilbert Space (RKHS), sau đó đo lường sự khác biệt giữa kỳ vọng của hai phân phối trong không gian này. MMD được định nghĩa ban đầu dựa trên một lớp hàm F như sau.

$$MMD(P, Q; F) = \sup_{\|f\| \leq 1} \left| E_{X \sim P}[f(X)] - E_{Y \sim Q}[f(Y)] \right|$$

Trong đó:

- P và Q: là 2 phân phối mà chúng ta muốn so sánh.
- X ~ P: biến ngẫu nhiên X được lấy mẫu từ phân phối P.
- Y ~ Q: biến ngẫu nhiên Y được lấy mẫu từ phân phối Q.
- $E_{X \sim P}[f(X)]$: kỳ vọng (trung bình) của hàm f(X) khi X được lấy mẫu từ P.
- $E_{Y \sim Q}[f(Y)]$: kỳ vọng (trung hình) của hàm f(Y) khi Y được lấy mẫu từ Q.
- $\|f\| \leq 1$: giới hạn các hàm f trong F sao cho chuẩn (norm) của chúng không vượt quá 1. Chuẩn $\|f\|$ thường được định nghĩa trong RKHS, đảm bảo các hàm được xem xét là bị chặn và phép so sánh là công bằng.
- sup: là supremum (giới hạn trên nhỏ nhất), tức là giá trị lớn nhất mà biểu thức có thể đạt được.

Trong thực tế, việc tìm sup trên F là khó khả thi. Do đó, MMD thường được triển khai trong không gian RKHS bằng cách sử dụng hàm kernel k(x, y), định nghĩa qua ánh xạ ϕ:

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_H$$

Trong đó:

- ϕ(x): là phép ánh xạ điểm x vào RKHS H.
- ⟨·,·⟩H:là tích vô hướng trong H.

Khi đó, khoảng cách MMD trong RKHS được biểu diễn là:

$$MMD(P,Q) = \left\| E_{X\sim P}\lceil \phi(X)\rceil - E_{Y\sim Q}\lceil \phi(Y)\rceil \right\|_H$$

Bình phương MMD thường được sử dụng hơn để đơn giản hóa công thức và tối ưu hiệu quả tính toán do sử dụng kernel trick [3].

$$MMD^2(P,Q) = \left\| E_{X\sim P}\lceil \phi(X)\rceil - E_{Y\sim Q}\lceil \phi(Y)\rceil \right\|_H^2$$

Khi ta khai triển công thức, công thức sau đó được mở rộng thành như sau:

$$MMD^2(P,Q) = E_{X,X\prime\sim P}\lceil k(X,X')\rceil + E_{Y,Y\prime\sim P}\lceil k(Y,Y')\rceil - 2E_{X\sim P,Y\sim Q}\lceil k(X,Y)\rceil$$

Công thức này cho phép chúng ta sử dụng các hàm kernel khác nhau tùy thuộc vào đặc điểm của dữ liệu, điều này mang lại khả năng tùy chỉnh và tính ứng dụng rộng. Đôi khi chúng ta không cần phải tính toán toàn bộ phân phối mà có thể dựa vào các mẫu dữ liệu từ P và Q. Điều này giúp ta xử lý được dữ liệu tốt trong thực tế đôi khi không tuân theo một phân phối cụ thể nào cả. Những thế mạnh này là nền tảng cho sự lựa chọn MMD cho phương pháp ShapeDD tới đây.

3.2.2. Shape Drift Detector (ShapeDD)

Phương pháp Shape Drift Detector là một phương pháp meta – statistic – based drift detector [2] được sử dụng để phát hiện sự trôi dạt dựa trên dữ liệu đầu vào bằng cách tận dụng khả năng tính toán của MMD để xác định sự thay đổi ở phân phối dữ liệu. Phương pháp này tập trung vào sự khác biệt giữa hai cửa sổ thời gian liên tục, giá trị khác biệt này được đánh giá bằng một đại lượng được gọi là độ lớn trôi dạt (Drift Magnitude).

$$\sigma_{d,l,D}(t) = d(D_{[t-2l,t-l]}, D_{[t-l,t]})$$

Trong đó, d là một hàm khoảng cách, với nhiều sự lựa chọn về d có thể áp dụng được vào giải thuật này làm cho bài toán được áp dụng rộng rãi. Ở đây, Maximum Mean Discrepancy (MMD) được lựa chọn để sử dụng làm hàm khoảng cách đó, ý tưởng ở đây là trong trường hợp xảy ra hiện tượng trôi dạt, σ không chỉ lấy giá trị lớn hơn 0 mà còn mang một giá trị có hình dạng đặc trưng được thể hiện rõ. ShapeDD Phương pháp được xây dựng dựa trên 4 stages dựa theo một sơ đồ được đề xuất cho các giải thuật phát hiện trôi dạt [2].

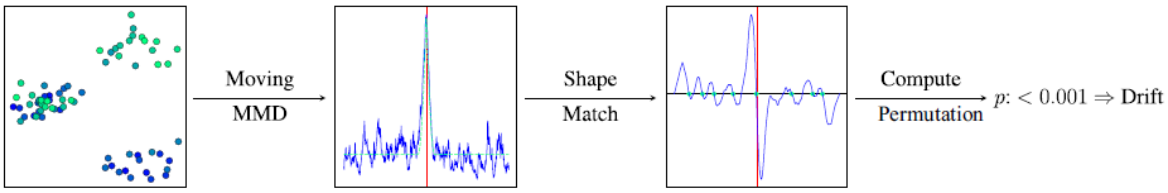

Figure 5 Cách phương pháp ShapeDD hoạt động

Stage 1: Thu thập dữ liệu

> Input: Luồng dữ liệu
> Output: Một (hoặc nhiều cửa sổ dữ liệu) ví dụ như là một cửa sổ dữ liệu dùng để tham chiếu và một cửa sổ dùng để chứa dữ liệu gần nhất

Với bước đầu tiên, có nhiều phương pháp thu thập dữ liệu có thể được lựa chọn. Tùy theo sự lựa chọn đó mà một hoặc hai phương pháp dùng một hay nhiều cửa sổ dữ liệu có thể được lựa chọn. Đa phần các phương pháp là dựa trên cửa sổ trượt. Đối với phương pháp ShapeDD, hai cửa sổ trượt liên tiếp là hướng tiếp cận được sử dụng để tính toán sự khác biệt giữa hai cửa sổ dữ liệu đó.

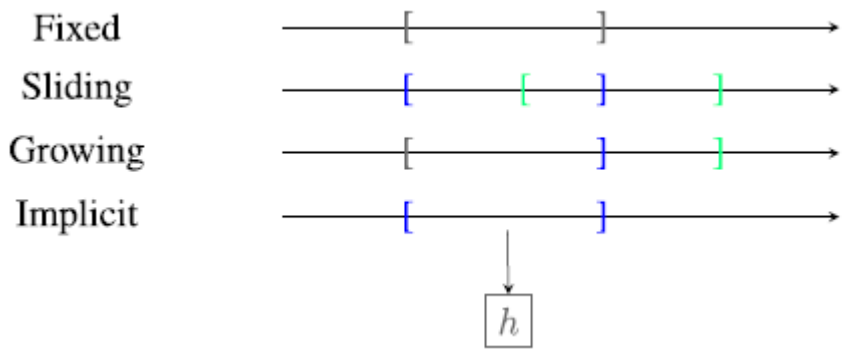

Figure 6 Các loại cửa sổ trượt

Stage 2: Xây dựng mô tả đặc trưng của dữ liệu

> Input: Một (hoặc nhiều) cửa sổ dữ liệu
> Output: Mô tả đặc trưng đã được xử lý của một (hay nhiều) cửa sổ dữ liệu

Mục tiêu của bước 2 là xây dựng được mô tả đặc trưng thu được từ phân phối dữ liệu của cửa sổ dữ liệu ở bước 1. Ở bước này có nhiều cách tiếp cận, ở đây ta chọn theo hướng kernel-based, điều mà ShapeDD sử dụng để xây dựng lên mô tả của dữ liệu sử dụng Gaussian RBF kernel.

