RESEARCH ARTICLE

TĂNG TỐC DỰA VÀO GPU GIẢI THUẬT PHÂN LỚP CHUỖI THỜI GIAN GỒM TỔ HỢP BỘ PHÂN LỚP 1-NN KẾT HỢP VỚI NHỮNG ĐỘ ĐO KHOẢNG CÁCH KHÔNG ĐÀN HỒI VÀ ĐÀN HỒI

Dương Tuấn Anh[1], Võ Đại Dương[2], Phạm Minh Trí[3]

[1] Khoa Công nghệ thông tin, Trường Đại Học Ngoại Ngữ -Tin Học, Thành Phố Hồ Chí Minh
[2] Công Ty Merkle, Thành Phố Hồ Chí Minh
[3]Viện Khoa Học và Công Nghệ Tính Toán, Thành Phố Hồ Chí Minh
anhdt@huflit.edu.vn, daiduongvo18@gmail.com, tri.pham.hpc@gmail.com

TÓM TẮT— Cải thiện độ hiệu quả và tính hữu hiệu của một phương pháp phân lớp chuỗi thời gian là một vấn đề rất quan trọng. Bài báo này trình bày một cách tiếp cận tổ hợp (ensemble) để phân lớp chuỗi thời gian sử dụng các bộ phân lớp 1-NN (one-nearest neighbor) kết hợp với những độ đo khoảng cách không đàn hồi (non-elastic) và đàn hồi (elastic). Ngoài ra, chúng tôi song song hóa thuật toán và hiện thực dựa vào GPU cho giải pháp đề xuất nhằm cải tiến tính hữu hiệu về thời gian của giải pháp này. Kết quả thực nghiệm của giải pháp đề xuất trên một số bộ dữ liệu chuỗi thời gian chuẩn cho thấy phương pháp đề xuất hiệu quả hơn phương pháp 1-NN kết hợp độ đo Xoắn thời gian động (Dynamic Time Warping- DTW), là phương pháp được giới học thuật liên quan xem là giải thuật hàng đầu (gold baseline) của bài toán phân lớp chuỗi thời gian và đồng thời phương pháp đề xuất đạt hiệu quả ngang bằng với phương pháp EE (Ensemble of Elastic Distances), phương pháp tổ hợp bộ phân lớp 1-NN kết hợp với tám độ đo khoảng cách đàn hồi. Bên cạnh đó, chúng tôi thực nghiệm so sánh tính hữu hiệu của cách hiện thực song song hóa dựa vào GPU với cách hiện thực tuần tự cho giải pháp đề xuất. Kết quả của thực nghiệm thứ hai cho thấy trung bình cách hiện thực giải pháp đề xuất bằng GPU thực thi nhanh gấp 48 lần so với cách hiện thực tuần tự cho giải pháp đề xuất.

Từ khóa— phân lớp, chuỗi thời gian, tổ hợp, 1-lân cận gần nhất, độ đo khoảng cách đàn hồi, độ đo khoảng cách không đàn hồi, GPU.

I. GIỚI THIỆU

Phân lớp chuỗi thời gian bao gồm việc huấn luyện một bộ phân lớp trên một tập mẫu mà mỗi mẫu là một chuỗi số thực được đo đạc tại những điểm thời gian cách đều nhau và một nhãn lớp. Phân lớp chuỗi thời gian là một bài toán quan trọng xuất hiện trong nhiều ứng dụng thực tế.

Do những đặc điểm riêng của thể loại dữ liệu chuỗi thời gian, nhiều giải thuật phân lớp nổi tiếng làm việc rất hiệu quả trên dữ liệu thông thường, lại không thể làm việc hiệu quả trên dữ liệu chuỗi thời gian. Trong những thập niên vừa qua có nhiều giải thuật được đề xuất để cải thiện hiệu năng của các mô hình phân lớp chuỗi thời gian tiêu biểu [1], [2].

Các phương pháp phân lớp chuỗi thời gian được phân loại thành ba nhóm chính: nhóm *dựa vào đặc trưng* (feature-based), nhóm *dựa vào mô hình* (model-based) và nhóm *dựa vào khoảng cách* (distance-based) [3]. Trong nhóm phương pháp phân lớp dựa vào đặc trưng, mỗi chuỗi thời gian được biến đổi thành tập *vector đặc trưng* (feature vector) và sau đó được phân lớp bằng một phương pháp phân lớp thông thường. Phương pháp phân lớp dựa vào mô hình giả định rằng tất cả các chuỗi thời gian thuộc về cùng một lớp là được sinh ra từ một mô hình và như vậy một chuỗi thời gian mới sẽ được gán vào một lớp có mô hình khớp với nó nhất. Sau cùng, phương pháp phân lớp dựa vào khoảng cách là phương pháp sử dụng một *độ đo khoảng cách* (distance measure) nào đó để quyết định một chuỗi thời gian mới thuộc về một lớp nào đó, thí dụ như phương pháp phân lớp *k-lân cận gần nhất* (k-nearest-neighbor-kNN).

Để so sánh chuỗi thời gian, đã có nhiều độ đo khoảng cách khác nhau được đề xuất và đánh giá hiệu quả thông qua bài toán phân lớp chuỗi thời gian [4]. Chúng bao gồm những biến thể của độ đo Xoắn thời gian động (Dynamic-Time-Warping-DTW), thí dụ như DTW Dẫn xuất (Derivarive DTW) [5], DTW với Cận dưới (lower bound) [6], DTW có trọng số (weighted DTW)[7], và những độ đo dựa vào *khoảng cách biên tập* (edit distance), bao gồm độ đo *Chuỗi con chung dài nhất* (longest common subsequence-LCS) [8], độ đo *Biên tập có hệ số phạt* (edit distance with real penalty) [9], độ đo *Xoắn thời gian với biên tập* (time warp with edit) [10], và độ đo *Di chuyển-tách-trộn* (move-split-merge) [11]. Tất cả những độ đo khoảng cách nêu trên đều là *độ đo khoảng cách đàn hồi* (elastic distance measure). Ngoài ra, vài biến thể của độ đo Euclid cũng đã được đề xuất, như độ đo khoảng cách *Bất biến độ phức tạp* (Complexity Invariant Distance - CID)[12] và độ đo khoảng cách *Hệ số nén* (Compression Rate Distance - CRD) [13].

Do *tổ hợp bộ phân lớp* (ensemble of classifiers) có thể đem lại sự cải thiện về độ chinh xác phân lớp, gần đây một số cách tiếp cận tổ hợp đã được đề xuất trong các công trình nghiên cứu về phân lớp chuỗi thời gian, thí dụ như phương pháp EE (Ensemble of Elastic Distances) của Lines và các cộng sự [14], phương pháp của Tan và các

cộng sự [15]. Một cách tiếp cận tổ hợp tiêu biểu, được đề xuất năm 2015, bởi Lines và các cộng sự [14], là sự kết hợp các bộ phân lớp 1-lân cận gần nhất (1-NN) mà chỉ sử dụng những độ đo khoảng cách đàn hồi.

Trong nghiên cứu này, chúng tôi đề xuất một phương pháp mới để phân lớp chuỗi thời gian bằng một tổ hợp sử dụng những độ đo khoảng cách khác nhau về thể loại. Phương pháp này có vẻ tương tự như hai phương pháp đi trước của Lines và các cộng sự [14], và của Tan và các cộng sự [15]. Tuy nhiên có hai điểm chính mà phương pháp của chúng tôi khác biệt với hai công trình đi trước ([14], [15]).

- Chúng tôi sử dụng một tổ hợp gồm 6 bộ phân lớp 1-NN mà kết hợp với cả những độ đo khoảng cách không đàn hồi và độ đo khoảng cách đàn hồi, khác với tổ hợp những bộ phân lớp 1-NN chỉ dùng 8 độ đo khoảng cách đàn hồi như trong [14], và [15]. Phương pháp đề xuất có tính tổng quát hơn và đưa vào tổ hợp *tính đa dạng* (diversity) nhằm cải thiện độ hiệu quả và tính hữu hiệu cho việc phân lớp chuỗi thời gian. Chúng tôi muốn kiểm tra xem liệu việc thay thế một số độ đo đàn hồi bằng những độ đo không đàn hồi có thể cải thiện tính hữu hiệu về thời gian của phương pháp tổ hợp mà không ảnh hưởng đến độ hiệu quả phân lớp.
- Sự hiện diện của những độ đo đàn hồi trong tổ hợp thường làm gia tăng chi phí tính toán của tổ hợp. Do đó chúng tôi thiết kế một cách hiện thực dựa vào GPU nhằm cải thiện tính hữu hiệu về thời gian cho phương pháp đề xuất thông qua việc song song hóa.

