PHÁT HIỆN BẤT THƯỜNG TRÊN CHUỖI THỜI GIAN DẠNG LUỒNG: MỘT GIẢI THUẬT DỰA VÀO PHÂN ĐOẠN

Huỳnh Thị Thu Thủy[1], Dương Tuấn Anh[1,2,*]

[1] Trường Đại học Bách Khoa, Đại học Quốc Gia TP.HCM
[2] Khoa Công nghệ thông tin, Trường Đại học Ngoại ngữ - Tin học TP.HCM
8141217@hcmut.edu.vn, anhdt@huflit.edu.vn

TÓM TẮT— Phát hiện chuỗi con bất thường trên chuỗi thời gian dạng luồng là một vấn đề quan trọng nhưng chưa được giải quyết đúng mức. Trong bài báo này, chúng tôi đề xuất một cách tiếp cận mới kết hợp phân đoạn và gom cụm để phát hiện bất thường trên chuỗi thời gian dạng luồng. Khi phân đoạn, phương pháp điểm cực trị quan trọng được dùng để rút trích các chuỗi con từ một chuỗi thời gian. Khi gom cụm, một giải thuật gom cụm gia tăng được sử dụng để gom cụm các chuỗi con đã được rút trích. Một phân đoạn cục bộ của chuỗi thời gian dạng luồng sẽ được lưu trong một bộ đệm xoay vòng, từ đấy các hệ số bất thường của các chuỗi con sẽ được tính một cách hữu hiệu. Ngoài ra, phương pháp đề xuất còn áp dụng chiến lược cập nhật trì hoãn để xử lý dữ liệu chuỗi thời gian dạng luồng một cách hợp lý hơn. Kết quả thực nghiệm trên sáu bộ dữ liệu mẫu cho thấy phương pháp đề xuất hữu hiệu hơn rất nhiều khi so với giải thuật BFHS, một phiên bản mở rộng của giải thuật HOT SAX, trong khi các chuỗi con bất thường được phát hiện bởi cả hai phương pháp này thì giống hệt nhau khi thực nghiệm trên các bộ dữ liệu mẫu.

Từ khóa— phát hiện bất thường, chuỗi thời gian dạng luồng, gom cụm, phân đoạn.

I. GIỚI THIỆU

Một chuỗi thời gian dạng luồng là một chuỗi không có giới hạn của những điểm dữ liệu mà trong đó các điểm dữ liệu mới đến liên tục theo thời gian. Gần đây, phát hiện bất thường trên chuỗi thời gian dạng luồng đã xuất hiện như một đề tài sôi động vì có nhiều ứng dụng cần xử lý công tác này theo thời gian thực. Vài thí dụ về những ứng dụng này có thể kể như: giám sát chuỗi dữ liệu thiên văn Tia Gamma [1], phát hiện những đoạn bất thường trên dữ liệu điện tâm đồ (ECG) dạng luồng đến từ bệnh nhân [2], phát hiện những mẫu bất thường trên dữ liệu GPS của xe cộ [3]. Và với sự gia tăng của việc kết nối cảm biến thời gian thực, phát hiện các mẫu bất thường trên dữ liệu cảm biến dạng luồng (stream sensor data) đã trở nên rất quan trọng [4].

Khi xử lý chuỗi thời gian tĩnh (static time series), tất cả các điểm dữ liệu của chuỗi thời gian đều đã có sẵn và được lưu trong bộ nhớ. So sánh với chuỗi thời gian tĩnh, chuỗi thời gian dạng luồng có những đặc điểm như sau: (1) Các điểm dữ liệu thường xuyên thêm vào chuỗi thời gian dạng luồng, (2) Kích thước của một chuỗi thời gian dạng luồng hầu như không có giới hạn, (3) do sự cập nhật dữ liệu liên tục, rất khó có thể lưu tất cả dữ liệu trong bộ nhớ chính hoặc trong đĩa, do đó cần có những giải thuật làm việc theo cách duyệt qua dữ liệu chỉ một lần (one-pass algorithm) để có thể đáp ứng theo thời gian thực. Do những đặc điểm như vậy, các phương pháp áp dụng cho chuỗi thời gian tĩnh không dễ cải tiến để làm việc được trong bối cảnh dữ liệu luồng.

Phát hiện bất thường trên chuỗi thời gian dạng luồng thường khó hơn nhiều so với bài toán tìm kiếm tương tự (similarity search) trên chuỗi thời gian dạng luồng. Hai thách thức chính của bài toán phát hiện bất thường trên chuỗi thời gian dạng luồng là làm cách nào để xác định chiều dài của chuỗi con bất thường và làm cách nào để cập nhật gia tăng chuỗi con bất thường nhất mỗi khi có một điểm dữ liệu mới tới.

Đóng góp chính của bài báo này được nêu ra như sau: trong công trình trước đây của nhóm (Thủy và các cộng sự, 2018 [5]), chúng tôi đã đề xuất EP-ILeader, một giải thuật hữu hiệu để phát hiện bất thường trên chuỗi thời gian tĩnh với độ đo Euclid. Trong giải thuật EP-ILeader, chúng tôi sử dụng một phương pháp phân đoạn để phân chia một chuỗi thời gian thành những chuỗi con. Sau đó một giải thuật gom cụm gia tăng, có tên I-Leader, được thực thi để gom cụm các chuỗi con và các cụm sẽ được dùng để phát hiện chuỗi con bất thường nhất. Trong công trình này, chúng tôi mở rộng giải thuật EP-ILeader để phát hiện bất thường trong một bối cảnh thách thức hơn: chuỗi thời gian dạng luồng. Giải thuật mới này, có tên SEP-Ileader, có thể phát hiện chuỗi con bất thường ngay khi chuỗi con này xuất hiện trên chuỗi thời gian dạng luồng. Phương pháp đề xuất vận dụng tính chất trực tuyến của giải thuật phân đoạn chuỗi thời gian và tính chất gia tăng của giải thuật gom cụm I-Leader [6].

Chúng tôi so sánh hiệu năng của giải thuật SEP-ILeader với giải thuật BFHS. Giải thuật BFHS là một phiên bản mở rộng của giải thuật HOT SAX [7] để phát hiện bất thường trên chuỗi thời gian dạng luồng. Kết qủa thực nghiệm cho thấy SEP-ILeader thực thi nhanh hơn giải thuật BFHS rất nhiều trong khi đem lại cùng chuỗi con bất thường được phát hiện. Do đó, SEP-ILeader rất thích hợp cho những ứng dụng cần sự đáp ứng thời gian thực.

* Coressponding Author

Phần còn lại của bài báo được tổ chức như sau: mục II giới thiệu một số khái niệm liên quan, mục III mô tả các công trình liên quan, mục IV giới thiệu giải thuật đề xuất SEP-ILeader để phát hiện bất thường trên chuỗi thời gian dạng luồng; kết quả thực nghiệm về giải thuật SEP-ILeader được trình bày ở mục V; mục VI nêu một vài kết luận và các hướng phát triển của đề tài.