Ở bước này, dữ liệu đầu vào sẽ được đưa qua hàm Gaussian RBF kernel để tạo ra một ma trận đo độ tương đồng của toàn bộ các cặp điểm trong tập dữ liệu.

Ví dụ với một tập dữ liệu mẫu gồm 8 điểm dữ liệu, mỗi 4 điểm dữ liệu sẽ mang phân phối khác nhau.

```
[0.2, 0.3],   # First distribution cluster
[-0.1, 0.4],
[0.3, 0.2],
[0.1, 0.3],
[9.8, 9.9],   # Second distribution cluster
[10.2, 10.1],
[9.9, 10.2],
[10.1, 9.8]
```

Figure 7 Tập dữ liệu mẫu

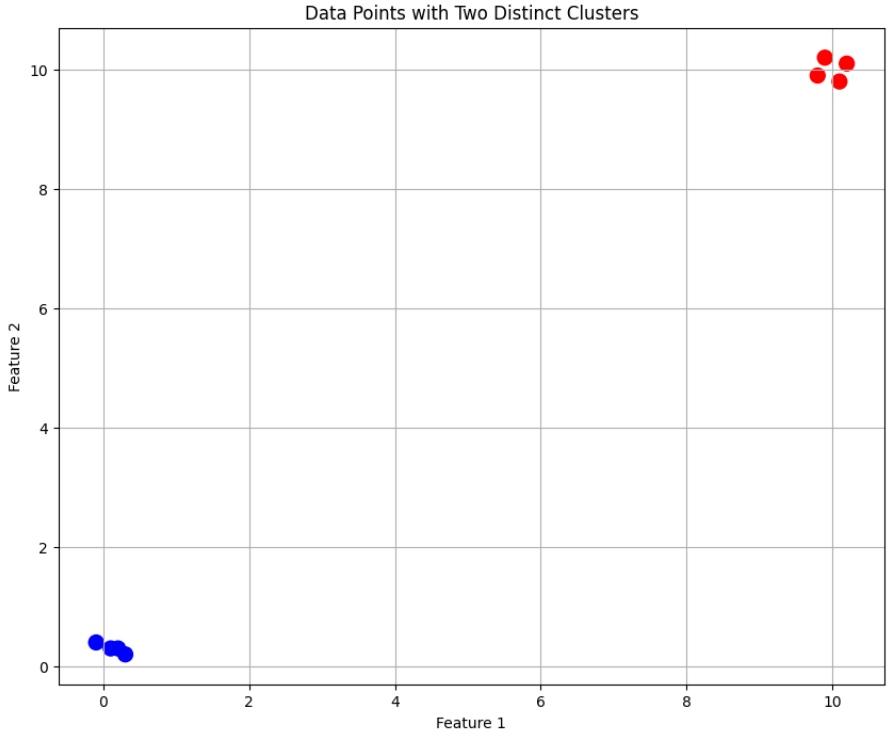

Figure 8 Phân phối của tập dữ liệu mẫu

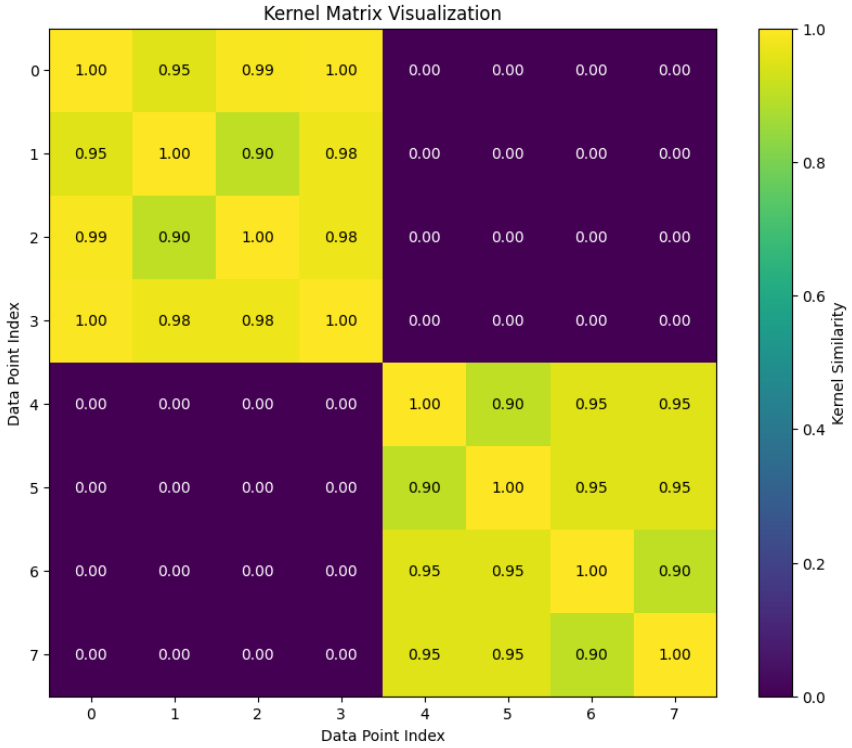

Figure 9 Ma trận kernel thể hiện sự tương đồng giữa các cặp dữ liệu

Bước này sẽ cho ta được ma trận kernel K thể hiện sự tương đồng giữa các cặp dữ liệu, ma trận này sẽ là đầu vào cho bước tiếp theo.

Stage 3: Tính toán sự khác biệt

> Input: Mô tả đặc trưng của dữ liệu
> Output: Độ khác nhau về mặt phân phối giữa hai cửa sổ dữ liệu

Mục tiêu ở bước này là tính toán điểm số của sự khác biệt, điều này có thể thực hiện bằng cách áp dụng tính toán MMD lên kết quả của bước trước đó. Kết quả ở bước này cho chúng ta được các điểm gọi là potential change point hay còn gọi là điểm tiềm năng khi mà dữ liệu bắt đầu xảy ra hiện tượng trôi dạt dựa trên giá trị độ lớn của sự trôi (drift magnitude) được nhắc đến ở trên.

Bước này chúng ta sẽ tạo ra một hàm gọi là hàm trọng số w có định nghĩa như sau:

$$w(t) = \begin{cases} \frac{-1}{l} & for \quad 0 \leq t < l \\ \frac{-1}{l} & for \quad l \leq t < 2l \end{cases}$$

Hàm này sẽ tạo một cửa sổ trượt với trọng số trái ngược nhau ở nửa đầu và nửa sau dựa theo kích thước của dữ liệu đầu vào và kích thước cửa sổ trượt ban đầu ta đặt ra. Kích thước của ma trận trọng số sẽ là:

((số lượng điểm dữ liệu) – 2*(kích thước cửa sổ), (số lượng điểm dữ liệu))

Lấy ví dụ với số lượng điểm dữ liệu ở trước đó là 8, kích thước cửa sổ trượt là 2, ta sẽ thu được ma trận W như sau:

[0.5 0.5 -0.5 -0.5 0   0   0   0]

[0   0.5 0.5 -0.5 -0.5 0   0   0]

[0   0   0.5 0.5 -0.5 -0.5 0   0]

[0 0 0 0.5 0.5 -0.5 -0.5 0]

Trong đó w(t) chính là [0.5 0.5 -0.5 -0.5].

Ma trận W này cùng với ma trận K sẽ được dùng để tính toán MMD bình phương để tính toán độ lớn sự khác biệt của hai phân phối.