Như vậy ý tưởng chính của nghiên cứu này là kết hợp một giải pháp về phần mềm để nâng cao độ hiệu quả với một giải pháp về phần cứng để nâng cao tính hữu hiệu thời gian của một phương pháp phân lớp chuỗi thời gian .

Kết quả thực nghiệm trên nhiều bộ dữ liệu mẫu chuẩn cho thấy phương pháp đề xuất trong nghiên cứu này vượt trội hơn về độ hiệu quả so với phương pháp 1-NN kết hợp DTW, là phương pháp được giới học thuật liên quan xem là *giải thuật hàng đầu* (gold baseline) của bài toán phân lớp chuỗi thời gian [16], và đồng thời đem lại hiệu quả phân lớp ngang bằng với phương pháp tổ hợp EE sử dụng 8 độ đo khoảng cách đàn hồi [14]. Ngoài ra, việc so sánh tính hữu hiệu của cách hiện thực song song hóa dựa vào GPU với cách hiện thực tuần tự cho giải pháp đề xuất cũng được thực nghiệm. Kết quả của thực nghiệm thứ hai cho thấy trung bình cách hiện thực giải pháp đề xuất bằng GPU thực thi nhanh gấp 48 lần so với cách hiện thực tuần tự cho giải pháp đề xuất.

Phần tiếp theo của bài báo được tổ chức như sau. Mục II giới thiệu về một số khái niệm căn bản và các công trình liên quan. Mục III mô tả phương pháp tổ hợp đề xuất để phân lớp chuỗi thời gian. Mục IV trình bày cách hiện thực song song hóa dựa vào GPU cho phương pháp đề xuất. Mục V tường thuật kết quả thực nghiệm đánh giá hiệu quả của phương pháp đề xuất và tính hữu hiệu của cách hiện thực song song hóa dựa vào GPU cho phương pháp đề xuất. Mục VI nêu một vài kết luận và các hướng phát triển của đề tài.

II. CÁC KHÁI NIỆM CĂN BẢN VÀ CÁC CÔNG TRÌNH LIÊN QUAN

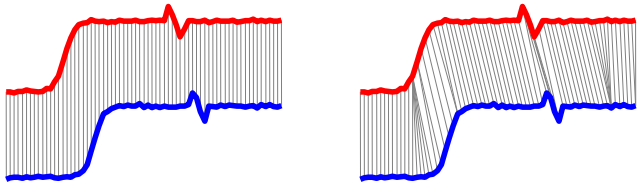
A. CÁC KHÁI NIỆM CĂN BẢN

Phương pháp một-lân cận gần nhất. Phương pháp một lân cận gần nhất (1-NN) phân lớp một mẫu mới bằng cách tính khoảng cách giữa mẫu này với mọi mẫu trong một tập mẫu đã được gắn nhãn lớp. Phương pháp này sẽ phân lớp mẫu mới vào lớp của mẫu trong tập huấn luyện mà gần với mẫu này nhất.

Độ đo khoảng cách. Hầu hết các bài toán khai phá dữ liệu chuỗi thời gian thường dùng những độ đo khoảng cách nào đó để định lượng mức độ tương tự hay khác biệt về thời gian của hai chuỗi thời gian. Những khoảng cách mà so sánh điểm thứ *i* của chuỗi thời gian này với điểm thứ *i* của chuỗi thời gian khác được gọi là những *độ đo không đàn hồi* (non-elasic measure), thí dụ như độ đo Euclid. Những độ đo mà cho phép ánh xạ từ một điểm đến nhiều điểm (one-to-many), thí dụ như độ đo DTW, hoặc cho phép ánh xạ từ một điểm đến nhiều điểm và từ một điểm đến không điểm (one-to-none) được gọi là những *độ đo đàn hồi* (elastic measure). Những đặc điểm chung của độ đo đàn hồi là chúng làm việc trên miền thời gian và có thể bù đắp cho sự cân chỉnh lệch cục bộ (localized misalignment) thông qua sự điều chỉnh có tính đàn hồi. Hình 1 minh họa sự khác biệt giữa độ đo khoảng cách không đàn hồi (độ đo Euclid) và độ đo khoảng cách đàn hồi (độ đo DTW).

Độ đo Euclid. Độ đo Euclid là độ đo thông dụng nhất trong khai phá dữ liệu chuỗi thời gian. Cho hai chuỗi thời gian có cùng chiều dài: $X = x_1, x_2,.., x_n$ and $Y = y_1, y_2,.., y_n$, khoảng cách Euclid giữa X và Y được định nghĩa bằng công thức sau đây:

$$Dist(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$



(a) Độ đo Euclid (b) Độ đo DTW

Hình 1. *Sự khác biệt giữa độ đo khoảng cách không đàn hồi (a) và độ đo khoảng cách đàn hồi (b).*

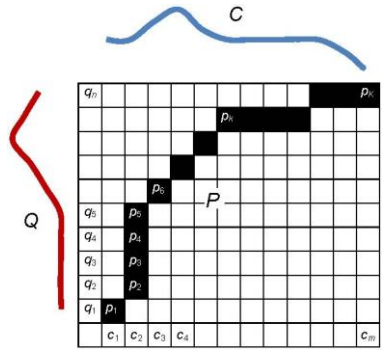
Độ đo Xoắn thời gian động. DTW là độ đo khoảng cách giữa hai chuỗi thời gian mà chúng ta có thể ánh xạ một điểm dữ liệu của chuỗi thời gian này đến nhiều hơn một điểm dữ liệu trên chuỗi thời gian khác. Nhằm đo lường sự tương tự về hình dạng, DTW là độ đo thường dùng để giảm thiểu sự vặn xoắn (distortion) trên trục thời gian [17].

Giả sử chúng ta có hai chuỗi thời gian $Q = q_1, q_2, \ldots, q_i, \ldots, q_n$ và $C = c_1, c_2, \ldots, c_j, \ldots, c_m$. Để tính khoảng cách DTW giữa hai chuỗi thời gian này, chúng ta xây dựng ma trận có kích thước $n \times m$, được gọi là *ma trận xoắn* (warping matrix), với $D_{ij} = d(q_i, c_j)$ là khoảng cách giữa hai điểm q_i và c_j (thí dụ: $d(q_i, c_j) = (q_i - c_j)^2$). Để tìm một ánh xạ *tốt nhất* giữa hai chuỗi, chúng ta phải tìm một lối đi xuyên qua ma trận sao cho *cực tiểu hóa* khoảng cách tích lũy giữa hai chuỗi. Một *lối đi xoắn* (warping path) W là một chuỗi tuần tự của các phần tử trong ma trận mà diễn tả một ánh xạ giữa hai chuỗi Q và C. Lối đi xoắn này có thể được tìm ra bằng cách sử dụng *quy hoạch động* (dynamic programming) để tính quan hệ truy hồi (recurrent relation) sau đây mà định nghĩa khoảng cách tích lũy $\gamma(i, j)$ bằng khoảng cách $d(i, j)$ tìm thấy tại ô hiện hành cộng với trị nhỏ nhất của khoảng cách tích lũy tại ba ô kế cận:

$$\gamma(i, j) = d(q_i, c_j) + \min\{\gamma(i-1, j), \gamma(i, j-1), \gamma(i-1, j-1)\}$$
$$\gamma(0, 0) = 0; \gamma(i, 0) = \gamma(0, j) = \infty; i = 1, 2, \ldots, n; j = 1, 2, \ldots, m \qquad (2)$$

trong đó $\gamma(i, j)$ bằng $d(i, j) = (q_i - c_j)^2$, bình phương khoảng cách giữa q_i và c_j, cộng với khoảng cách tích lũy nhỏ nhất của ba ô lân cận với (i, j). Khoảng cách DTW giữa hai chuỗi thời gian Q và C là căn hai của khoảng cách tích lũy tính tại ô cuối cùng (m, n).