II. CÁC KHÁI NIỆM

A. CÁC ĐỊNH NGHĨA

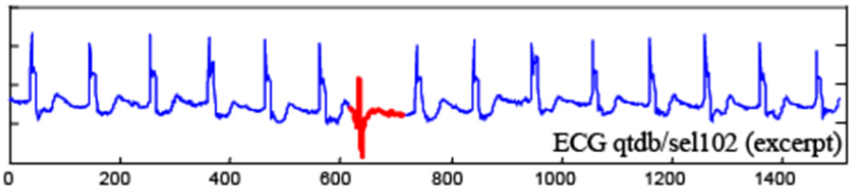
Phát hiện bất thường trên chuỗi thời gian là tìm kiếm chuỗi con bất thường nhất (most unusual subsequence, discord) trên một chuỗi thời gian. Chuỗi con bất thường nhất là chuỗi con mà khác biệt nhiều nhất đối với chuỗi con mà tương tự với nó nhất. Tuy nhiên, đôi khi những chuỗi con trùng khớp với nhau nhất có thể phủ lấp lên nhau (overlap). Những chuỗi con như vậy được gọi là những chuỗi con trùng khớp tầm thường (trivial matches). Để có thể phát hiện những chuỗi con bất thường có ý nghĩa, chúng ta nên loại bỏ những chuỗi con trùng khớp tầm thường.

Một *chuỗi thời gian* $T = (t_1, t_2,..., t_m)$ là một tập có thứ tự gồm m trị số thực được thu thập tại những điểm thời gian cách đều nhau. Một *chuỗi con* S với chiều dài k của chuỗi T được ký hiệu là $S = (t_i, t_{i+1}, ..., t_{i+k-1})$, với $1 \le i \le m - k+1$.

Định nghĩa 1 (*Khớp không tầm thường*): Giả sử chuỗi thời gian T có chứa một chuỗi con C_p với chiều dài n bắt đầu tại vị trí p và một chuỗi con trùng khớp với nó là C_q bắt đầu tại vị trí q, thì C_q là chuỗi con trùng khớp không tầm thường của C_p nếu $|p - q| \ge n$, tức C_p không phủ lấp lên C_q.

Định nghĩa 2 (*Chuỗi con bất thường nhất*, *1-discord*): Cho chuỗi thời gian T, chuỗi con D trong T được gọi là chuỗi con bất thường nhất (hay *1-discord*) trong T nếu D có khoảng cách lớn nhất đến chuỗi con trùng khớp không tầm thường lân cận nhất với nó.

Hình 1 minh họa chuỗi thời gian điện tâm đồ (ECG) có chứa một chuỗi con bất thường nhất (được tô đậm).



Hình 1. *Chuỗi thời gian ECG và chuỗi con bất thường nhất ([7])*

B. PHÂN ĐOẠN CHUỖI THỜI GIAN TRỰC TUYẾN

Trong công trình này, chúng tôi sử dụng phương pháp phân đoạn chuỗi thời gian dựa vào các điểm cực trị quan trọng (important extreme point) của Fink và Gandhi [8]. Trong số nhiều phương pháp phân đoạn chuỗi thời gian, nhưng phương pháp điểm cực trị quan trọng được chọn vì phương pháp này có tính trực tuyến nên có thể thích nghi với dữ liệu chuỗi thời gian dạng luồng.

Với phương pháp điểm cực trị quan trọng, có một tham số mà người dùng phải xác định là hệ số nén R. Hệ số nén có giá trị càng lớn sẽ khiến cho số điểm cực trị quan trọng được rút trích sẽ càng ít.

Khởi sự từ điểm đầu tiên của chuỗi thời gian T, chúng ta có thể nhận dạng tất cả các điểm cực tiểu và cực đại quan trọng của chuỗi thời gian bằng cách dùng giải thuật All-Extrema, được đề xuất trong công trình [8]. Giải thuật All-Extrema có độ phức tạp về thời gian và chỗ bộ nhớ đều là tuyến tính. Giải thuật này có thể xử lý các điểm dữ liệu mới khi chúng được truyền tới mà không cần lưu tòan bộ cả chuỗi thời gian đã có.

Trong công trình này, sau khi tìm ra tất cả các điểm cực trị quan trọng của chuỗi thời gian, chúng tôi có thể rút trích ra những phân đoạn từ một chuỗi thời gian tĩnh hay chuỗi thời gian dạng luồng. Tóm lại, giải thuật All-Extrema có thể giúp chúng ta thực hiện phân đoạn chuỗi thời gian một cách *trực tuyến* (online).

III. CÁC CÔNG TRÌNH LIÊN QUAN

A. CÁC CÔNG TRÌNH VỀ PHÁT HIỆN BẤT THƯỜNG TRÊN CHUỖI THỜI GIAN TĨNH

Đã có nhiều công trình nghiên cứu về phát hiện bất thường trên chuỗi thời gian tĩnh. Một số giải thuật tiêu biểu được đề xuất cho bài toán này được liệt kê như sau. Giải thuật Brute-Force, của Keogh và các cộng sự [7], là một giải thuật chân phương để phát hiện bất thường bao gồm hai vòng lặp lồng nhau. Giải thuật HOT SAX, của Keogh và các cộng sự [7], sử dụng phương pháp Xấp Xỉ Gộp Từng Đoạn (Piecewise Aggregate Approximation -PAA) (Keogh và các cộng sự [9]) như là một kỹ thuật thu giảm số chiều, phương pháp Xấp Xỉ Gộp Ký Hiệu Hóa (Symbolic Aggregate Approximation -SAX) (Lin và các cộng sự, [10]) như là một kỹ thuật rời rạc hóa và áp dụng

hai heuristic để sắp thứ tự cho vòng lặp trong và vòng lặp ngoài nhằm cải tiến quá trình phát hiện bất thường. Trong giải thuật WAT, Bu và các cộng sự phát hiện bất thường bằng cách sử dụng phép biến đổi Haar Wavelet và áp dụng cây gia tố như là một cấu trúc dữ liệu hỗ trợ [11]. Ma và Perkins đề xuất phát hiện bất thường dựa vào máy vector hỗ trợ một lớp (one-class support vector machine) [12]. Trường và Anh đề xuất phương pháp phát hiện bất thường và motif trên chuỗi thời gian dựa vào phân đoạn và gom cụm [13]. Sanchez và Butos đề xuất một phương pháp phát hiện bất thường trên chuỗi thời gian với nhiều mức phân giải [14]. Zhu và các cộng sự đề xuất khung thức để phát hiện bất thường và motif trên chuỗi thời gian dựa vào sự song song hóa với công nghệ GPU [15].

Các phương pháp nêu trên được chia làm ba thể loại: các phương pháp dựa vào cửa sổ trượt, các phương pháp dựa vào phân đoạn và các phương pháp dựa vào phân lớp [16]. Trong số những phương pháp dựa vào cửa sổ trượt, HOT SAX được xem là phương pháp được ưa chuộng nhất.

B. CÁC CÔNG TRÌNH VỀ PHÁT HIỆN BẤT THƯỜNG TRÊN CHUỖI THỜI GIAN DẠNG LUỒNG

Đối với bài toán phát hiện bất thường trên chuỗi thời gian dạng luồng, do độ khó của bài toán, có khá ít những công trình nghiên cứu. Một vài nghiên cứu về bài toán này được tóm lược như sau.

Liu và các cộng sự, năm 2009, đề xuất một khung thức phát hiện bất thường trên chuỗi thời gian dạng luồng có tên DCD (Detection of Continuous Discords) [17]. DCD có thể phát hiện chuỗi con bất thường từ các phân đoạn cục bộ của một chuỗi thời gian dạng luồng được lưu trên một bộ đệm (buffer) mà có kích thước cho trước. Khung thức DCD dùng kỹ thuật chính là giới hạn không gian tìm kiếm nhằm gia tăng tính hữu hiệu của quá trình phát hiện bất thường. Vì khung thức DCD thuộc nhóm phương pháp dựa vào cửa sổ trượt nên DCD có độ phức tạp tính toán cao.