Nhắc lại công thức MMD^2 ta có như sau:

$$MMD^2(P,Q) = E_{X,X' \sim P}[k(X,X')] + E_{Y,Y' \sim P}[k(Y,Y')] - 2E_{X \sim P, Y \sim Q}[k(X,Y)]$$

Tuy nhiên, vì không thể truy cập trực tiếp vào phân phối thực sự của dữ liệu trong thực tế, MMD được ước lượng dựa trên các mẫu dữ liệu quan sát được. Một công thức ước lượng sau dựa trên các điểm dữ liệu của hai phân phối có thể được sử dụng như sau.

$$MMD^2(P,Q) = \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} k(x_i, x_j) - 2\frac{1}{m.m} \sum_i \sum_j k(x_i, y_j) + \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} k(y_i, y_j) \quad [2]$$

Với kích thước cửa sổ dữ liệu mà ta chọn ban đầu (gọi là l1), công thức tương ứng với MMD sẽ là

$$MMD^2(P,Q) = \frac{1}{l1^2} \sum_i \sum_{j \neq i} k(x_i, x_j) - \frac{2}{l1^2} \sum_i \sum_j k(x_i, y_j) + \frac{1}{l1^2} \sum_i \sum_{j \neq i} k(y_i, y_j)$$

Công thức này đồng nghĩa với việc ta tính toán công thức sau.

$$MMD^2(P,Q) = WKW^T$$

Kết quả thu được sẽ là một mảng giá trị chứa độ khác biệt giữa hai phân phối dữ liệu, lấy số liệu từ ví dụ tính toán trước đó, ta thu được kết quả như sau

stat = [0.03066485, 0.4771088 , 1.94007367, 0.54758129]

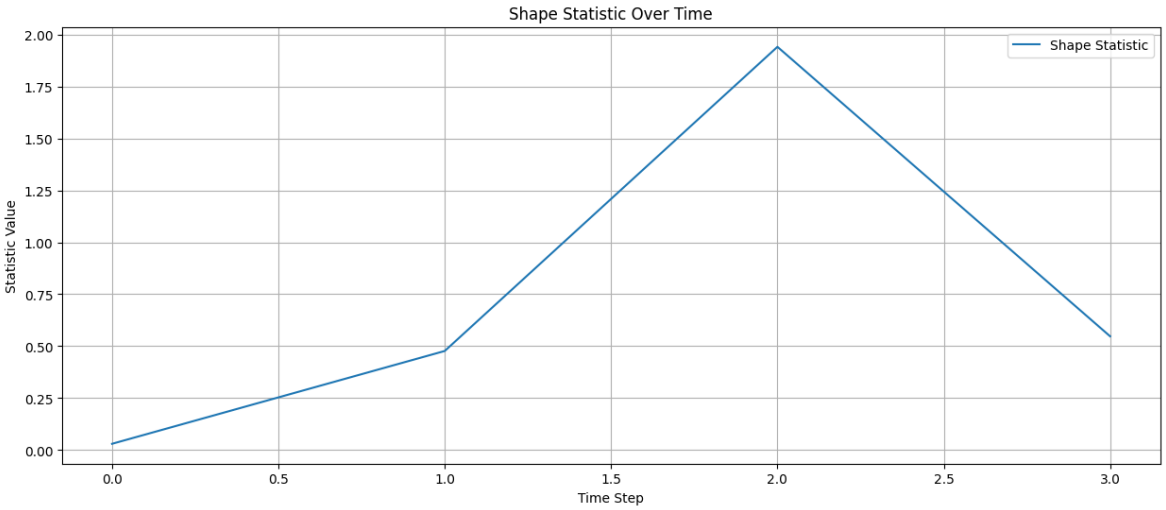

Figure 10 Giá trị kết quả khi ta tính toán MMD

Lý do tại sao ta sử dụng MMD^2 là vì:

- MMD^2 là một **khoảng cách bình phương** trong không gian đặc trưng của kernel (RKHS), luôn không âm và chỉ bằng 0 khi hai phân phối trùng nhau.
- Tính bình phương giúp tránh phải lấy căn (root), giữ thứ tự so sánh đúng, và cho dạng biểu thức ma trận/véctơ thuần túy–rõ ràng để tính nhanh.
- Trong ngữ cảnh sliding-window, ta chỉ cần so sánh giá trị MMD^2 tại các vị trí khác nhau để tìm "đỉnh" khác biệt, không cần độ lớn gốc của MMD.

Kết quả thu được sau cùng chính là độ khác biệt của hai phân phối giữa hai cửa sổ dữ liệu có kích thước l1.

Stage 4: Chuẩn hóa

> Input: Độ khác nhau về mặt phân phối giữa hai cửa sổ dữ liệu
> Output: Độ khác nhau về mặt phân phối giữa hai cửa sổ dữ liệu được chuẩn hóa

Sau khi thu được độ khác biệt phân phối ở bước trước, bước này sẽ giúp ta tính toán và chuẩn hóa lại giá trị thu được để có thể xác định được giá trị chính xác. Ở bước này, ta sẽ tiến hành sử dụng tích chập của kết quả thu được ở bước trước (độ khác nhau về mặt phân phối giữa hai cửa sổ dữ liệu) và hàm trọng số w, mục đích là để đo sự chênh lệch giữa mức độ drift trung bình ở "nửa trước" và "nửa sau" của một cửa sổ dữ liệu mới đồng thời giảm đi các giá trị nhiễu. Vì sau khi có chuỗi MMD^2 trượt, ta cần tìm những điểm mà độ khác biệt phân phối thay đổi đột ngột. Kết quả thu được khi ta thực hiện tích chập là:

shape_values =

[0.01533242, 0.25388682, 1.19325881, 0.98994066, -0.93480059,-1.24382748, -0.27379065]

Ở đây ta thấy được rằng có một thời điểm shape_values[i] và shape_values[i+1] trái dấu → đổi chiều từ dương sang âm hoặc ngược lại. Nó thể hiện một trường hợp drift điển hình:

- Trước drift: stat thấp (hai nửa cửa sổ dữ liệu có phân phối tương tự nhau)
- Khi bắt đầu drift: stat tăng dần (xuất hiện khác biệt)
- Sau drift: stat giảm dần (phân phối đã ổn định ở trạng thái mới)

Việc áp dụng tích chập sẽ tạo ra pattern:

- shape_values sẽ **tăng** khi ta đi qua vùng "trước → trong drift"
- shape_values sẽ **giảm** khi ta đi qua vùng "trong drift → sau drift"

Sau khi thực hiện tích chập, kết quả thu được sẽ được dùng đem đi để tính toán phép nhân từng phần tử (element-wise) giữa hai mảng liền kề. Đây là một mẹo toán học để phát hiện zero-crossing (điểm cắt qua 0). Dựa theo kết quả thu được, ta sẽ xác định được vị trí nơi mà có sự thay đổi mạnh về phân phối.

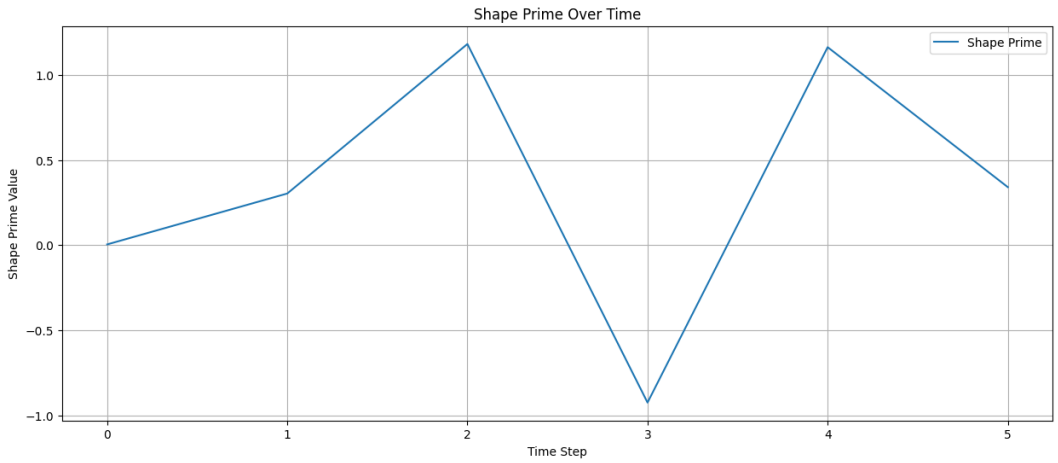

Figure 11 Giá trị sau cùng khi ta thực hiện zero-crossing để tìm ra điểm thay đổi

Dựa theo kết quả sau cùng thu được, những vị trí thay đổi dấu sẽ gọi là những điểm tiềm năng bị trôi dạt (potential change points). Những điểm này sẽ được tiến hành đánh giá lại bằng MMD với permutation test với kích thước cửa sổ lớn hơn.