Hình 2. minh họa lối đi xoắn (warping path), được tô đậm, trong quá trình tính khoảng cách DTW giữa hai chuỗi thời gian Q và C.



Hình 2. *Để tính khoảng cách giữa C và Q, một lối đi xoắn P, gồm những ô màu đen, được hình thành.*

Công thức quy hoạch động của độ đo DTW làm cho việc tính toán khoảng cách này rất tốn chi phí thời gian hơn là khoảng cách Euclid. Độ phức tạp tính toán của DTW là $O(n^2)$ trong khi độ phức tạp tính toán của độ đo Euclid là $O(n)$ với n là chiều dài của hai chuỗi thời gian. Lưu ý rằng nếu chúng ta giới hạn việc tìm kiếm lối đi xoắn trong tầm một *cửa sổ xoắn* (warping window) [6] thì chúng ta có thể làm giảm độ phức tạp tính toán của DTW xuống thành $O(nr)$, với r là kích thước của cửa sổ xoắn.

Độ đo DTW Dẫn xuất. Keogh và Pazzani, năm 2001, đề xuất một cải tiến của độ đo DTW được gọi là độ đo *DTW Dẫn Xuất* (Derivative Dynamic Time Warping -DDTW) [5]. Độ đo này trước tiên biến đổi chuỗi thời gian gốc thành chuỗi thời gian hiệu bậc nhất. Cho một chuỗi thời gian $t = \{t_1, t_2, \ldots, t_m\}$, chuỗi thời gian hiệu của chuỗi t là chuỗi $t' = \{t'_1, t'_2, \ldots, t'_{m-1}\}$ với t'_i được định nghĩa là trung bình độ dốc giữa t_{i-1} với t_i, và t_i với t_{i+1}, tức là:

$$t'_i = ((t_i - t_{i-1}) + ((t_{i+1} - t_{i-1})/2))/2 \qquad (3)$$

với $1 < i < m$. DTW Dẫn Xuất được định nghĩa để làm giảm ảnh hưởng của nhiễu trong chuỗi thời gian đối với độ đo DTW.

Độ đo DTW với Cận dưới. Một cách tiếp cận để tăng tốc việc tính độ đo DTW là thay thế phần lớn những tính toán khoảng cách DTW phức tạp bằng những tính toán cận dưới đơn giản hơn. Một kỹ thuật tính toán cận dưới nổi tiếng là cận dưới LB_Keogh (được đề xuất bởi Keogh and Ratanamahatana, năm 2005 [6]). Trong kỹ thuật tính cận dưới cho DTW, cận dưới của độ đo khoảng cách DTW giữa hai chuỗi thời gian Q và C, ký hiệu là $LB(Q, C)$, phải thỏa mãn bất đẳng thức: $LB(Q, C) \le DTW(Q, C)$.

Giả sử *best_so_far* là biến lưu khoảng cách giữa hai chuỗi thời gian mà tốt nhất (nhỏ nhất) cho đến hiện giờ, nếu $LB(Q, C) > best\_so\_far$, thì $DTW(Q, C) > best\_so\_far$ và do đó chúng ta không cần tính $DTW(Q, C)$ mà vẫn có thể kết luận rằng hai chuỗi thời gian Q và C không thể tương tự với nhau.

Độ đo Bất biến độ phức tạp. Batista và các cộng sự [12] đã đề xuất độ đo *Bất biến độ phức tạp* (Complexity-Invariant distance - CID), là một độ đo không đàn hồi. Độ đo này dùng thông tin về những khác biệt độ phức tạp tính toán của hai chuỗi gian như một hệ số điều chỉnh cho độ đo khoảng cách hiện có. Cho hai chuỗi thời gian Q và C, độ đo khoảng cách CID giữa Q và C được tính bằng công thức:

$$\mathrm{CID}(Q, C) = ED(Q, C) \times CF(Q,C) \qquad (4)$$

với $ED(Q, C)$ là khoảng cách Euclid giữa Q và C , và CF là hệ số điều chỉnh độ phức tạp mà được định nghĩa như sau:

$$CF(Q, C) = \frac{\max(CE(Q), CE(C))}{\min(CE(Q), CE(C))} \qquad (5)$$

trong đó $CE(T)$ là giá trị ước lượng độ phức tạp của chuỗi thời gian T, mà được tính bằng công thức sau.

$$CE(Q) = \sqrt{\sum_{i=1}^{n-1}(q_i - q_{i+1})^2} \qquad (6)$$

Chú ý rằng vì độ đo CID dựa vào độ đo Euclid, nên để tính độ đo CID của hai chuỗi thời gian, chúng ta giả định rằng hai chuỗi này phải có cùng chiều dài.

Độ đo Hệ số nén. Vinh và Anh năm 2015 [13] đã đề xuất độ đo Hệ Số Nén (Compression Rate Distance -CRD), là một độ đo khoảng cách không đàn hồi. Ý tưởng chính của độ đo khoảng cách này là dựa vào nguyên lý *Chiều dài mô tả tối thiểu* (Minimum Description Length (MDL) principle) trong lý thuyết thông tin. Hệ số nén giữa hai chuỗi thời gian càng cao thì hai chuỗi này càng tương tự nhau.

Cho hai chuỗi thời gian $Q = q_1, q_2, \ldots, q_n$ và $C = c_1, c_2, \ldots, c_n$. Độ đo CRD giữa Q và C được định nghĩa như sau:

$$CRD(Q,C) = CR(Q,C)^{\alpha} \times ED(Q,C)$$

với CR là *hệ số nén* (compression rate), α là *độ lệch* (bias) mà là một số thực lớn hơn hay bằng 0, giá trị α càng lớn thì ảnh hưởng của hệ số nén lên khoảng cách càng nhiều, và ED là độ đo Euclid. Hệ số nén CR được định nghĩa như sau:

$$CR(Q,C) = \frac{DL(Q-C)}{\min\{DL(Q), DL(C)\} + \varepsilon} \qquad (7)$$

trong đó ε là một trị cực nhỏ nhằm tránh phép chia zero, và DL là *chiều dài mô tả* (description length) của chuỗi thời gian.

Chiều dài mô tả DL của một chuỗi thời gian T là tổng số bit cần dùng để biểu diễn chuỗi thời gian.

$$DL(T) = w \times E(T) \qquad (8)$$

với w là chiều dài của chuỗi T và $E(T)$ là entropy của chuỗi T.

Vì chiều dài của hai chuỗi thời gian bằng nhau, nên hệ số nén được xấp xỉ bằng công thức sau đây:

$$CR(Q,C) = \frac{E(Q-C)}{\min\{E(Q), E(C)\} + \varepsilon} \qquad (9)$$

Entropy $E(T)$ của một chuỗi thời gian T được định nghĩa như sau:

$$E(T) = -\sum_{t} P(T=t)\log_2 P(T=t) \qquad (10)$$

với $P(T = t)$ là xác xuất để trị t có xuất hiện trong chuỗi thời gian T.

Chú ý rằng vì độ đo CRD dựa vào độ đo Euclid, nên để tính độ đo CRD của hai chuỗi thời gian, chúng ta giả định rằng hai chuỗi này phải có cùng chiều dài.

Vinh và Anh [13] thực nghiệm độ đo CRD vào bài toán phân lớp chuỗi thời gian với 45 bộ dữ liệu mẫu và so sánh với ba độ đo thông dụng khác là độ đo Euclid (ED), độ đo Bất Biến Độ Phức Tạp (CID) và độ đo Xoắn Thời gian động (DTW). Kết quả thực nghiệm cho thấy độ đo CRD vượt trội hơn độ đo ED và độ đo CID trên phần lớp các bộ dữ liệu mẫu thử nghiệm. Bên cạnh đó, độ đo CRD có hiệu quả phân lớp tốt hơn độ đo DTW trên 2/3 tổng số bộ dữ liệu được thử nghiệm.