Sanchez và Bustos năm 2014, giới thiệu một phương pháp phát hiện bất thường trên chuỗi thời gian dạng luồng mà sử dụng những hình chữ nhật bao và cấu trúc R-tree để thiết lập hai heuristic sắp thứ tự cho vòng lặp trong và vòng lặp ngoài trong quá trình phát hiện bất thường [18]. Tuy nhiên khái niệm chuỗi con bất thường trong công trình này không tương thích với định nghĩa chuẩn về chuỗi con bất thường nhất (1-discord) (xem Định nghĩa 2) mà thường được dùng trong cộng đồng nghiên cứu.

Giao và Anh, năm 2020, đề xuất giải thuật SKDIS để phát hiện k chuỗi con bất thường nhất trên chuỗi thời gian dạng luồng [19]. Tuy nhiên, do giải thuật SKDIS đi theo hướng tiếp cận dựa vào cửa sổ trượt nên độ phức tạp tính toán của giải thuật này vẫn còn cao.

C. GIẢI THUẬT I-LEADER ĐỂ GOM CỤM CÁC CHUỖI CON

Một giải thuật gom cụm gia tăng (incremental clustering), I-Leader, để gom cụm các chuỗi con từ một chuỗi thời gian (Thủy và các cộng sự, 2019 [6]), là một sự cải tiến từ giải thuật gom cụm Leader [20] với những ý tưởng chính sau đây.

- Một là trong I-Leader, chúng tôi sử dụng tâm cụm (centroid) thay vì "leader" trong vai trò đại diện cụm.
- Hai là trong I-Leader, các tâm cụm được tính theo kiểu gia tăng (incremental). Phương pháp cập nhật lại tâm cụm của một cụm khi có một chuỗi con mới được đưa vào cụm được mô tả như sau. Khi một chuỗi con mới được thêm vào một cụm, tâm cụm của cụm này sẽ được tính lại theo kiểu gia tăng bằng cách áp dụng công thức sau đây để tính thành phần thứ i của tâm cụm mới.

$$NewCentroid_i = \frac{Centroid_i * n + t_i}{n+1} \qquad (1)$$

với t_i là thành phần thứ i của chuỗi con t được gán vào cụm, n là số phần tử trong cụm, $Centroid_i$ là thành phần thứ i của tâm cụm hiện hành và $NewCentroid_i$ là thành phần thứ i của tâm cụm mới. Khi một chuỗi con bị loại bỏ ra khỏi một cụm, để tính lại tâm cụm mới cho cụm này, ta cũng có thể áp dụng công thức (1) nhưng trong đó hai dấu cộng được thay bằng hai dấu trừ.

- Ba là trong I-Leader, chất lượng tốt của các cụm được gom sẽ được duy trì tại bước cập nhật cụm bằng cách kiểm tra thông tin *kiểu cụm* (cluster type). Sau chặng gom cụm đầu tiên, cụm nào thuộc loại "under-filled" (có số phần tử quá ít) sẽ được gộp vào một cụm thuộc loại "good" (có số phần tử đủ nhiều) mà gần với nó nhất. Về khoảng cách giữa một chuỗi con đến một cụm, chúng tôi sử dụng độ đo khoảng cách Euclid từ chuỗi con đó đến tâm cụm của cụm ấy. Khi mọi cụm đều thuộc loại "good", chúng ta biết rằng kết quả gom cụm đã đạt được một chất lượng tốt. Bước tinh chế cụm bằng cách gộp này có thể được lặp nhiều lần cho đến khi không còn cụm nào thuộc loại "under-filled" trong tất cả các cụm.

Giải thuật I-Leader để gom cụm gia tăng các chuỗi con được mô tả như sau:

Input: Chuỗi thời gian dạng luồng T được phân đoạn thành các chuỗi con và một ngưỡng khoảng cách ε
Output: C là một tập các cụm được tạo thành.

Bước 1: Gán chuỗi con thứ nhất T_1, vào cụm C_1; gán $i = 1$ và $j = 1$ và nhận T_j là tâm cụm của cụm C_i.

Bước 2: Gán $j = j + 1$; xét các cụm từ C_1 đến C_i theo thứ tự tăng dần của chỉ số cụm và gán chuỗi con T_j vào cụm C_m ($1 \le m \le i$) nếu khoảng cách từ tâm cụm của cụm C_m đến chuỗi con T_j là nhỏ nhất và khoảng cách này phải nhỏ hơn ngưỡng ε. (Trong trường hợp này, tâm cụm của C_m được tính lại bằng cách dùng Công thức 1). Ngược lại, gán $i = i+1$ và đưa chuỗi con T_j vào cụm mới C_i. Nhận T_j là tâm cụm của cụm C_i..

Bước 3: Tinh chỉnh kết quả gom cụm ở Bước 2 bằng cách gộp những chuỗi con thuộc các cụm "under-filled" vào các cụm "good" mà gần với chúng nhất.

Bước 4: Quay lại Bước 2 để lặp cho đến khi mọi chuỗi con trong chuỗi thời gian T đều đã được gán vào các cụm.

D. EP-ILEADER, GIẢI THUẬT PHÁT HIỆN BẤT THƯỜNG TRÊN CHUỖI THỜI GIAN TĨNH

Giải thuật EP-ILeader để phát hiện bất thường trên chuỗi thời gian tĩnh dựa vào các ý tưởng sau. Trước tiên, chuỗi thời gian được phân đoạn thành các chuỗi con dựa vào các điểm cực trị quan trọng đã tìm thấy. Sau đó, một giải thuật gom cụm được dùng để gom cụm các chuỗi con đó thành nhiều cụm. Sau khi các chuỗi con đã được gom cụm, hệ số bất thường (anomaly score) của mỗi chuỗi con sẽ được tính. Chuỗi con có hệ số bất thường cao nhất sẽ được nhận dạng là chuỗi con bất thường nhất (1-discord). Nếu có nhiều chuỗi con cùng có hệ số bất thường cao nhất thì một trong những chuỗi con này sẽ được chọn là chuỗi con bất thường nhất.

Với những ý tưởng nêu trên, chúng tôi đã đề xuất giải thuật EP-ILeader (viết tắt cho Extreme Points and I-Leader clustering), một giải thuật dựa vào phân đoạn để phát hiện bất thường trên chuỗi thời gian tĩnh. Chi tiết của giải thuật này được mô tả như sau.

Trước tiên, phương pháp điểm cực trị quan trọng [8] được dùng để tìm ra tất cả các điểm cực trị quan trọng mà có công dụng tách hai phân đoạn kế cận nhau. Chúng tôi gọi những điểm như vậy là những điểm thay đổi (*change points*).

Trong quá trình phân đoạn, mỗi phân đoạn được hình thành từ điểm thay đổi thứ i đến điểm thay đổi thứ ($i+k$), với k lớn hơn hay bằng 2 và nhỏ hơn tổng số điểm thay đổi tìm thấy.

Sau khi được phân đoạn từ một chuỗi thời gian, các chuỗi con được phân đoạn có thể có chiều dài khác nhau. Do đó, một phương pháp nội suy sẽ được dùng để biến đổi các chuỗi con này về cùng một chiều dài. Trong EP-ILeader, chúng tôi áp dụng phép biến đổi vị tự (homothety) để biến đổi các chuỗi con này về cùng một chiều dài (chính là chiều dài trung bình của mọi chuỗi con). Chi tiết về giải thuật biến đổi vị tự các chuỗi con của chuỗi thời gian được mô tả trong bài báo [13].