Mục đích của bước này là để kiểm chứng các điểm nghi ngờ drift với ý nghĩa thống kê dùng p-value (sử dụng permutation test). Tốc độ thực thi của bước này chậm hơn, nhưng chỉ chạy trên các điểm nghi ngờ trôi dạt.

Một ma trận trọng số mới sẽ được tạo ra để thực thi chuyện này. Lấy ví dụ ta có cửa sổ l1 = 2 ở lần ví dụ trước, lần này ta xác định một cửa sổ mới l2, lấy ví dụ giá trị bằng 3.

Khi đó ma trận trọng số mới được tạo ra là

$$w = [0.5, 0.5, -0.333, -0.333, -0.333]$$

Kết hợp với hoán vị, ta thu được ma trận trọng số hoàn chỉnh như sau.

```
[0.5,   0.5,   -0.333, -0.333, -0.333],    # Hàng 0: Vector gốc
[-0.333, 0.5,   0.5,   -0.333, -0.333],    # Hàng 1: Hoán vị 1
[0.5,  -0.333, -0.333,  0.5,   -0.333],    # Hàng 2: Hoán vị 2
```

[-0.333, -0.333, 0.5, 0.5, -0.333], *# Hàng 3: Hoán vị 3*

[0.5, -0.333, 0.5, -0.333, -0.333] *# Hàng 4: Hoán vị 4*

Áp dụng ma trận này với ví dụ trên, ta thu được kết quả là.

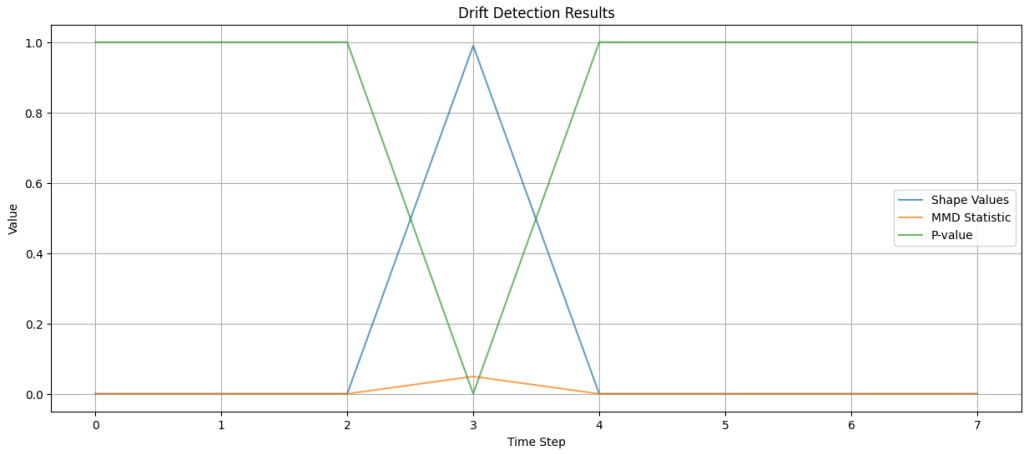

Figure 12 Kết quả khi ta thực thi lại MMD xung quanh các điểm trôi dạt tiềm năng

Có thể nhận thấy được rằng giá trị p-value nhỏ, điều đó thể hiện rằng sự trôi dạt đã xảy ra bắt đầu tại vị trí thứ 3, ứng với dữ liệu ví dụ ban đầu. Bây giờ ta sẽ tiến hành thực nghiệm trên tập dữ liệu tổng hợp (synthetic dataset).

Chương 4

Ứng dụng ShapeDD trong việc phát hiện trôi dạt của dữ liệu tổng hợp

4.1. Xây dựng tập dữ liệu

Để xác định được độ chính xác của giải thuật đối với tập dữ liệu lớn, ta xây dựng một bộ dữ liệu tổng hợp **(synthetic dataset),** đây là một tập dữ liệu được tạo ra để xác định độ chính xác của mô hình trong việc phát hiện trôi dạt, vì đây là dữ liệu tự tạo nên ta có thể kiểm soát được các tham số liên quan đến sự trôi dạt như thời điểm xảy ra trôi dạt, số lần trôi dạt và mức độ trôi dạt,... Đề cương này sẽ tiến hành áp dụng phương pháp lên 2 tập dữ liệu khác nhau, một loại là dạng abrupt drift, nơi mà phân phối thay đổi đột ngột tại một vị trí bất kỳ nào đó, tập dữ liệu thứ hai là incremental drift, nơi mà phân phối của tập dữ liệu thay đổi theo thời gian.

Tập dữ liệu abrupt drift: Dữ liệu được lấy mẫu đồng đều trong kich thước 1 đơn vị, sự trôi dạt (drift) được đưa vào bằng cách dịch chuyển phân phối của dữ liệu theo đường chéo. Việc độ trôi dạt lớn đến mức nào và số lần xảy ra trôi dạt là có thể kiểm soát được thông qua các tham số của việc tạo ra dữ liệu, nhưng vị trí xảy ra trôi dạt là không kiểm soát được.

Tập dữ liệu incremental drift: Dữ liệu được lấy mẫu từ phân phối đồng đều (hoặc Gaussian) trong không gian đa chiều. Sự trôi dạt được đưa vào bằng cách dịch chuyển dần dần các chiều đầu tiên của dữ liệu theo thời gian, quá trình trôi dạt có thể theo các dạng như: Linear (Tăng đều theo thời gian), Sigmoid (Tăng chậm ban đầu, nhanh ở giữa, chậm lại ở cuối (dạng chữ S)) và Exponential (Tăng nhanh dần theo hàm mũ).

Cả hai tập dữ liệu đều được tạo ra với 10000 điểm dữ liệu, với độ trôi đều là 0.5. Đối với tập dữ liệu abrupt drift, có 10 điểm trôi ngẫu nhiên xảy ra và còn tập incremental drift chỉ có 1 điểm trôi xảy ra trong toàn bộ tập dữ liệu.