B. CÁC CÔNG TRÌNH LIÊN QUAN

Lines và các cộng sự, năm 2015 [14] đã đề xuất một phương pháp tổ hợp để phân lớp chuỗi thời gian bao gồm tám bộ phân lớp 1-NN kết hợp với tám độ đo khoảng cách đàn hồi: độ đo khoảng cách Xoắn thời gian động (DTW), độ đo DTW Dẫn xuất (Derivative Dynamic Time Warping -DDTW) [5], độ đo DTW Có trọng số (Weighted Dynamic Time Warping - WDTW) [7], độ đo DTW Dẫn xuất có trọng số (Weighted Derivative Dynamic Time Warping - WDDTW), độ đo Chuỗi con chung dài nhất (Longest Common Subsequence -LCS) [8], độ đo Biên tập có mức phạt (Edit Distance with Real Penalty -ERP) [9], độ Đo xoắn thời gian biên tập (Time Warp Edit Distance - TWED) [10] và độ đo Di chuyển-tách- gộp (Move-Split-Merge - MSM) [11]. Các tác giả xây dựng mỗi bộ phân lớp thành phần một cách độc lập và áp dụng chiến lược *phiếu bầu tỉ lệ thuận* (proportional voting scheme) khi sử dụng các kết quả phân lớp từ các bộ phân lớp thành phần. Chiến lược phiếu bầu tỉ lệ thuận gán trọng số cho từng bộ phân lớp thành phần dựa trên độ chính xác phân lớp được đánh giá bằng *kiểm tra chéo* (cross-validation) mà được chuẩn hóa qua số lần thực nghiệm. Điều đó có nghĩa là các trọng số được gán đến các bộ phân lớp thành phần được tính dựa vào hiệu quả của chúng khi làm việc trên tập huấn luyện trong quá trình kiểm tra chéo. Kết quả thực nghiệm trên 46 bộ dữ liệu mẫu lấy từ kho dữ liệu mẫu UCR cho thấy phương pháp tổ hợp này có hiệu quả vượt trội phương pháp 1-NN kết hợp độ đo DTW, một phương pháp phân lớp chuỗi thời gian mà đã được đánh giá là "khó bị đánh bại"("difficult to beat"). Phương pháp tổ hợp gồm 8 độ đo đàn hồi nêu trên (được ký hiệu là phương pháp EE), là phương pháp phân lớp đầu tiên hiệu quả hơn phương pháp 1-NN kết hợp với độ đo DTW.

Tan và các cộng sự, năm 2020 [15] đề xuất một biến thể cải tiến cho phương pháp tổ hợp được đưa ra bởi Lines và các cộng sự [14]. Biến thể cải tiến trong [15], được gọi là phương pháp FastEE, định nghĩa các cận dưới mới cho các độ đo WDTW, MSM và TWED, và áp dụng một phương pháp hữu hiệu [18] để ước lượng kích thước cửa sổ xoắn (warping window) cho cả tám độ đo đàn hồi nhằm cải tiến tính hữu hiệu thời gian cho phương pháp tổ hợp ở công trình [14]. Thời gian huấn luyện của phương pháp FastEE nhanh hơn gần 10 lần so với phương pháp EE ở công trình [14].

III. TỔ HỢP BỘ PHÂN LỚP VỚI CÁC ĐỘ ĐO KHOẢNG CÁCH KHÔNG ĐÀN HỒI VÀ ĐÀN HỒI

Trong nghiên cứu này, phương pháp tổ hợp được đề xuất gồm 6 bộ phân lớp 1-NN, mỗi bộ phân lớp kết hợp với một độ đo khoảng cách khác biệt nhau. Sáu độ đo khoảng cách được dùng trong phương pháp tổ hợp này là độ đo Euclid, độ đo DTW, độ đo DTW Dẫn Xuất [5], độ đo DTW với cận dưới LB-Keogh [6], độ đo CID [12] và độ đo CRD [13]. Trong số sáu độ đo này, độ đo Euclid, độ đo CID và độ đo CRD là ba độ đo không đàn hồi (còn được gọi là *độ đo khóa từng bước* (lock-step measure)) trong khi ba độ đo kia là những độ đo đàn hồi. Lưu ý rằng các độ đo không đàn hồi thường dễ hiện thực và hữu hiệu về thời gian tính toán hơn là những độ đo đàn hồi. Để tiện lợi, chúng tôi ký hiệu sáu bộ phân lớp thành phần như sau: ED (1-NN kết hợp độ đo Euclid), DTW (1-NN kết hợp độ đo DTW distance), CDTW (1-NN kết hợp độ đo DTW dùng cận dưới LB_Keogh), DDTW (1-NN kết hợp độ đo DTW Dẫn Xuất), CID (1-NN kết hợp độ đo CID) and CRD (1-NN kết hợp độ đo CRD).

Giai đoạn huấn luyện: Trong giai đoạn này, phương pháp đề xuất sẽ tính toán và gán trọng số cho từng bộ phân lớp thành phần dựa vào tỉ lệ phân lớp sai của mỗi bộ phân lớp. Nhưng tỉ lệ phân lớp sai của mỗi bộ phân lớp thành phần phải được chuẩn hóa theo công thức sau:

$$w_j = \frac{error\_rate_j}{\sum_{j=1}^{n} error\_rate_j} \qquad (11)$$

với *n* là tổng số bộ phân lớp thành phần trong tổ hợp.

Tỉ lệ phân lớp sai (error rate) được tính theo công thức sau:

$$Error\ rate = n/N \qquad (12)$$

với *n* là tổng số mẫu bị phân lớp sai và *N* là tổng số mẫu trong tập huấn luyện.

Toàn bộ quá trình của giai đoạn huấn luyện được mô tả ở Algorithm 1 như sau.

Algorithm 1

Input: D is the dataset,
Output: The array *weight* with weight[C_j] for each classifier C_j
Divide D into two subsets: *Train* and *Test*
Divide *Train* into six subsets, termed as $Train_j$
sum = 0
for j:= 1 **to** 6 **do**
compute ER[C_j] for each classifier C_j in the dataset $Train_j$
// ER[C_j] is error rate of classifier C_j, using Eq. (12)
sum = sum + ER[C_j]
endfor
for j:= 1 **to** 6 **do**
weight[C_j] = ER[C_j]/sum

Lưu ý: Algorithm 1 có thể được cải biên khi sử dụng trong cho việc kiểm tra chéo k-phần (k-fold cross-validation).

Giai đoạn thử: Trong giai đoạn này, kết quả phân lớp của mỗi mẫu trong tập thử được quyết định bởi mỗi bộ phân lớp thành phần và được tổng hợp lại từ những quyết định của mọi bộ phân lớp thành phần. Trong phương pháp tổ hợp của chúng tôi, kết quả cuối cùng được xác định dựa vào trọng số của một bộ phân lớp đặc biệt. Đối với một mẫu thử t, kết quả cuối cùng $F(t)$ của mẫu thử t được tính như sau:

$$F(t) = f_j(t) \mid \text{weight}[C_j] \text{ là nhỏ nhất} \qquad (13)$$

Tức là, kết quả cuối cùng của mẫu thử đạt được từ bộ phân lớp (thứ j) mà có trọng số nhỏ nhất.

Quá trình của giai đoạn thử được mô tả ở Algorithm 2, được trình bày như sau.

Algorithm 2
Input: *Test* is the test dataset,
Output: classified *Test* set
for each x in *Test*
for j: = 1 **to** 6
obtain the predicted label as $y(x) = f_j(x)$
//$f_j(x)$ is the predicted label obtained from C_j for x
endfor
determine the final outcome, $F(x)$ by using Eq. (13)
endfor

IV. HIỆN THỰC SONG SONG HÓA DỰA VÀO GPU CHO PHƯƠNG PHÁP TỔ HỢP ĐỀ XUẤT

A. SONG SONG HÓA DỰA VÀO GPU

Đơn vị xử lý đồ họa (Graphics Processing Unit- GPU) là một công cụ tính toán được thiết kế để xử lý những tính toán đồng thời và song song [19], [20], [21]. Các công cụ GPU được thiết kế với kiến trúc tính toán song song cao cấp mà tạo điều kiện cho những công cụ này thực hiện rất nhiều tính toán một cách đồng thời, và làm cho chúng thích hợp với những tác vụ tính toán song song.