Khi tất cả các chuỗi con có cùng chiều dài, giải thuật gom cụm I-Leader sẽ thực thi để gom cụm các chuỗi con tương tự nhau vào cùng một cụm và những chuỗi con khác nhau vào những cụm khác nhau. Các cụm sẽ được phân loại về kiểu cụm là "good" hoặc "under-filled" để tiện theo dõi trong quá trình gom cụm. Căn cứ vào thông tin kiểu cụm, giải thuật I-Leader có thể cải thiện chất lượng gom cụm bằng cách thực hiện các bước tinh chế theo kiểu gộp cho đến khi không còn cụm nào thuộc kiểu "under-filled".

Với kết quả gom các chuỗi con thành các cụm, chúng tôi tính toán hệ số bất thường của từng chuỗi con bằng cách dùng hai định nghĩa sau đây được đề nghị bởi He và các cộng sự trong công trình phát hiện điểm ngoại biên [21]. Trong cách tính toán này, các chuỗi con bất thường được thể hiện thông qua những khoảng cách từ các chuỗi con này đến các cụm lớn và các cụm nhỏ.

Định Nghĩa 3. (*Cụm lớn và cụm nhỏ*): Cho một tập chuỗi con D mà được gom cụm thành một tập các cụm $C = \{C_1, C_2, ..., C_k\}$ sao cho $|C_1| \ge |C_2| \ge .... \ge |C_k|$. Với hai tham số α_1, α_2, nếu ($|C_1| + |C_2| + ... + |C_b|$) lớn hơn hay bằng $|D| * \alpha_1$ và $|C_b| / |C_{b+1}|$ lớn hơn hay bằng α_2, thì các cụm $C_1, C_2, ..., C_b$ là các cụm lớn và các cụm $C_{b+1}, C_{b+2}, ..., C_k$ là các cụm nhỏ, với b là số nguyên trong tầm từ 1 đến k.

Các cụm lớn được ký hiệu là *LC* và các cụm nhỏ được ký hiệu là *SC*.

Định Nghĩa 4. (*Hệ số bất thường*): Cho một tập chuỗi con được phân đoạn từ chuỗi thời gian T và tập chuỗi con này được gom cụm thành một tập các cụm $C = \{C_1, C_2, ..., C_k\}$ sao cho $|C_1| \ge |C_2| \ge .... \ge |C_k|$. Với mỗi chuỗi con t được phân đoạn từ chuỗi thời gian T, hệ số bất thường của t được tính bằng công thức sau sau:

$$Score(t) = \begin{cases} |C_i| * \min(distance(t, C_j)) & if\ t \in C_i, C_i \in SC \text{ and } C_j \in LC (j = 1..b) \\ |C_i| * distance(t, C_i) & if\ t \in C_i, C_i \in LC \end{cases} \quad (2)$$

Với $distance(t, C_i)$ là khoảng cách từ chuỗi con t đến cụm C_i.

Kết quả thực nghiệm ở công trình [5] cho thấy với năm bộ dữ liệu thử nghiệm, khi phát hiện bất thường EP-ILeader thực thi hữu hiệu hơn rất nhiều so với giải thuật HOT SAX mà vẫn phát hiện bất thường một cách chính xác như giải thuật HOT SAX. Chi tiết về giải thuật EP-ILeader, độc giả quan tâm có thể tham khảo đến bài báo [5].

IV. GIẢI THUẬT ĐỀ XUẤT ĐỂ PHÁT HIỆN BẤT THƯỜNG TRÊN CHUỖI THỜI GIAN DẠNG LUỒNG

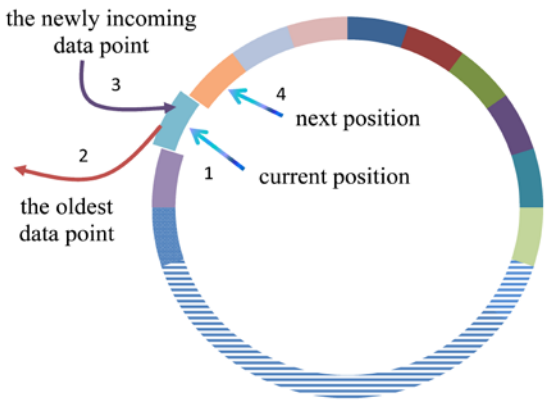
Để phát hiện bất thường trên chuỗi thời gian dạng luồng, chúng tôi đề xuất giải thuật SEP-ILeader, một dạng mở rộng của giải thuật EP-ILeader. Trong giải thuật SEP-ILeader, một *cửa sổ dịch chuyển* (moving window) dưới hình thức một *bộ đệm xoay vòng* (a circular buffer) được dùng làm *cấu trúc trung gian* để lưu chuỗi thời gian dạng luồng bằng cách lưu chỉ một phân đoạn cục bộ của chuỗi thời gian bao gồm những điểm dữ liệu mới nhất. Chúng tôi định nghĩa thêm một chiến lược *cập nhật trì hoãn* (delayed update policy) để quyết định khi nào cần tái phân đoạn chuỗi thời gian và khi nào quá trình nhận dạng bất thường được tái khởi động để phát hiện một chuỗi con bất thường mới một cách trực tuyến. Như vậy, giải thuật SEP-ILeader giống giải thuật EP-ILeader ở chỗ vẫn sử dụng cách tiếp cận dựa vào phân đoạn và gom cụm; và giải thuật SEP-ILeader khác giải thuật EP-ILeader ở chỗ phải bổ sung thêm việc sử dụng bộ đệm xoay vòng làm cấu trúc trung gian và chiến lược cập nhật trì hoãn để ứng phó với những đặc điểm của dữ liệu luồng.

A. BỘ DỆM XOAY VÒNG

Hầu hết các giải thuật xử lý chuỗi thời gian dạng luồng thường quan tâm đến phần mới nhất của chuỗi bằng cách áp dụng một cửa sổ dịch chuyển với kích thước W lên chuỗi thời gian dạng luồng. Bằng cách này, chỉ có W trị mới nhất của chuỗi thời gian dạng luồng được xem xét khi khai phá dữ liệu, trong khi những giá trị cũ hơn bị xem như là lỗi thời và chúng sẽ không được xem xét. Trong nghiên cứu này, cửa sổ dịch chuyển được hiện thực như là một bộ đệm xoay vòng. Bộ đệm này là một danh sách vận hành theo kiểu xoay vòng (circular list).

Cách vận hành của bộ đệm xoay vòng như là một cấu trúc trung gian được minh họa ở Hình 2. Khi xử lý một chuỗi thời gian dạng luồng, điểm dữ liệu mới tới sẽ ghi đè lên điểm dữ liệu cũ nhất trong bộ đệm xoay vòng.

Kích thước của bộ đệm có thể ảnh hưởng đến tính hữu hiệu của quá trình phát hiện bất thường trên chuỗi thời gian dạng luồng. Trong nghiên cứu này, chiều dài của bộ đệm được ước lượng dựa vào *chu kỳ* (period) của chuỗi thời gian đang xử lý. Chiều dài của bộ đệm nên là bội số của chu kỳ của chuỗi thời gian để cho với chuỗi thời gian có tính chu kỳ, điểm dữ liệu mới tới và điểm dữ liệu cũ nhất bị loại bỏ sẽ có sự biến thiên tương đối giống nhau. Có một vài phương pháp để phát hiện chu kỳ trên dữ liệu chuỗi thời gian.



Hình 2. *Bộ đệm xoay vòng*

Trong giải thuật SEP-ILeader, chúng tôi áp dụng phương pháp ước lượng chu kỳ của một chuỗi thời gian được đề xuất bởi Phiên [22]. Phương pháp này dựa vào cách tiếp cận khám phá motif có chiều dài thay đổi mà vận dụng một giải thuật *suy diễn văn phạm*(grammar induction).