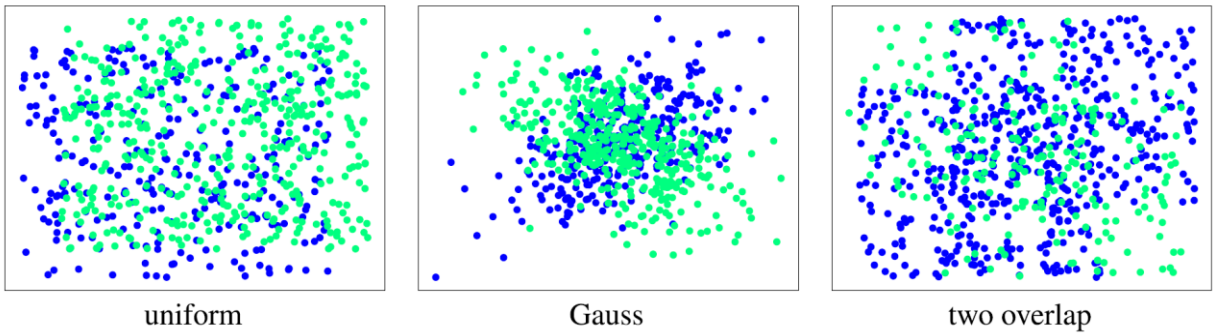

Figure 13 Các loại phân phối của dữ liệu tổng hợp

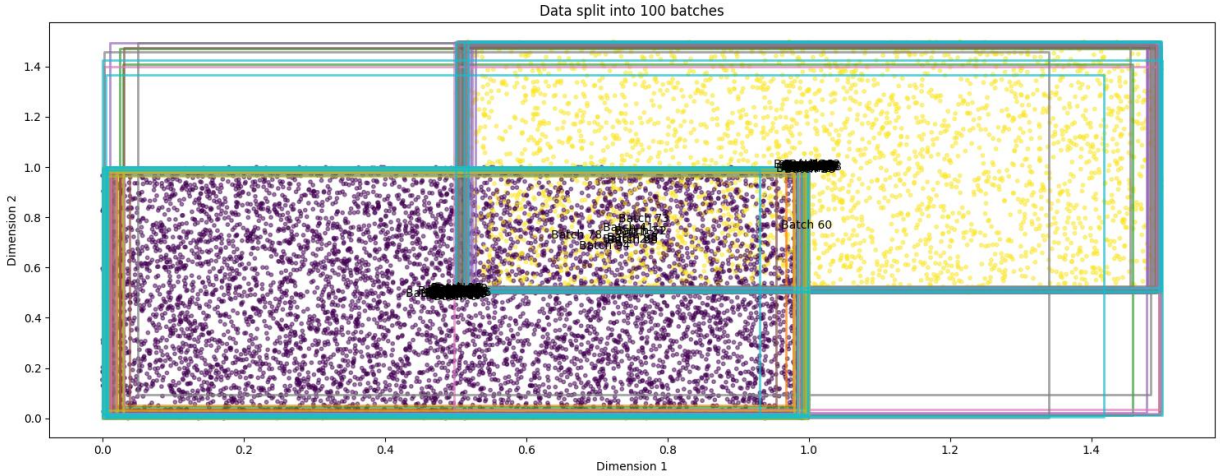

Figure 14 Phân phối dữ liệu bị abrupt drift được tạo ra

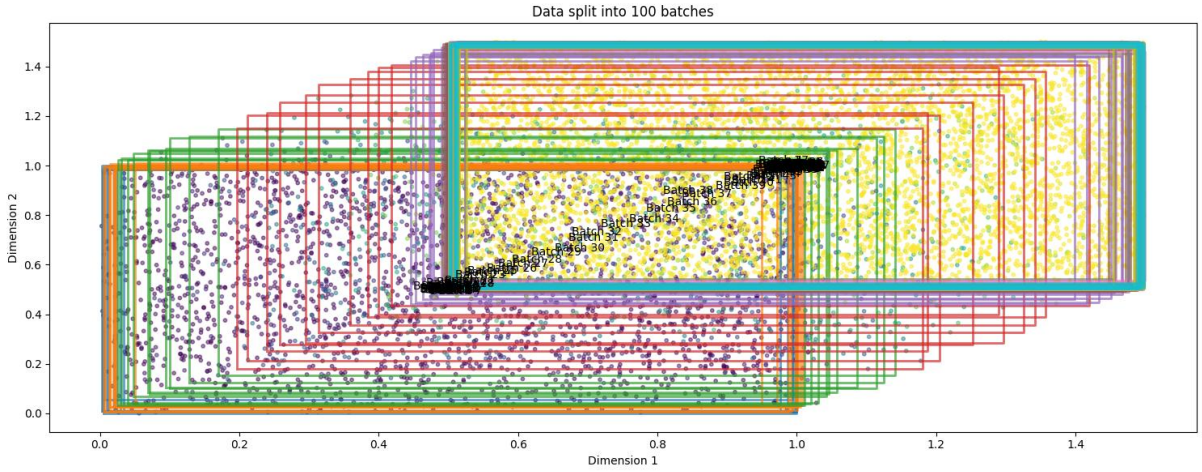

Figure 15 Phân phối dữ liệu bị incremental drift được tạo ra

4.2. Kết quả thực nghiệm ban đầu

4.2.1. Tập dữ liệu abrupt drift

Sau khi thực hiện việc xây dựng dữ liệu, dữ liệu được tạo ra sau đó sẽ được đưa vào ShapeDD để phát hiện sự thay đổi về phân phối. Kích thước của cửa sổ dữ liệu ban đầu được đưa vào là 50 điểm dữ liệu, với nửa trước và nửa sau là [25 25].

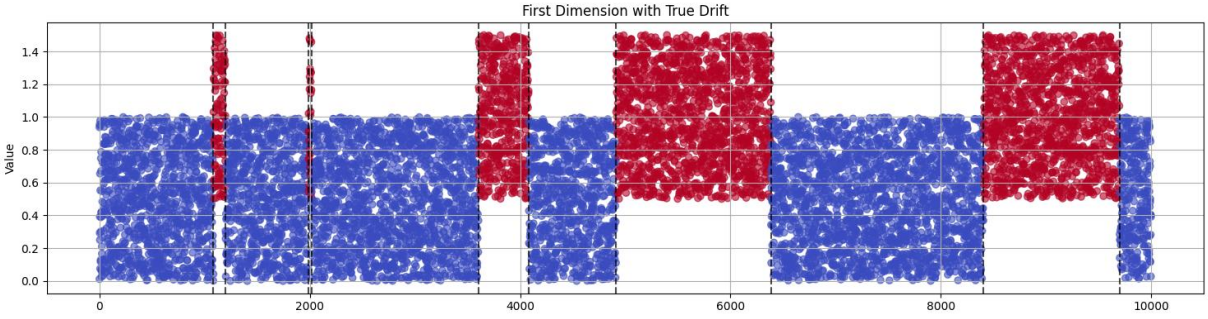

Figure 16 Dữ liệu được tạo ra theo uniform thể hiện cho các vị trí xảy ra abrupt drift

Theo như tập dữ liệu được tạo ra, có 10 vị trí xảy ra sự thay đổi về phân phối và các vị trí đều là xảy ra abrupt drift, vì đây là tập dữ liệu tự tạo, việc kiểm soát cái vị trí thay đổi đều có thể được đánh dấu lại để có thể dùng để so sánh với kết quả dự đoán của mô hình.

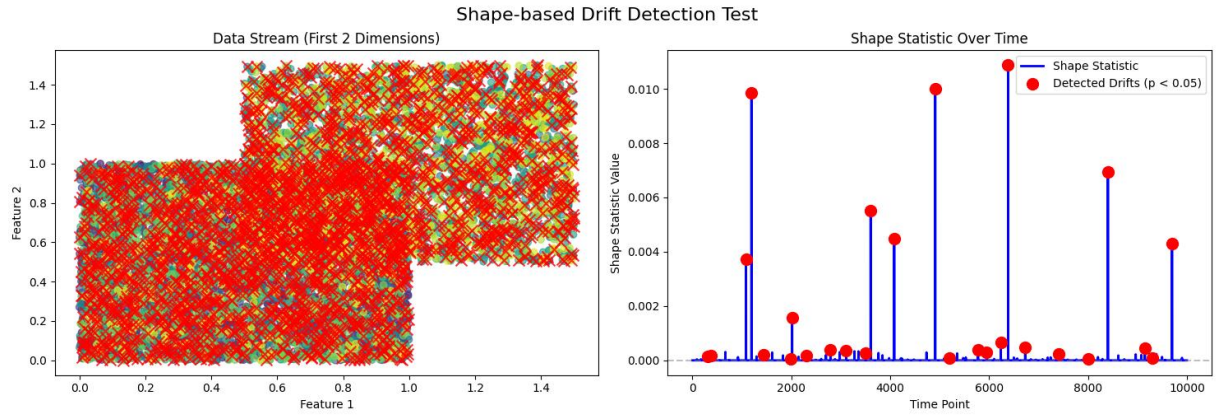

Figure 17 Kết quả dự đoán của ShapeDD dựa trên MMD (Stage 2)

Sau khi áp dụng MMD để tính toán shape statistic từ tập dữ liệu, ta thu được kết quả như hình trên, những giá trị thống kê cao thể hiện cho những sự thay đổi lớn mà ta sẽ gọi là những điểm tiềm năng xuất hiện sự trôi dạt. Sau khi xác định được những điểm đó, MMD sẽ được thực hiện lại xung quanh những điểm đó với permutation là 2500 để đảm bảo là xung quanh những điểm này thực sự có sự thay đổi về phân phối.