Để vận dụng song song hóa dựa vào một GPU, tải công việc được chia thành nhiều *luồng* (thread), là những đơn vị thực thi căn bản. Những luồng như vậy được gom nhóm thành những *khối* (block), và nhiều khối như vậy hình thành nên một *lưới* (grid). Mỗi luồng được gán một *định danh* (identifier) đơn nhất, cho phép nó xử lý một phần chuyên biệt của tập dữ liệu một cách độc lập. Bộ xếp lịch GPU quản lý sự thực thi của những luồng này, vận dụng một cách hiệu quả khả năng xử lý song song của GPU.

Song song hóa dựa vào GPU có nhiều ưu điểm. Một là, nó cho phép xử lý hữu hiệu những lượng dữ liệu lớn bằng cách song song hóa, đem lại một sự cải thiện hiệu năng đáng kể so với việc xử lý tuần tự lượng dữ liệu này trên một CPU (Central Processing Unit). Hai là, số lượng lớn các *lõi* (core) tính toán trên một GPU tạo điều kiện cho việc thực thi nhiều luồng một cách đồng thời , cực đại hóa hiệu năng tính toán. Ba là, các GPU được tối ưu hóa một cách chuyên biệt cho các tải công việc có tính song song, làm cho chúng đặc biệt ưu việt đối với những bài toán có qui mô lớn về dữ liệu và có các phép tính toán toán học phức tạp.

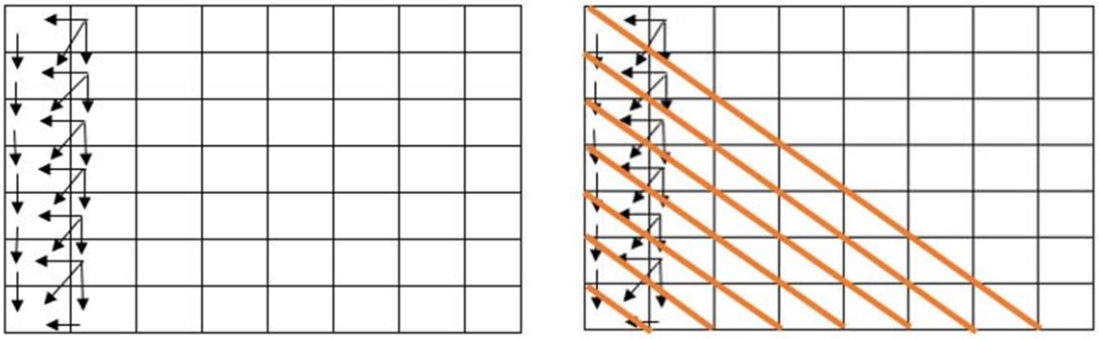
Trong ngữ cảnh của phân tích chuỗi thời gian, song song hóa trên một GPU rất hữu ích cho những công tác như phát hiện bất thường, pháp hiện motif, phân lớp, gom cụm và tìm kiếm tương tự. Bằng cách vận dụng khả năng tính toán song song của GPU, các công tác tính toán như vậy được thực thi hữu hiệu hơn, đem lại một sự phân tích dữ liệu nhanh hơn và cải tiến hiệu năng toàn cục.

B. HIỆN THỰC SONG SONG HÓA DỰA VÀO GPU CHO PHƯƠNG PHÁP TỔ HỢP ĐỀ XUẤT

Việc xây dựng các bộ phân lớp thành phần của phương pháp tổ hợp được thực hiện một cách tuần tự. Mỗi bộ phân lớp thành phần 1-NN lại có thể được thực hiện một cách song song thông qua việc song song hóa các bước tính toán độ đo khoảng cách trong giải thuật 1-lân cận gần nhất. Mỗi bộ phân lớp 1-NN hiện thực dựa vào GPU sẽ gán các luồng đảm nhiệm việc tính các độ đo khoảng cách một cách đồng thời rất đơn giản: mỗi luồng trong GPU đảm nhiệm việc tính độ đo khoảng cách giữa chuỗi thời gian thử và mỗi chuỗi thời gian trong tập huấn luyện.

Đặc biệt khi độ đo khoảng cách là độ đo DTW và độ đo DTW Dẫn Xuất, do mối quan hệ truy hồi và tính *phụ thuộc dữ liệu* (data dependency) giữa các phần tử (của ma trận xoắn) trong quá trình tính toán, chúng ta phải phân tích tính phụ thuộc dữ liệu trong giải thuật tính toán và thiết kế một chiến lược cải tiến mức độ song song hóa. Kỹ thuật song song hóa mà chúng tôi áp dụng trong nghiên cứu này là dựa vào chiến lược song song hóa *Dạng Sóng* (Waveform) dành cho GPU được đề xuất bởi Belviranli và các cộng sự, năm 2015 [22]. Kỹ thuật này được mô tả chi tiết như sau.

Hình 3 minh họa tính phụ thuộc dữ liệu trong khi tính toán độ đo DTW. Trong ma trận xoắn, mỗi ô biểu diễn độ đo DTW giữa hai điểm. Việc tính toán tại mỗi ô phụ thuộc vào kết quả tại các ô bên trái, ô bên dưới, và ô chéo bên dưới. Tức là, để tính ô (i, j) trong ma trận D, giá trị tại ô $D[i, j]$ phụ thuộc vào giá trị tại ba ô $D[i\text{-}1, j]$, $D[i, j\text{-}1]$, $D[i\text{-}1, j\text{-}1]$ (xem công thức (2) ở tiểu mục II.A).



(a) Sự phụ thuộc tính toán (b) Cập nhật ma trận theo những đường chéo

Hình 3. *Thứ tự cập nhật ma trận tính độ đo*

Trong giải thuật DTW cổ điển, quá trình tính toán có thể thực hiện theo thứ tự hàng hoặc thứ tự cột. Những ô trên cùng một hàng hoặc cùng một cột sẽ được tính một cách tuần tự. Tuy nhiên, không hề có sự phụ thuộc dữ liệu giữa các ô ở trên cùng một đường chéo như được minh họa trong Hình 3.b. Do đó, những ô này có thể được tính một cách song song (mức độ song song bậc hai). Có sự phụ thuộc dữ liệu giữa hai đường chéo kế cận nhau. Như vậy, tại một lúc, chúng ta có thể tính dữ liệu trong cùng một đường chéo. Trong Hình 3, mũi tên biểu thị sự phụ thuộc dữ liệu và hướng tính toán.

Dựa vào quan sát trên, chúng tôi đi đến một chiến lược song song như sau. Quá trình tính toán bắt đầu từ ô thứ nhất (ô ở vị trí (0,0)), rồi cập nhật các ô trên đường chéo gần với ô này nhất. Vì những ô này chỉ phụ thuộc vào ô thứ nhất nên chúng có thể được tính song song. Sau đó quá trình tính toán được chuyển qua đường chéo kế cận. Bằng cách cập nhật ma trận theo cách thức như vậy, sự phụ thuộc dữ liệu được tôn trọng và một mức độ song song hóa hợp lý đã được thực hiện.

V. KẾT QUẢ THỰC NGHIỆM

Mục này mô tả các thực nghiệm trên nhiều bộ dữ liệu mẫu để đánh giá hiệu quả của phương pháp tổ hợp đề xuất cho bài toán phân lớp chuỗi thời gian.

Các phương pháp phân lớp đối sánh được hiện thực bằng ngôn ngữ R và C++ và tiến hành các thực nghiệm trên máy tính có cấu hình Intel® Core i7-6700k 4.00 GHz CPU, 32 GB RAM PC. Để sử dụng công cụ GPU, chúng tôi sử dụng NVIDIA GTX 960 GPU. Bảng 1 mô tả các thông số kỹ thuật của NVIDIA GTX 960 GPU.

Các thực nghiệm nhằm so sánh các phương pháp đối sánh dựa vào độ chính xác phân lớp và độ hữu hiệu thời gian.