B. CHIẾN LƯỢC CẬP NHẬT TRÌ HOÃN

Khi các điểm dữ liệu mới đến liên tục, sẽ rất tốn thời gian để kích hoạt mọi quá trình liên quan đến phát hiện bất thường (thí dụ, giải thuật All-Extrema, giải thuật I-Leader và giải thuật EP-ILeader) để cập nhật đối với từng điểm dữ liệu mới. Để đạt được tính hữu hiệu cao trong bối cảnh chuỗi thời gian dạng luồng, chúng tôi định nghĩa một chiến lược cập nhật trì hoãn để quyết định khi nào cần tái phân đoạn chuỗi thời gian và khi nào quá trình nhận dạng bất thường được tái khởi động để phát hiện một chuỗi con bất thường nhất mới.

Với chiến lược cập nhật trì hoãn, thay vì dựa vào sự xuất hiện của mỗi điểm dữ liệu mới, chúng tôi dựa vào sự xuất hiện của một điểm cực trị quan trọng mới để kích hoạt quá trình phát hiện bất thường. Tức là mỗi khi có một

điểm cực trị quan trọng được nhận dạng trên các điểm dữ liệu mới, quá trình phát hiện bất thường sẽ được kích hoạt để phát hiện ra chuỗi con bất thường nhất mới có trong bộ đệm hiện hành. Vào lúc đó, chúng tôi có được một chuỗi con chứa đựng những điểm dữ liệu mới. Chuỗi con này sẽ được gán đến cụm mà tâm cụm gần với nó nhất. Ngoài ra nếu điểm dữ liệu cũ nhất trong bộ đệm chưa phải là điểm tận cùng bên phải của chuỗi con cũ nhất thì chuỗi con cũ nhất vẫn được tồn tại trong cụm nào đó của nó. Ngược lại, nếu điểm dữ liệu cũ nhất trong bộ đệm lại là điểm tận cùng bên phải của chuỗi con cũ nhất thì chuỗi con đó được xóa bỏ ra khỏi cụm của nó.

C. GIẢI THUẬT SEP-ILEADER

Đầu tiên, khi bộ đệm đã chứa đầy các điểm dữ liệu đến từ một chuỗi thời gian dạng luồng, giải thuật SEP-ILeader khởi động giải thuật EP-ILeader để phát hiện chuỗi con bất thường nhất đầu tiên có trong bộ đệm. Sau bước khởi đầu có tính lịch sử này, SEP-ILeader bắt đầu áp dụng chiến lược cập nhật trì hoãn để phát hiện thêm các chuỗi con bất thường nhất mới từ phần còn lại của chuỗi thời gian dạng luồng. Từ đó trở đi, SEP-ILeader phát hiện thêm một chuỗi con bất thường nhất mới mỗi khi điểm dữ liệu mới nhất đi vào bộ đệm được nhận dạng là một điểm cực trị quan trọng.

Những điểm chính của giải thuật SEP-ILeader được mô tả sơ lược tại Giải Thuật 1.

Giải thuật 1 SEP-ILeader

Input: - *y*: một chuỗi thời gian có chiều dài *n*
- *R*: hệ số nén
- *Threshold*: ngưỡng khoảng cách dùng trong giải thuật gom cụm I-Leader

Output: - chuỗi con bất thường nhất và vị trí của nó

1: Nhập các điểm của chuỗi thời gian vào bộ đệm *B* /* Khởi tạo bộ đệm */
2: Call EP-ILeader(*B*, *R*, *Threshold*, *Sub_List*, *Cluster*, *PositionofAnomaly*)
/* Phát hiện chuỗi con bất thường nhất trong bộ đệm hiện hành; *Sub_List* là danh sách các chuỗi con được rút trích; *Cluster* là tập hợp các cụm được hình thành sau bước gom cụm của giải thuật EP-ILeader */
3: Xuất ra chuỗi con bất thường nhất trong bộ đệm hiện hành và vị trí của nó
4: **repeat**
5: **if** Type(Last_Extreme_Point) = Max **then**
6: Tìm điểm cực tiểu quan trọng kế tiếp trong các điểm dữ liệu vừa được truyền tới
7: **else** /* điểm cực trị mới nhất là điểm cực tiểu
8: Tìm điểm cực đại quan trọng kế tiếp trong các điểm dữ liệu vừa được truyền tới
9: **endif**
10: Với điểm cực trị quan trọng mới nhận dạng, hình thành chuỗi con mới nhất vừa đến Sub
11: Homothety(Sub, NewSub) /* NewSub là chuỗi con vừa được biến đổi bằng phép vị tự từ chuỗi con Sub*/
12: Insert(Newsub, *NewSub_List*, *Cluster*) /* Đưa chuỗi con Newsub vào cấu trúc cụm hiện hành *Cluster*; *NewSub_List* là danh sách *Sub_List* sau khi có thêm chuỗi con Newsub. */
13: **if** mọi điểm dữ liệu thuộc chuỗi con cũ nhất (FirtSub) trong *NewSub_List* đều đã ra khỏi bộ đệm *B* **then**
14: Delete(FirstSub, *NewSub_List*, *Cluster*)
/* xóa chuỗi con cũ nhất *FirstSub* ra khỏi cấu trúc cụm hiện hành *Cluster* và ra khỏi danh sách *NewSub_List* */
15: **endif**
16: Tính hệ số bất thường cho mọi chuỗi con trong cấu trúc cụm hiện hành *Cluster*
17: Xuất ra chuỗi con có hệ số bất thường lớn nhất trong bộ đệm hiện hành và vị trí của nó
18: **until** Stop

Chú ý rằng đầu tiên tại lúc bộ đệm bị đầy dữ liệu, SEP-ILeader sẽ gọi EP-ILeader để phát hiện chuỗi con bất thường nhất đầu tiên (các dòng lệnh 1 – 3). Từ đó, SEP-ILeader kích hoạt quá trình phát hiện bất thường liên tiếp bất cứ khi nào có một điểm cực trị quan trọng mới được nhận dạng (các dòng lệnh 5-9). Điều kiện dừng của vòng lặp (**repeat until**) hàm ý rằng toàn bộ chuỗi thời gian đã được xem xét. Trong thực tế, quá trình lặp cứ tiếp tục nếu còn có điểm dữ liệu mới truyền tới. Một chuỗi con mới hình thành (*Newsub*) được đưa vào cụm thích hợp bởi trình con *Insert* (dòng lệnh 12) và chuỗi con cũ nhất (*Firstsub*) trong bộ đệm bị xóa khỏi cụm nào đó bởi trình con *Delete* (dòng lệnh 14). Chú ý rằng do đặc điểm của giải thuật All-Extrema [8], bước nhận dạng các điểm cực trị quan trọng trong giải thuật SEP-ILeader có thể làm việc theo cách gia tăng trên chuỗi thời gian dạng luồng. Ngoài ra, trình con *Insert* có thể đưa một chuỗi con mới *Newsub* vào một cụm thích hợp bằng cách tìm một

cụm sao cho khoảng cách giữa *Newsub* và tâm cụm của cụm đó là nhỏ nhất và sau đó tâm cụm của cụm đó sẽ được cập nhật (cách cập nhật tâm cụm khi có thêm hay bớt một chuỗi con được mô tả chi tiết trong ở tiểu mục III.C). Cuối cùng, trình con *Delete* có thể phải xóa bỏ chuỗi con cũ nhất (*Firstsub*) ra khỏi cụm hiện hành của nó và do đó tâm cụm của cụm này sẽ được cập nhật.