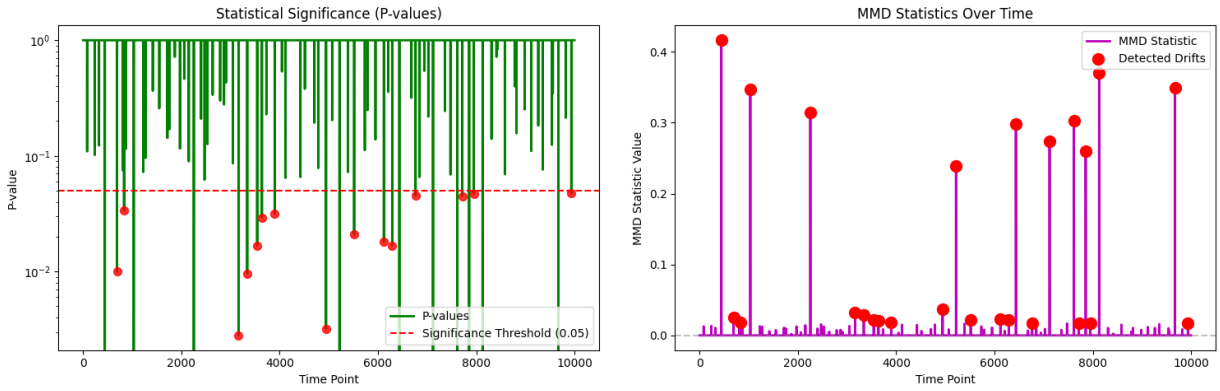

Figure 18 Kết quả thu được khi chạy kiểm tra lại với MMD đối với các điểm tiềm năng

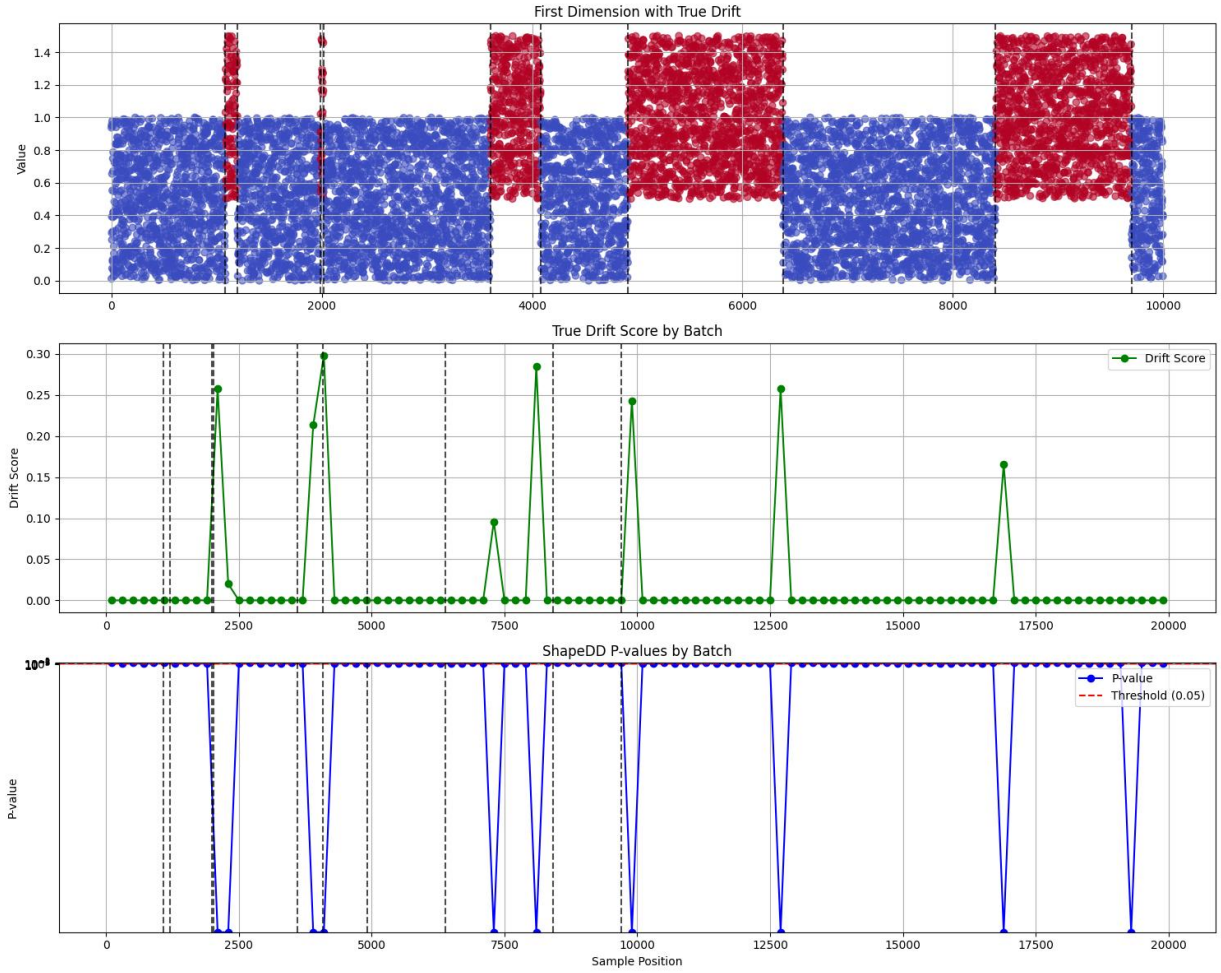

Figure 19 Kết quả khi chọn ra những p-value nhỏ nhất để tránh nhiễu theo từng batch dữ liệu

Sử dụng p-value trong thống kê, ta xác định được những điểm trôi dạt xảy ra khi giá trị p-value đạt dưới ngưỡng một ngưỡng mà chúng ta đề ra.

Vì đây là tập dữ liệu mang tính chất abrupt drift, điều này làm cho phân phối của dữ liệu thay đổi đột ngột, điều này giúp cho phương pháp ShapeDD dễ dàng phát hiện ra sự thay đổi phân phối do việc sử dụng chiến thuật cửa sổ trượt liên tiếp do phân phối gần như thay đổi ngay lập tức và khác nhau.

4.2.2. Tập dữ liệu incremental drift

Đối với tập dữ liệu incremental drift, khi ta giữ nguyên kích thước cửa sổ đầu vào khi đo với tập dữ liệu abrupt drift, kết quả của ShapeDD cung cấp lại không được tốt lắm.

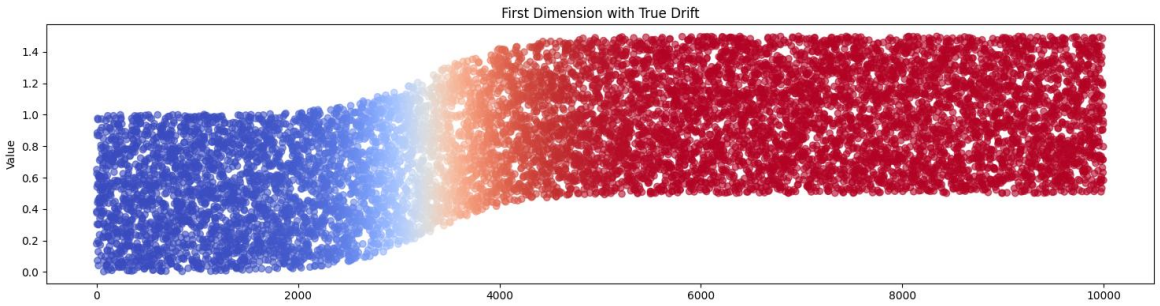

Figure 20 Phân phối của dữ liệu khi xảy ra trôi dạt theo incremental

Kết quả thu được khi ta áp dụng cùng kích thước cửa sổ khi ta thực hiện với tập dữ liệu abrupt drift lên tập dữ liệu incremental drift.