Chúng tôi sử dụng *tỉ lệ phân lớp sai* (classification error rate) làm tiêu chí đánh giá về độ chính xác phân lớp trong các thực nghiệm.

Các thực nghiệm được tiến hành trên có bộ dữ liệu mẫu lấy từ kho dữ liệu mẫu về phân lớp chuỗi thời gian của UCR (UCR Time Series Classification Archive) [23]. Có tất cả 14 bộ dữ liệu mẫu được dùng trong các thực nghiệm này. Số thứ tự, thể loại và tên của từng bộ dữ liệu được cho trong Bảng 2.

Bảng 1. *Các thông số kỹ thuật của Nvidia GTX 960 GPU*

Nvidia GTX 960	
SMs	8
CUDA Cores	1024
Memory Size	4 GB
Peak Performance (single precision)	2.644 TFLOPS
Peak Performance (double precision)	82.62 GFLOPS
Memory Bandwidth	112.2 GB/s

Bảng 2. *Các bộ dữ liệu thử nghiệm*

Số thứ tự	Thể loại	Tên bộ dữ liệu
1	Image	ArrowHead
2	Spectro	Beef
3	Image	BeetleFly
4	Simulated	CBF
5	Image	DistalPhalanxOutlineCorrect
6	Image	Fish
7	Motion	GunPoint
8	Spectro	Ham
9	Image	FaceFour
10	Sensor	Trace
11	ECG	ECGFiveDays
12	Traffic	Chinatown
13	HRM	Fungi
14	EPG	InsectEPGRegularTrain

Bảng 3. *Chi tiết về* 14 *bộ dữ liệu thử nghiệm*

Số thứ tự	Tập huấn luyện	Tập thử	Số lớp	Chiều dài
1	39	175	3	251
2	30	30	5	470
3	20	20	2	512
4	30	900	3	128
5	600	276	2	80
6	175	175	7	463
7	50	150	2	150
8	109	105	2	431
9	24	88	4	350
10	100	100	5	275
11	23	861	2	136
12	20	343	2	24
13	18	186	18	201
14	62	249	3	601

Mỗi bộ dữ liệu được chia thành hai tập con: tập huấn luyện và tập thử. Bảng 3 mô tả số lượng mẫu trong tập huấn luyện và trong tập thử cùng với chiều dài của mỗi mẫu (là một chuỗi thời gian) trong mỗi bộ dữ liệu.

Để đánh giá chất lượng phân lớp của phương pháp đề xuất, chúng tôi sử dụng kỹ thuật *kiểm tra chéo 10 phần* (10-fold cross-validation) được mô tả như sau. Tại lượt lặp thứ k ($0 \le k \le 10$), tập thử bao gồm phần mười thứ k của tập dữ liệu và tất cả những phần mười còn lại được dùng cho tập huấn luyện. Quá trình thực nghiệm thực hiện 10 lần thử nghiệm với 10 tập huấn luyện và tập thử khác nhau để đánh giá hiệu quả phân lớp. Hiệu quả cuối cùng được tính bằng cách lấy trung bình 10 tỉ lệ phân lớp sai của 10 lần thử nghiệm.

Thực nghiệm 1 so sánh phương pháp tổ hợp đề xuất với 6 phương pháp phân lớp: 1-NN kết hợp với một trong sáu độ đo khoảng cách. Chúng tôi ký hiệu những phương pháp phân lớp này như sau:

ED: giải thuật 1-lân cận gần nhất với độ đo khoảng cách Euclid.

DTW: giải thuật 1-lân cận gần nhất với độ đo khoảng cách DTW trực tiếp.

CDTW: giải thuật 1-lân cận gần nhất với độ đo khoảng cách DTW có dùng cận dưới LB_Keogh.

DDTW: giải thuật 1-lân cận gần nhất với độ đo khoảng cách DTW Dẫn xuất.

CID: giải thuật 1-lân cận gần nhất với độ đo khoảng cách Bất biến độ phức tạp (CID).

CRD: giải thuật 1-lân cận gần nhất với độ đo khoảng cách Hệ số nén (CRD).

Phương pháp tổ hợp đề xuất được thực hiện dưới hai dạng thức, được ký hiệu như sau:

ENE1: phương pháp tổ hợp gồm 5 bộ phân lớp 1-NN với 5 độ đo khoảng cách (ED, DTW, DTW_LB, DDTW, và CID). Như vậy, trong ENE1, có hai bộ phân lớp với độ đo khoảng cách không đàn hồi (ED, CID) và ba độ đo khoảng cách đàn hồi (DTW, CDTW và DDTW).

ENE2: phương pháp tổ hợp gồm 6 bộ phân lớp 1-NN với 6 độ đo khoảng cách (ED, DTW, DTW_LB, DDTW, CID và CRD). Như vậy, trong ENE2, có ba bộ phân lớp với độ đo khoảng cách không đàn hồi (ED, CID, CRD) và ba bộ phân lớp với độ đo khoảng cách đàn hồi (DTW, CDTW và DDTW).

A. THỰC NGHIỆM 1

Thực nghiệm 1 nhằm mục đích so sánh độ chính xác phân lớp của 8 phương pháp phân lớp chuỗi thời gian nêu trên.

Bảng 4 và Bảng 5 trình bày tỉ lệ phân lớp sai, sử dụng công thức (12), của 8 phương pháp phân lớp trên 14 bộ dữ liệu mẫu. Những tỉ lệ phân lớp sai nhỏ nhất sẽ được in đậm.

Trong số 14 bộ dữ liệu mẫu được dùng trong thực nghiệm 1, có 7 bộ dữ liệu trùng khớp với những bộ dữ liệu được dùng trong thực nghiệm chính của công trình nghiên cứu bởi Lines và các công sự [14]. Do đó, tỉ lệ phân lớp sai của phương pháp tổ hợp chúng tôi đề xuất sử dụng 6 độ đo khác nhau (đàn hồi và không đàn hồi) có thể so sánh với tỉ lệ phân lớp sai của phương pháp tổ hợp EE ở công trình sử dụng 8 độ đo đàn hồi [14] và kết quả so sánh được trình bày ở Bảng 6. Những tỉ lệ phân lớp sai nhỏ nhất trong Bảng 6 sẽ được in đậm.

Từ những kết quả thực nghiệm trong Bảng 4, Bảng 5 và Bảng 6, chúng ta có thể rút ra những nhận xét sau đây:

- 1-NN với độ đo CRD (độ đo không đàn hồi) vượt trội hơn 1-NN với độ đo DTW (độ đo đàn hồi) trên 11 trong số 14 bộ dữ liệu mẫu. Nhận xét này chỉ ra rằng không phải độ đo đàn hồi nào cũng vượt trội hơn mọi độ đo không đàn hồi trên mọi bộ dữ liệu khi phân lớp chuỗi thời gian.
- Phương pháp tổ hợp đề xuất (ENE2) đem lại tỉ lệ phân lớp sai nhỏ nhất (tức là độ chính xác phân lớp cao nhất) trong số 8 phương pháp phân lớp được so sánh trên 14 bộ dữ liệu mẫu.
- Phương pháp tổ hợp đề xuất (ENE2) vượt trội hơn phương pháp 1NN với độ đo DTW trên 14 bộ dữ liệu mẫu. Sự kiện này hàm ý rằng phương pháp tổ hợp chúng tôi đề xuất cũng đạt được cùng mục tiêu mà phương pháp tổ hợp EE của Lines và các cộng sự [14] đạt được: là phương pháp phân lớp chuỗi thời gian đầu tiên thắng được phương pháp chuẩn vàng (gold baseline) 1-NN kết hợp DTW.
- So sánh kết quả thực nghiệm của ENE1 và ENE2 cho thấy việc thêm bộ phân lớp thành phần 1-NN với độ đo CRD (là độ đo không đàn hồi) vào phương pháp tổ hợp ENE1 để hình thành phương pháp tổ hợp EN2 đã cải thiện đáng kể độ chính xác phân lớp của tổ hợp trước đó, là ENE1. Điều này nói lên sự đóng góp đáng kể của bộ phân lớp 1-NN kết hợp độ đo CRD đối với tổ hợp bộ phân lớp ENE2.
- Phương pháp tổ hợp đề xuất (ENE2) có độ chính xác phân lớp tốt hơn phương pháp tổ hợp EE (Elastic Ensemble) trong công trình [14] trên 3 trong số 7 bộ dữ liệu mẫu (xem Bảng 5). Điều này cho thấy độ chính xác phân lớp của ENE2 gần như ngang bằng với độ chính xác phân lớp của EE mặc dù EE bao gồm 8 bộ phân lớp với 8 độ đo đàn hồi trong khi ENE2 chỉ gồm 6 bộ phân lớp với 3 độ đo không đàn hồi và 3 độ đo đàn hồi. Điều này hàm ý rằng những độ đo không đàn hồi vẫn có thể đóng góp tốt vào

công việc phân lớp ngang ngửa với sự đóng góp của các độ đo đàn hồi trong khi những độ đo không đàn hồi thường có độ phức tạp tính toán thấp hơn những độ đo đàn hồi.