V. THỰC NGHIỆM

Trong phần đánh giá bằng thực nghiệm, chúng tôi hiện thực hai giải thuật để so sánh hiệu quả phát hiện bất thường trên chuỗi thời gian dạng luồng: SEP-ILeader và giải thuật BFHS (viết tắt cho Brute-Force HOT SAX). BFHS là một dạng thức mở rộng của giải thuật HOT SAT để phát hiện bất thường trên chuỗi thời gian dạng luồng. Trong giải thuật BFHS, HOT SAX được khởi động mỗi khi có điểm dữ liệu mới được truyền vào bộ đệm. Điều này hàm ý rằng BFHS phải tìm kiếm trên toàn bộ dữ liệu trong bộ đệm để thực hiện một quá trình phát hiện chuỗi con bất thường nhất mỗi khi có điểm dữ liệu mới được truyền tới. Chúng tôi hiện thực hai giải thuật bằng ngôn ngữ lập trình Visual C#. Các thực nghiệm được tiến hành trên máy tính có cấu hình HP Intel® Core ™ i7-3630 QM CPU 2.40GHz, 8GB RAM.

Trong thực nghiện, chúng tôi thử nghiệm với 6 bộ dữ liệu mẫu. Chúng có tên là POWER, AF_learning-set-n01, AEM, power_demand_Italy, AHA_0001_ECG và ECG. Bốn bộ dữ liệu POWER, AEM, power_demand_Italy và ECG được lấy từ kho dữ liệu chuỗi thời gian *The UCR Time series Classification/Clustering* [23]. Hai bộ dữ liệu AF_learning-set-n01 và AHA_0001_ECG lấy từ trang Web: https://www.physionet.org. Tên, chiều dài và các thông số liên quan được trình bày ở Bảng 1. Các bộ dữ liệu này thuộc về hai lãnh vực khác nhau. Ba bộ dữ liệu AF_learning-set-n01, AHA_0001_ECG và ECG thuộc lãnh vực y khoa. Ba bộ dữ liệu còn lại thuộc lãnh vực công nghiệp. Các bộ dữ liệu này được lưu dưới dạng tập tin văn bản. Là dữ liệu nhập, các tập tin này sẽ được giả lập thành những chuỗi thời gian dạng luồng để thực nghiệm.

Đối với EP-ILeader có hai tham số phải xác định: hệ số nén R và ngưỡng khoảng cách ε dùng trong giải thuật I-Leader. Đối với SEP-ILeader, có thêm hai tham số phải xác định: chu kỳ của chuỗi thời gian và chiều dài bộ đệm xoay vòng,. Đối với BFHS, do sử dụng giải thuật HOT SAX như là trình con căn bản, cần có thêm ba tham số phải xác định: chiều dài của chuỗi con được rút trích, kích thước của mỗi đoạn PAA, và chiều dài từ SAX *w.* Giá trị của các tham số trong hai giải thuật ứng với mỗi bộ dữ liệu được nêu ở Bảng 1. Thí dụ như với bộ dữ liệu POWER, chu kỳ của chuỗi thời gian được tìm thấy là 226, chúng tôi chọn chiều dài bộ đệm là 452 (452 = 2*226); chiều dài chuỗi con bất thường được tìm thấy bởi SEP-ILeader là 186 và chiều dài mỗi đoạn PAA được xác định là 3 và chiều dài từ SAX được xác định là 62 cho giải thuật BFHS.

Chú ý rằng EP-ILeader không đòi hỏi phải xác định chiều dài chuỗi con bất thường như là một tham số. Do đó, sau khi thực thi SEP-ILeader trên mỗi bộ dữ liệu, chúng tôi dùng chiều dài chuỗi con được tìm thấy bởi SEP-ILeader như là tham số chiều dài chuỗi con bất thường tương ứng cho giải thuật BFHS.

Bảng 1. *Tên, chiều dài, chu kỳ, chiều dài bộ đệm, chiều dài chuỗi con và hai thông số khác của mỗi bộ dữ liệu*

STT	Bộ dữ liệu	Chiều dài	Chu kỳ	Chiều dài bộ đệm	Chiều dài chuỗi con	Đoạn PAA	Chiều dài từ *w*
1	POWER	1000	226	452	186	3	62
2	AF_learning-set-n01	1280	433	866	56	2	28
3	AEM	1000	80	480	23	2	12
4	power_demand_Italy	1000	80	480	179	2	90
5	AHA_0001_ECG	2500	106	1272	200	2	100
6	ECG	5000	33	2310	101	3	34

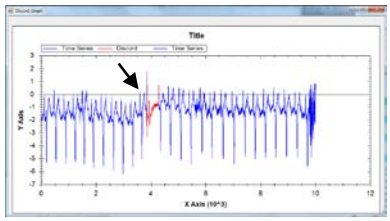
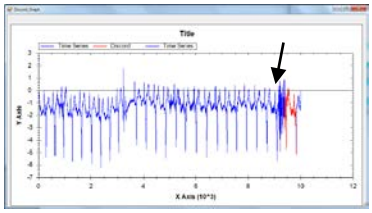
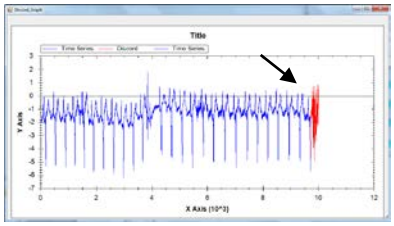
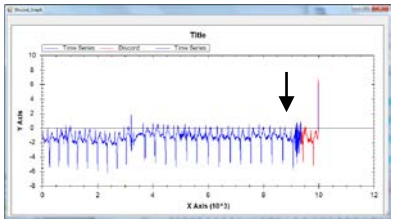
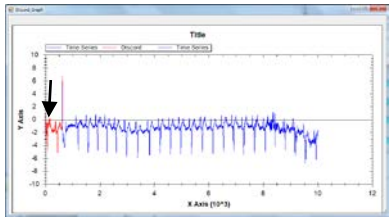
A. ĐỘ CHÍNH XÁC

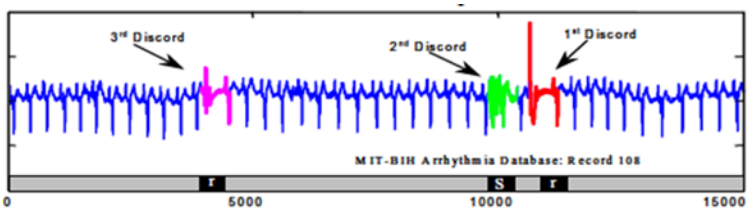
Bảng 2 trình bày hình dạng của các chuỗi con bất thường nhất (được chỉ bằng mũi tên) tìm thấy trên bộ dữ liệu ECG (điện tâm đồ) sau khi SEP-ILeader nhận dạng được vài điểm cực trị quan trọng.

Trong Bảng 2, trạng thái thứ nhất là hình ảnh của một phân đoạn cục bộ của chuỗi thời gian dạng luồng sau khi được đưa vào bộ đệm. Vị trí của chuỗi con bất thường nhất được tìm thấy trong bộ đệm tại lúc này có vị trí xấp xỉ điểm 4000. Các trạng thái thứ hai, ba, bốn và năm trong Bảng 2 minh họa hình ảnh của bộ đệm sau khi nhận dạng thêm vài điểm cực trị quan trọng mới và từ đó phát hiện thêm những chuỗi con bất thường nhất tương ứng tại bộ đệm hiện hành. Các kết quả chuỗi con bất thường nhất được phát hiện trên bộ dữ liệu ECG gần như khớp với ba chuỗi con bất thường bậc 1, bậc 2 và bậc 3 đã được đánh dấu bởi các chuyên gia và được tường thuật

trong bài báo (Keogh và các cộng sự [7]) (trình bày ở Hình 3). Ngoài ra, không hề có *lỗi tìm sai* (false alarm) nào xảy ra với giải thuật SEP-ILeader.