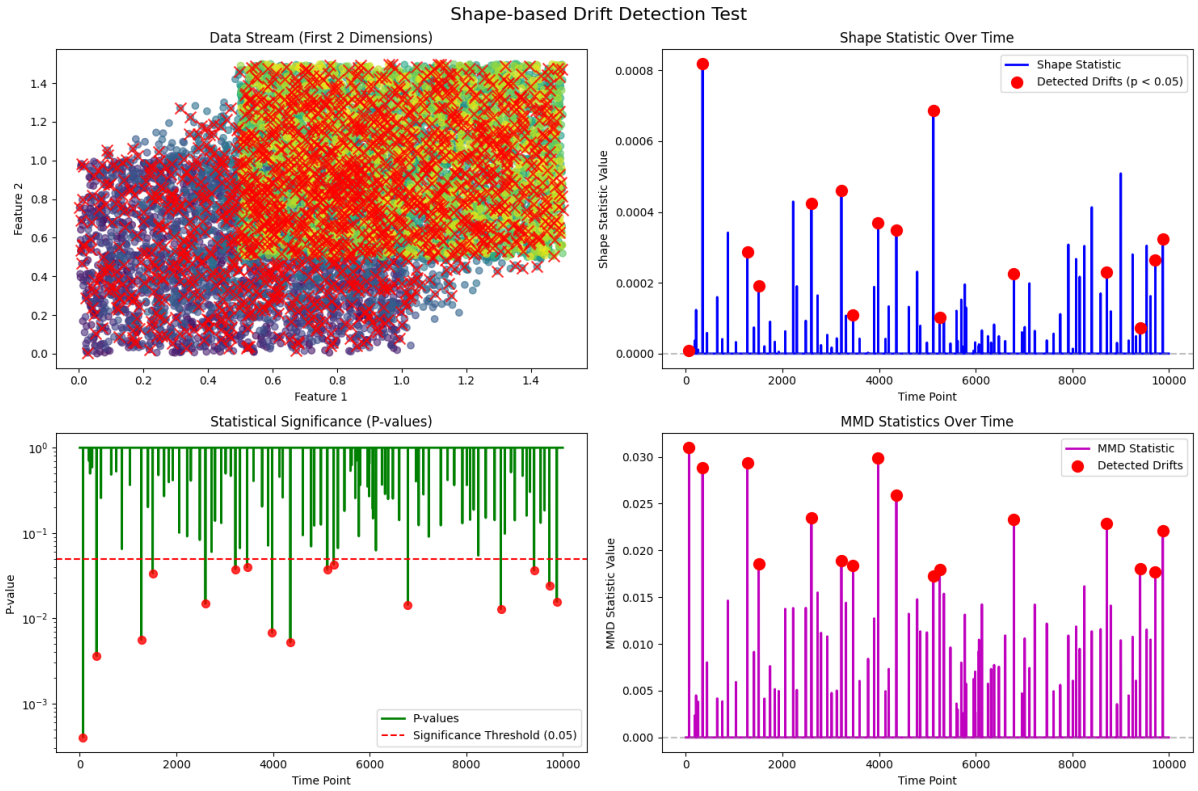

Figure 21 Kết quả thực hiện với kích thước cửa sổ nhỏ

Dựa theo kết quả trên, tại vị trí mà phân phối thay đổi rõ ràng nhất, ShapeDD có thể phát hiện được điểm tiềm trôi dạt lớn nhất, tuy nhiên vẫn còn nhiễu và cho kết quả chưa chính xác. Điều này có thể do cách ta lựa chọn về cửa sổ ở (stage 1)[2] có thể làm ảnh hưởng đến kết quả và làm cho việc bỏ lỡ mất một số drift events hoặc do dữ liệu tạo ra chưa mô phỏng được chính xác quá trình incremental drift xảy ra.

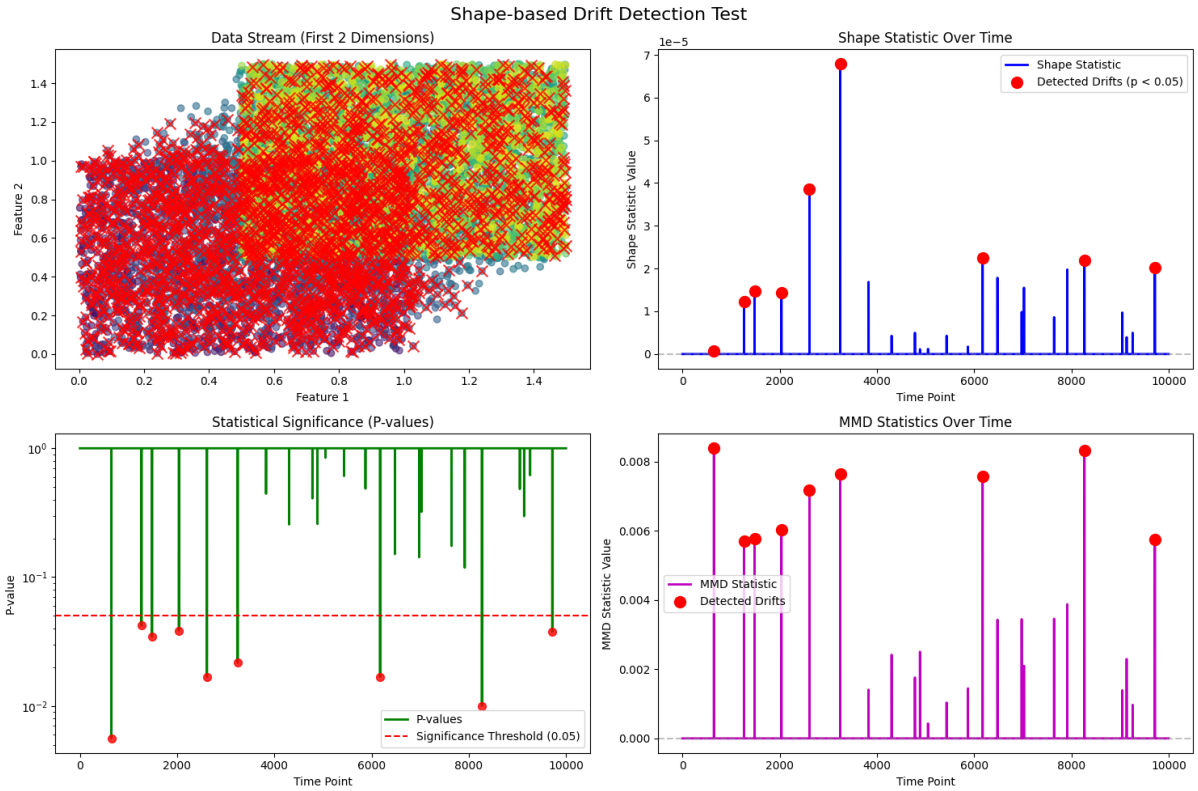

Figure 22 Kết quả thu được khi thay đổi kích thước cửa sổ xử lý dữ liệu

Khi ta thay đổi về kích thước cửa sổ đầu vào (tăng lên gấp đôi kích thước cửa sổ ban đầu) và thực hiện lại với cùng phân phối dữ liệu ban đầu, phương pháp có thể phát hiện ra được vị trí các điểm thay đổi chính xác hơn và ít thấy được sự nhiễu. Tuy nhiên, phương pháp cũng chỉ có thể phát hiện được ở những vị trí mà phân phối thay đổi lớn như ở đoạn giữa của quá trình thay đổi phân phối nơi mà sự chênh lệch là rõ ràng nhất.

Chương 5

Tổng kết

5.1. Đánh giá kết quả thực nghiệm

Từ những kết quả thu được, đề tài tổng kết được những đánh giá về phương pháp đối với tập dữ liệu synthetic như sau:

- **Đối với dữ liệu có abrupt drift**: Phương pháp ShapeDD cho thấy khả năng phát hiện trôi dạt hiệu quả, ngay cả khi sự thay đổi giữa hai cửa sổ dữ liệu là nhỏ. Đánh giá dựa trên giá trị p–value thu được từ phép kiểm định Maximum Mean Discrepancy (MMD) cho thấy các điểm trôi dạt được phát hiện có sự tương quan cao với các điểm trôi dạt thực tế trong tập dữ liệu.
- **Đối với dữ liệu có incremental drift**: ShapeDD hoạt động chưa thực sự ổn định. Khi sử dụng kích thước cửa sổ nhỏ, độ chính xác giảm rõ rệt và kết quả có nhiều nhiễu. Việc tăng kích thước cửa sổ giúp cải thiện độ chính xác, nhưng vẫn chưa triệt tiêu được nhiễu. Nguyên nhân có thể đến từ hai yếu tố: (1) lựa chọn kích thước cửa sổ chưa phù hợp với tính chất thay đổi dần của dữ liệu, hoặc (2) mô phỏng dữ liệu synthetic chưa đủ phản ánh bản chất của incremental drift một cách mượt mà và thực tế.