Bảng 4. *Tỉ lệ phân lớp sai của 8 phương pháp trên các bộ dữ liệu 1-8*

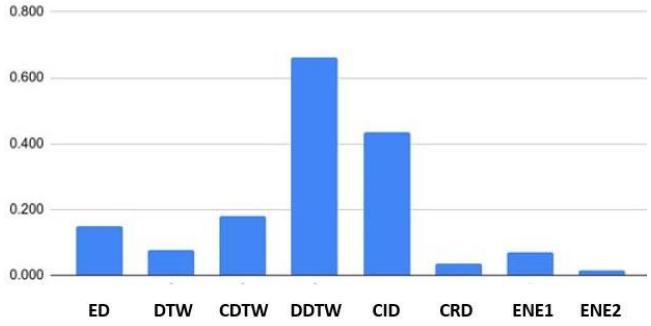
	Bộ dữ liệu 1	Bộ dữ liệu 2	Bộ dữ liệu 3	Bộ dữ liệu 4	Bộ dữ liệu 5	Bộ dữ liệu 6	Bộ dữ liệu 7	Bộ dữ liệu 8
ED	0.200	0.333	0.250	0.148	0.283	0.217	0.087	0.400
DTW	0.366	0.533	0.350	0.077	0.333	0.874	0.293	0.457
CDTW	0.474	0.500	0.350	0.179	0.464	0.697	0.153	0.457
DDTW	0.446	0.367	0.150	0.661	0.254	0.857	0.033	0.590
CID	0.177	0.400	0.300	0.437	0.283	0.857	0.087	0.448
CRD	0.206	0.333	0.250	0.037	0.261	0.206	0.047	0.486
ENE1	0.106	0.156	0.096	0.070	0,156	0.061	0.076	0.170
ENE2	**0.095**	**0.135**	**0.091**	**0.016**	**0.135**	**0.055**	**0.029**	**0.141**

Bảng 5. *Tỉ lệ phân lớp sai của 8 phương pháp trên các bộ dữ liệu 9-14*

	Bộ dữ liệu 9	Bộ dữ liệu 10	Bộ dữ liệu 11	Bộ dữ liệu	Bộ dữ liệu	Bộ dữ liệu 14
ED	0.216	0.240	0.203	0.047	0.172	0.000
DTW	0.159	0.040	0.322	0.044	0.253	0.000
CDTW	0.455	0.250	0.178	0.023	0.339	0.000
DDTW	0.409	0.060	0.321	0.160	0.371	0.281
CID	0.148	0.150	0.233	0.047	0.172	0.000
CRD	0.193	0.010	0.211	0.029	0.134	0.000
ENE1	0.106	0.027	0.110	0.026	0.131	0.000
ENE2	**0.094**	**0.007**	**0.102**	**0.014**	**0.093**	**0.000**

Bảng 6. *Tỉ lệ phân lớp sai của hai phương pháp ENE2 và EE trên bảy bộ dữ liệu*

Bộ dữ liệu	Phương pháp đề xuất (ENE2)	Phương pháp [14] (EE)
Beef	**0.135**	0.367
CBF	0.016	**0.002**
Fish	0.055	**0.034**
GunPoint	0.029	**0.007**
FaceFour	0.094	**0.091**
Trace	**0.007**	0.010
ECGFiveDays	**0.102**	0.178



Hình 4. *Tỉ lệ phân lớp sai của 8 phương pháp trên bộ dữ liệu CBF*

Hình 4 minh họa kết quả thực nghiệm của 8 phương pháp phân lớp trên một bộ dữ liệu đặc biệt: CBF. Từ tỉ lệ phân lớp sai của 8 phương pháp trên bộ dữ liệu mẫu CBF, chúng ta có thể rút ra hai nhận xét sau đây:

- Với bộ dữ liệu đặc biệt này, bộ phân lớp 1-NN với hai độ đo đàn hồi: 1NN+DDTW và 1NN+CDTW đem lại hiệu quả phân lớp kém hơn hai độ đo không đàn hồi: 1NN+ED và 1NN+CRD.
- Với bộ dữ liệu đặc biệt này, bộ phân lớp 1NN + CRD tốt hơn tất cả các bộ phân lớp 1-NN với mọi độ đo không đàn hồi và đàn hồi khác (ED, CID, DTW, DDTW, CDTW).

B. THỰC NGHIỆM 2

Trong Thực Nghiệm 2, để kiểm tra thời gian thực thi của việc hiện thực phương pháp đề xuất bằng GPU, chúng tôi đo đạc thời gian thực thi của hai phiên bản CPU và GPU của phương pháp tổ hợp ENE2 trên 14 bộ dữ liệu mẫu.

Kết quả thực nghiệm (tính bằng giây) được trình bày ở Bảng 7. *Độ tăng tốc* (speed-up) của phiên bản GPU so với phiên bản CPU trên từng bộ dữ liệu mẫu được nêu ở cột thứ tư cuả Bảng 7. Chúng ta có thể thấy trung bình cách hiện thực giải pháp đề xuất bằng GPU thực thi nhanh gấp 48 lần so với cách hiện thực tuần tự cho giải pháp đề xuất.

Bảng 7. *Thời gian thực thi (tính bằng giây) của hai cách hiện thực (CPU và GPU) của phương pháp ENE2*

Bộ dữ liệu	ENE2		
	CPU	GPU	Độ tăng tốc
1	304.56	**6.63**	46
2	85.92	**2.14**	40
3	43.46	**1.38**	32
4	662.39	**12.59**	53
5	2640.23	**45.88**	58
6	3091.88	**38.30**	81
7	208.03	**5.41**	38
8	1007.54	**14.28**	71
9	137.75	**3.63**	38
10	505.07	**8.36**	60
11	487.16	**11.87**	41
12	50.70	**3.34**	15
13	125.08	**2.90**	43
14	2039.23	**38.40**	53
Trung bình			**48**

VI. KẾT LUẬN

Sự đóng góp của một độ đo khoảng cách vào hiệu quả của phân lớp chuỗi thời gian tùy thuộc vào đặc điểm của dữ liệu trong một miền ứng dụng cụ thể. Ngoài ra, tính đa dạng của các độ đo khoảng cách trong một phương pháp tổ hợp có thể góp phần làm cho phuơng pháp này trở nên tổng quát hơn và cải thiện được hiệu quả phân lớp hơn. Được gợi cảm hứng từ hai ý tưởng trên, trong nghiên cứu này, chúng tôi đề xuất một phương pháp tổ hợp cho bài toán phân lớp chuỗi thời gian mà kết hợp các bộ phân lớp 1-lân cận gần nhất với các độ đo khoảng cách đàn hồi lẫn độ đo khoảng cách không đàn hồi. Chúng tôi còn song song hóa giải pháp và hiện thực dựa vào GPU cho phương pháp tổ hợp đề xuất.