Bảng 2. *Các chuỗi con bất thường nhất được phát hiện bởi SEP-ILeader trên bộ dữ liệu ECG sau khi nhận dạng được vài điểm cực trị quan trọng.*

Trạng thái	Chuỗi con bất thường nhất	Trạng thái	Chuỗi con bất thường nhất
1		3	
2		4	
		5	



Hình 3. *Các chuỗi con bất thường bậc 1, bậc 2 và bậc 3 được tìm thấy trên bộ dữ liệu ECG ([6])*

B. TÍNH HỮU HIỆU

Bảng 3 trình bày thời gian thực thi (tính bằng giây) của hai giải thuật phát hiện bất thường trên 6 bộ dữ liệu chuỗi thời gian dạng luồng. Từ kết quả thống kê ở Bảng 3, chúng ta có thể thấy SEP-ILeader thực thi nhanh hơn BFHS rất nhiều trên tất cả các bộ dữ liệu. Trung bình, SEP-ILeader thực thi nhanh hơn BFHS khoảng 659.8 lần.

Sự kiện SEP-ILeader hữu hiệu hơn nhiều so với BFHS có thể được giải thích như sau. Một là giải thuật SEP-ILeader dựa vào giải thuật EP-ILeader và giải thuật này hữu hiện hơn nhiều so với giải thuật HOT SAX mà giải thuật BFHS dựa vào. Hai là bằng cách sử dụng chiến lược cập nhật trì hoãn, SEP-ILeader kích hoạt quá trình phát hiện bất thường mỗi khi nó nhận diện được một điểm cực trị quan trọng mới trong khi BFHS kích hoạt quá trình phát hiện bất thường mỗi khi có một điểm dữ liệu mới truyền tới. Ba là, nhờ vào tính chất trực tuyến của giải thuật All-Extreme trong việc nhận diện các điểm điểm cực trị quan trọng và tính gia tăng của giải thuật gom cụm I-Leader, SEP-ILeader có thể phát hiện bất thường một cách rất hữu hiện trong bối cảnh dữ liệu luồng.

Để kiểm tra khả năng đáp ứng tức thời của giải thuật đề xuất khi làm việc với chuỗi thời gian dạng luồng, chúng tôi đo đạc thời gian thực thi từ lúc một điểm cực trị quan trọng mới xuất hiện cho đến lúc hoàn tất việc phát hiện chuỗi con bất thường nhất tương ứng. Thời gian đáp ứng cho từng bộ dữ liệu mẫu được tổng kết ở Bảng 4. Kết qủa trong Bảng 4 cho thấy trung bình, giải thuật SEP-ILeader có thể đáp ứng với khoảng 21.8 mili- giây mỗi khi có một điểm cực trị quan trọng mới xuất hiện. Bảng 4 cũng cho thấy trung bình, giải thuật BFHS có thể đáp ứng với thời gian khoảng 168 mili giây mỗi khi có một điểm dữ liệu mới truyền tới.

Bảng 3. *Thời gian thực thi của BFHS và SEP-ILeader trên 6 bộ dữ liệu*

STT	Bộ dữ liệu	Chiều dài bộ đệm	Thời gian thực thi (giây)	
			BFHS	SEP-ILeader
1	POWER	452	12.144	0.059
2	AF_learning-set-n01	866	50.529	0.59
3	AEM	480	13.352	0.59
4	power_demand_Italy	480	16.529	1.382
5	AHA_0001_ECG	1272	354.845	0.1
6	ECG	2310	1962.522	23.285

Bây giờ chúng tôi muốn trả lời câu hỏi, liệu giải thuật phát hiện bất thường SEP-ILeader có thể đáp ứng được yêu cầu về thời gian khi làm việc với các ứng dụng thực tế. Hãy xem xét trường hợp làm việc với bộ dữ liệu ECG (điện tâm đồ). Với bộ dữ liệu ECG, chiều dài của một nhịp tim tương đương với khoảng một giây. Mỗi điểm cực trị quan trọng cách nhau khoảng nửa nhịp tim. Giải thuật SEP-ILeader có thể phát hiện chuỗi con bất thường trong tầm 30 mili-giây (xem Bảng 4). Như vậy, tốc độ phát hiện bất thường của giải thuật SEP-Ileader nhanh gấp 16.7 lần so với tốc độ truyền của dữ liệu ECG. Sự phân tích trên cho thấy giải thuật đề xuất để phát hiện bất thường trên chuỗi thời gian dạng luồng có thể đáp ứng được yêu cầu về thời gian khi làm việc với dữ liệu ECG. Hãy xem xét trường hợp chúng ta dùng giải thuật BFHS để phát hiện bất thường trên dữ liệu ECG dạng luồng. Mỗi nhịp tim bao gồm khoảng 100 điểm dữ liệu được truyền tới với tốc độ 1 giây. Như vậy mỗi điểm dữ liệu ECG sẽ được truyền tới với tốc độ 10 mili-giây trong khi thời gian đáp ứng của BFHS là khoảng 178 mili-giây (xem Bảng 4) đối với mỗi điểm dữ liệu mới tới. Như vậy, thời gian đáp ứng của giải thuật BFHS là quá chậm, không thể phù hợp với tốc độ truyền của dữ liệu ECG.

Bảng 4. *Thời gian đáp ứng (tính bằng giây) của SEP-ILeader và BFHS trên 6 bộ dữ liệu chuỗi thời gian dạng luồng*

STT	Bộ dữ liệu	Chiều dài	Chiều dài bộ đệm	SEP-ILeader	BFHS
1	POWER	1000	452	0.001	0.44
2	AF_learning-set-n01	1280	866	0.030	0.239
3	AEM	1000	480	0.020	0.152
4	power_demand_Italy	1000	480	0.030	0.290
5	AHA_0001_ECG	2500	1272	0.020	0.105
6	ECG	5000	2310	0.030	0.178

VI. KẾT LUẬN

Trong bài báo này, chúng tôi đã đề xuất một giải thuật hiệu quả để phát hiện bất thường trên chuỗi thời gian dạng luồng. Giải pháp này dựa vào kỹ thuật phân đoạn có tính trực tuyến và kỹ thuật gom cụm gia tăng.

Dựa vào một giải thuật phát triển trước đây, EP-ILeader, để phát hiện bất thường trên chuỗi thời gian tĩnh, chúng tôi đề xuất một giải thuật mới, SEP-ILeader, có sử dụng thêm một bộ đệm xoay vòng và một chiến lược cập nhật trì hoãn để phát hiện chuỗi con bất thường nhất trên chuỗi thời gian dạng luồng. Giải thuật đề xuất, SEP-ILeader, khi thực nghiệm trên nhiều bộ dữ liệu mẫu, đã thực thi hữu hiệu hơn rất nhiều so với giải thuật BFHS, một dạng thức mở rộng của giải thuật HOT SAX để làm việc với chuỗi thời gian dạng luồng. Bên cạnh đó, SEP-ILeader phát hiện ra những chuỗi con bất thường khớp với những chuỗi con bất thường mà BFHS phát hiện.