Từ những đánh giá trên, đề tài đề xuất ra thêm một số cải tiến thêm như sau:

- Tối ưu hóa chiến lược lựa chọn kích thước cửa sổ (window size) thích ứng theo tốc độ thay đổi của dữ liệu.
- Cân nhắc kết hợp với các phương pháp làm trơn (smoothing) hoặc giảm nhiễu đầu ra.
- Cải thiện quy trình sinh dữ liệu synthetic cho incremental drift để phản ánh tốt hơn các đặc trưng của quá trình trôi dạt liên tục.

- Sử dụng Phương pháp tổng hợp (Ensemble method): Các phương pháp tổng hợp (Ensemble Methods - EMs) kết hợp nhiều mô hình khác nhau nhằm nâng cao độ chính xác và tính ổn định trong việc phát hiện concept drift. Nhiều nghiên cứu đã chứng minh rằng các phương pháp này đạt hiệu suất cao trên nhiều loại dữ liệu khác nhau [14].

5.2. Những công việc đã thực hiện

Cho đến hiện tại, đề cương này đã thực hiện được những công việc sau:

- Tìm hiểu các bài báo liên quan đề tài.
- Tìm hiểu các giải thuật phát hiện trôi dạt, phân loại và các thức hoạt động của các nhóm giải thuật.
- Tìm hiểu lý thuyết về Maximum Mean Discrepancy và hiểu cách phương pháp Shape Drift Detector hoạt động.
- Xây dựng được bộ dữ liệu tổng hợp (synthetic dataset) liên quan đến các trường hợp xảy ra trôi dạt phổ biến như abrupt hay là incremental.
- Triển khai ShapeDD, đánh giá độ chính xác của phương pháp trên các tập dữ liệu tổng hợp và dựa theo cách chọn kích thước cửa sổ dữ liệu.

5.3. Kế hoạch trong giai đoạn luận văn

Từ những gì đã tìm hiểu được, đề cương này đề xuất hướng phát triển tiếp theo của đề tài trong giai đoạn luận văn như sau:

1. **Tiếp tục đánh giá và cải thiện đối phương pháp đối với các loại trôi dạt khác nhau (3 tuần):**
   - Tiếp tục thực hiện đánh giá hiệu năng của phương pháp ShapeDD trên các loại concept drift khác nhau, bao gồm: abrupt drift, incremental drift, gradual drift và recurring drift.
   - Phân tích sâu các yếu tố ảnh hưởng đến độ chính xác của mô hình trong từng trường hợp cụ thể.

- Tối ưu các tham số như kích thước cửa sổ (window size), độ trễ phát hiện drift (detection delay), và ngưỡng p-value để nâng cao khả năng phát hiện.
- Nghiên cứu đến phương pháp Ensemble Methods, nghiêng cứu việc kết hợp với nhiều mô hình khác hoặc lựa chọn một phương pháp khác mang lại kết quả tốt hơn.

2. **Xây dựng hệ thống giám sát và cập nhật mô hình khi xảy ra concept drift (5 tuần):**
   - Thiết kế hệ thống theo dõi dữ liệu đầu vào và giám sát sự thay đổi phân phối theo thời gian thực.
   - Phát triển cơ chế phát hiện drift và tự động phản hồi bằng cách cập nhật mô hình học máy phù hợp.
   - Thử nghiệm các chiến lược cập nhật mô hình như: huấn luyện lại toàn bộ, học gia tăng (incremental learning), hoặc chuyển mô hình về trạng thái cảnh báo.
   - Tích hợp logging và hệ thống cảnh báo để hỗ trợ việc theo dõi và kiểm tra lại kết quả.
   - Đảm bảo tính ổn định và hiệu quả của hệ thống trong điều kiện dữ liệu thực hoặc mô phỏng thời gian thực.
3. **Ứng dụng mô hình vào trong ứng dụng thực tế (3 tuần):**
   - Triển khai mô hình đã được tối ưu vào một hệ thống hoặc ứng dụng thực tế, ví dụ: nhận diện trạng thái cảm biến, phát hiện gian lận, hoặc giám sát thiết bị.
   - Đánh giá hiệu năng của mô hình trong môi trường thực, bao gồm: độ chính xác, tốc độ phản ứng, khả năng thích nghi với dữ liệu thay đổi liên tục.
   - Ghi nhận phản hồi và điều chỉnh mô hình, cũng như hệ thống xử lý dữ liệu đầu vào để cải thiện hiệu quả hoạt động.
   - Kiểm tra khả năng tương thích với hệ thống hiện có và tối ưu hiệu suất hoạt động trong môi trường triển khai.
4. **Đánh giá và viết báo cáo (3 tuần):**

- Tổng hợp kết quả đánh giá mô hình trên cả dữ liệu mô phỏng (synthetic) và dữ liệu thực tế.
- Phân tích các chỉ số đánh giá như độ chính xác, độ trễ, tỷ lệ phát hiện sai, số lần cập nhật mô hình, v.v.
- So sánh mô hình với các baseline để làm rõ điểm mạnh và điểm cần cải thiện.
- Viết báo cáo tổng kết kỹ thuật, trình bày phương pháp, quá trình thực hiện, kết quả và đề xuất hướng phát triển tiếp theo.
- Chuẩn bị tài liệu trình bày (slide, báo cáo hoặc poster) nếu cần cho buổi báo cáo nội bộ hoặc hội thảo.

Tài liệu tham khảo

[1] ScienceDirect, “Concept drift: A comparative study on online machine learning techniques for network traffic streams analysis,” [Online]. Available: https://www.sciencedirect.com/topics/computer-science/concept-drift.

[2] Hinder F, Vaquet V and Hammer B (2024) One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: detecting concept drift. *Front. Artif. Intell.* 7:1330257. doi: 10.3389/frai.2024.1330257

[3] Maximum Mean Discrepancy (MMD) in Machine Learning - Mixed random variables. (2019, March 8). Mixed Random Variables. https://www.onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning.html

[4] N. Jourdan, L. Longard, T. Biegel, and J. Metternich, “Machine learning for intelligent maintenance and quality control: A review of existing datasets and corresponding use cases,” vol. 2, 2021.

[5] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, “Machine learning in manufacturing: advantages, challenges, and applications,” *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016.

[6] M. Chui, B. Hall, A. Singla, and A. Sukharevsky, “The state of AI in 2021,” Tech. Rep., McKinsey & Company, 2021.

[7] M. Chui, B. Hall, H. Mayhew, and A. Singla, “The state of AI in 2022 - and half decade in review,” Tech. Rep., QuantumBlack by McKinsey, 2022.

[8] X. Wu, M. El-Shamouty, and P. Wagner, “Dependable AI: Using AI in safety-critical industrial applications,” Tech. Rep., Fraunhofer Institute for Manufacturing Engineering and Automation (IPA), 2021.

[9] S. Tripathi, D. Muhr, M. Brunner, H. Jodlbauer, M. Dehmer, and F. Emmert-Streib, “Ensuring the robustness and reliability of data-driven knowledge discovery models in production and manufacturing,” *Frontiers in Artificial Intelligence*, vol. 4, p. 22, 2021.

[10] N. Jourdan, T. Bayer, T. Biegel, and J. Metternich, “Handling concept drift in deep learning applications for process monitoring,” *Procedia CIRP*, vol. 120, pp. 33–38, 2023.
[11] T. Biegel et al., “Combining process monitoring with text mining for anomaly detection in discrete manufacturing,” *SSRN*, 2022.
[12] K. Ramakrishnan, D. Preuveneers, and Y. Berbers, “Enabling self-learning in dynamic and open IoT environments,” *Procedia Comput. Sci.*, vol. 32, pp. 207–214, 2014.
[13] J. C. Schlimmer and R. H. Granger, “Incremental learning from noisy data,” *Mach. Learn.*, vol. 1, no. 3, pp. 317–354, 1986.
[14] Hovakimyan, Gurgen & Bravo, Jorge. (2024). Evolving Strategies in Machine Learning: A Systematic Review of Concept Drift Detection. Information. 15. 1-24. 10.3390/info15120786.