Trong tương lai, chúng tôi dự định phát triển một vài ứng dụng thực tế cho giải thuật SEP-ILeader. Bên cạnh đó, chúng tôi dự định xây dựng một giải thuật mới để phát hiện *k* chuỗi con bất thường nhất trên chuỗi thời gian dạng luồng.

VII. TÀI LIỆU THAM KHẢO

[1] Y. Zhu and D. Shasha, (2003), Efficient elastic burst detection in data streams, Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 336-345.

[2] D. H. Ngo and B. Veeravalli (2015), Design of a real-time morphology-based anomaly detection methods from ECG streams, *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine* (BIBM), pp. 829-836.

[3] W. Kuang, S. An and H. Jiang (2015), Detecting traffic anomalies in urban areas using taxi GPS data, *Mathematical Problems in Engineering*, vol. 2015.

[4] S. Ahmad, S. Purdy (2017), Unsupervised real-time anomaly detection for streaming data, *Neurocomputing*, vol. 262, pp134-147.

[5] H. T. T. Thuy, D. T. Anh, V. T. N. Chau (2018), A Novel Method for Time Series Anomaly Detection based on Segmentation and Clustering, *Proc. of 10th International Conference on Knowledge and System Engineering* (KSE), IEEE, Ho Chi Minh City, 1-3 November, 2018, pp. 276-281.

[6] H. T. T. Thuy, D. T. Anh, V. T. N. Chau (2019), Incremental Clustering for Time Series based on an Improved Leader Algorithm, *Proc. of 2019 IEEE-RIVF International Conference on Computing and Communication Technologies*, IEEE, Danang, Vietnam, March 20 - 22, 2019, pp. 13-18.

[7] E. Keogh, J. Lin and A. Fu (2005), HOT SAX: Efficiently finding the most unusual time series subsequence, *Proceedings of the fifth IEEE International Conference on Data mining* , Houston, Texas, pp. 226-233.

[8] E. Fink and H.S. Gandhi (2007), Important extrema of time series, *Proceedings of IEEE International Conference on System, Man and Cybernetics.* Montreal, Canada, pp. 366-372.

[9] E. Keogh, K. Chakrabarti, M. Pazzani and S. Mehrotra (2000), Dimensionality reduction for fast similarity search in large time series databases, *Knowledge and Information System,* Vol. 3, No. 3, pp 263-286.

[10] J. Lin, E. Keogh, S. Lonardi and B. Chiu (2003), A symbolic representation of time series, with implications for streaming algorithms, *Proceedings of the 8th ACM SIGMOD Workshop on Research issues in Data mining and Knowledge Discovery*, San Diego, CA, USA, pp. 2-11.

[11] Y. Bu, T.W. Leung, A.W.C Fu, E. Keogh, J. Pei and S. Meshkin (2007), WAT: Finding top-k discords in time series database, *Proceedings of the 2007 SIAM International Conference on Data Mining*, April, pp. 449-454.

[12] J. Ma and S. Perkins (2003), Time series novelty detection using one-class support vector machines, *Proceedings of International Joint Conference on Neural Networks*, Vol. 3, pp. 1741-1745.

[13] C.D. Truong and D. T. Anh (2015), An efficient method for motif and anomaly detection in time series based on clustering, *International Journal of Business Intelligence and Data Mining*, Vol. 10, No. 4, pp. 356-377.

[14] H. Sanchez and B. Butos (2017), Multiresolution time series discord discovery, *Proc. of IWANN 2017*, Part II, LSNCS 10306, pp. 116-128.

[15] Zhu, B., Jiang, Y., Deng, Y. (2021), A GPU acceleration framework for motif and discord based pattern mining, *IEEE Trans. on Parallel and Distributed Systems*, vol.32, no. 8, August, pp. 198-2004.

[16] D. Cheboli (2010), *Anomaly Detection of Time Series*. Master Thesis, University of Minnesota.

[17] Y. Liu, X. Chen, F. Wang and J. Yin (2009), Efficient Detection of Discords for Time Series Stream, *in Advances in Data and Web Management*, Springer Berlin Heidelberg, pp. 629-634.

[18] H. Sanchez and B. Bustos (2014), Anomaly detection in streaming time series based on bounding boxes, *Proc. of Int. Conf. on Similarity Search and Applications*. (SISAP), LNCS 8821, Springer, pp. 201-213.

[19] B.C. Giao and D.T.Anh (2020), Efficient search for top-k discords in streaming time series, *International Journal of Business Intelligence and Data Mining*, vol. 16, no. 4, pp. 397-417.

[20] J. A. Hartigan (1975), *Clustering Algorithms*, John Wiley & Sons, New York.

[21] Z. He, X. Xu and S. Deng (2003), Discovering Cluster-based Local Outliers, *Pattern Recognition Letters*, Vol. 24, No. 9-10, June 2003, pp. 1641-1650.

[22] N. N. Phien (2018), An Efficient Method for Estimating Time Series Motif Length using Sequitur Algorithm, in: (Meng L, Zhang Y. eds.) *Proceeding of Int. Conference on Machine Learning and Intelligent Communication* (MLICOM 2018), LNICST 251, Springer, Cham.

[23] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen and G. Batista, The UCR Time series Classification/Clustering, homepage: www.cs.ucr.edu/~eamonn/time_series_data, (Accessed in 2017).

ANOMALY DETECTION IN STREAMING TIME SERIES: A SEGMENTATION-BASED ALGORITHM

Huynh Thi Thu Thuy, Duong Tuan Anh

ABSTRACT— Anomaly detection in streaming time series is an important problem that has not been fully supported. In this paper, we propose a novel hybrid approach combining segmentation and clustering for anomaly detection in streaming time series. For segmentation, the major extrema method is used to extract subsequences. For clustering, an incremental algorithm is defined on these extracted subsequences. The local segment of a streaming time series is stored in a circular buffer where the anomaly scores of all extracted subsequences are calculated efficiently. Besides, the proposed method uses a delayed update policy to improve the efficiency of anomaly detection in streaming time series. Experimental results on six

benchmark datasets show that our proposed method is more efficient than BFHS, an adaptive version of HOT SAX, while the same anomaly patterns are returned.

Keywords — anomaly detection, streaming time series, clustering, segmentation.

Huỳnh Thị Thu Thủy tốt nghiệp cử nhân tin học tại Trường Đại học Cần Thơ năm 1997 và tốt nghiệp thạc sĩ ngành Khoa học máy tính tại Trường Đại học Bách Khoa Tp. Hồ Chí Minh năm 2009. Hiện nay chị là nghiên cứu sinh về Khoa học máy tính tại Trường Đại học Bách Khoa Tp. Hồ Chí Minh. Lĩnh vực nghiên cứu chính của chị là khai phá dữ liệu chuỗi thời gian.

Dương Tuấn Anh tốt nghiệp tiến sĩ ngành Khoa học máy tính tại Học Viện Công nghệ Á Châu (Asian Institute of Technology), Bangkok, Thái Lan, năm 1998 và đó cũng là nơi mà ông tốt nghiệp thạc sĩ với cùng chuyên ngành. Ông đã là Phó Giáo Sư tại khoa Khoa học và Kỹ thuật máy tính, trường Đại học Bách Khoa, ĐHQG Tp. Hồ Chí Minh từ năm 2007. Hiện nay, ông là giảng viên khoa Công nghệ thông tin, Trường Đại học Ngoại ngữ-Tin học Tp. Hồ Chí Minh. Lĩnh vực nghiên cứu chính của ông là metaheuristics, học máy và khai phá dữ liệu chuỗi thời gian. Ông là đồng tác giả của trên 100 bài báo khoa học.