Kết quả thực nghiệm trên 14 bộ dữ liệu mẫu cho thấy phương pháp tổ hợp đề xuất vượt trội hơn phương pháp 1NN+DTW mà được xem là phương pháp "khó đánh bại" và đồng thời đem lại độ chính xác phân lớp ngang bằng với phương pháp tổ hợp EE của công trình [14] mà sử dụng 8 độ đo khoảng cách đàn hồi. Nhận định này hàm ý rằng một phương pháp tổ hợp sử dụng cả độ đo đàn hồi và độ đo không đàn hồi có thể đem lại một hiệu quả

phân lớp ngang bằng với một phương pháp tổ hợp dùng toàn độ đo đàn hồi trong khi các độ đo không đàn hồi thường đem lại chi phí tính toán thấp hơn các độ đo đàn hồi và không đòi hỏi những nỗ lực để ước lượng các tham số liên quan. Ngoài ra, chúng tôi so sánh tính hữu hiệu của cách hiện thực dựa vào GPU cho phương pháp đề xuất với cách hiện thực tuần tự cho phương pháp này. Kết quả của thực nghiệm cho thấy trung bình cách hiện thực giải pháp đề xuất bằng GPU thực thi nhanh gấp 48 lần so với cách hiện thực tuần tự cho giải pháp đề xuất.

Trong tương lai, chúng tôi dự định sẽ đánh giá bằng thực nghiệm phương pháp tổ hợp đề xuất ENE2 trên nhiều bộ dữ liệu mẫu hơn nữa và so sánh với một số phương pháp phân lớp chuỗi thời gian khác. Ngoài ra, chúng tôi dự định (i) áp dụng một phương pháp hữu hiện để ước lượng kich thước cửa sổ xoắn cho các độ đo đàn hồi như trong các công trình ([15], [18]) và (ii) áp dụng hai kỹ thuật để tối ưu các tham số trong từng bộ phân lớp 1-NN thành phần kết hợp với một độ đo khoảng cách như trong công trình [24] để cải thiện hơn nữa tính hữu hiệu về thời gian của giải pháp tổ hợp đề xuất cho bài toán phân lớp chuỗi thời gian.

VII. TÀI LIỆU THAM KHẢO

[1] A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, **31:** 606-660, 2017.

[2] M. Middlehurst, P. Schafer, A. Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *arXiv* 2304.13029, 2023.

[3] A. Abanda, U. Mori, and J. A. Lozano. A review on distance based time series classification. *Data Min. Knowl. Discov.*, **33**: 2, 378–412, 2019.

[4] J. Serra, J. L. Arcos. An empirical evaluation of similary measures for time series classification. *Knowledge-based Systems*, **67**:305-314, 2014.

[5] E. J. Keogh and M. J. Pazzani. Derivative Dynamic Time Warping. *Proceedings of 6th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1–11, 2001.

[6] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, **7:** 3: 358–386, 2005.

[7] Y. Jeong, M. Jeong, O. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, **44**: 2231-2240, 2011.

[8] M. Vlachos, G. Killios, D. Gunopulos. Discovering Similar Multidimensional Trajectories. *Proceedings of 18th Int. Conf. on Data Engineering*, San Jose, CA, USA, 26 Feb – 01 March, 2002.

[9] L. Chen, R. Ng, On the marriage of p-norms and edit distance. *Proc. of the 30th International Conference on Very Large Databases*, vol. 30, pp. 792-803, 2004.

[10] P.F. Marteau. Time warping edit distance with stiffness adjustment for time series matching. *IEEE Trans. Pattern Anal Mach Intll*, **311**: 2: 306-318, 2009.

[11] A. Stefan, V. Athitsos, G. Das. The move-split-merge metric for time series. *IEEE Trans Knowl Data Eng*, **25**:6: 1425 -1438, 2012.

[12] G. E. A. P. A. Batista, E. J. Keogh, O. M. Tataw, and V. M. A. De Souza. CID: An efficient complexity-invariant distance for time series. *Data Min. Knowl. Discov.*, **28**: 3: 634–669, 2014.

[13] V. T. Vinh, D. T. Anh. Compression rate distance measure for time series. *Proc. of 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Paris, France, October 19-21, 2015.

[14] J. Lines and A. Bagnall. Time series classification with ensembles of elastic distance measures. *Data Min. Knowl. Discov.*, **29**: 3: 565–592, 2015.

[15] C. W. Tan, F. Petijean, G. I. Webb. FastEE: Fast Ensembles of Elastic Distances for Time Series Classification. *Data Mining and Knowledge Discovery,* **34**: 231-272, 2020.

[16] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, **26**: 275–309, 2013.

[17] C. Ratanamahatana, E. Keogh. Three myths about dynamic time warping data mining. *Proc. of the 2005 SIAM International Conference on Data Mining* (SDM), pp. 506-510, 2005.

[18] C. W. Tan, M. Herrmann, G. Forestier, G. I. Web, F. Petitjean. Efficient search of the best warping window for dynamic time warping. *Proc. of 2018 SIAM International Conference on Data Mining* (SDM), pp. 225-233, 2018.

[19] S. Cook. *CUDA programming – A developer's guide to parallel computing with GPUs*, Morgan Kaufmenn, 2013.

[20] NVIDIA. 2017. *CUDA Programming Guide Version 8.0*, https://docs.nvidia.com/cuda/index.html

[21] NVIDIA. 2017. *CUDA Toolkit Documentation Version 8.0*, https://docs.nvidia.com/cuda/index.html

[22] M. E. Belviranli, P. Deng, L. N. Bhuyan, R. Gupta, and Q. Zhu. PeerWave: Exploiting Wavefront Parallelism on GPUs with Peer-SM Synchronization. *Proceedings of the 29th ACM on International Conference on Supercomputing - ICS '15*, New York, USA, June, pp. 25–35, 2015.

[23] H. A. Dau, E. Keogh, K. Kamgar, C. M. Yeh, Y. Zhu, S. Gharghabi , C. A. Ratanamahatana, Y. Chen, B. Hu, N. Begum, A. Bagnall , A. Mueen, G. Batista, & Hexagon-ML, The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/ (Accessed in 2020).

[1] G. Oaster, J. Lines. A significantly faster elastic ensemble for time series classificaton, In: Yin, H., Camacho, D., Tino, P., Tallón-Ballesteros, A., Menezes, R., Allmendinger, R. (eds) *Intelligent Data Engineering and Automated Learning* (IDEAL 2019), LNCS 11871, Springer, Cham., pp. 446-453, 2019.

GPU-BASED TIME SERIES CLASSIFICATION THROUGH ENSEMBLES OF NON-ELASTIC AND ELASTIC DISTANCES

Duong Tuan Anh, Vo Dai Duong, Pham Minh Tri

ABSTRACT— Improving the effectiveness and efficiency of time series classification is a very important task. This paper presents a new ensemble approach for time series classification in which each 1-NN (one-nearest-neighbor) classifier is coupled with a non-elastic or elastic distance measure. We also design a GPU-based parallel implementation for the proposed ensemble method to improve its efficiency. Experimental results over a collection of benchmark datasets showed that our proposed method remarkably outperforms the 1NN with the Dynamic Time Warping measure which has been considered in the literature as "difficult to beat" and brings out the same classification accuracy as the ensemble method which uses eight elastic distance measures (the Elastic Ensemble method). Besides, we compare the efficiency of GPU-based implementation to that of sequential implementation for the proposed method. Results in the latter experiment showed that on average the GPU version can run faster than the sequential version by about 48 times.

Keywords— time serie, classifition; ensemble, k-nearest-neighbors, elastic distance, non-elastic distance, GPU.

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. Từ năm 2020 đến nay, ông là giảng viên khoa Công Nghệ Thông Tin, trường Đại Học Ngoại ngữ-Tin học Tp. Hồ Chí Minh. Lãnh 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 120 bài báo khoa học.

Phạm Minh Trí tốt nghiệp cử nhân công nghệ điện tử tại Trường Đại Học Công Nghiệp Tp. Hồ Chí Minh năm 2012 và tốt nghiệp thạc sĩ ngành Khoa Học Máy Tính tại Trường Đại Học Bách Khoa Tp. Hồ Chí Minh năm 2020. Anh là nghiên cứu viên công tác tại Viện Khoa Học và Công Nghệ Tính Toán, Tp. Hồ Chí Minh đến 2010 đến nay. Lãnh vực nghiên cứu chính của anh là khai phá dữ liệu chuỗi thời gian.

Võ Đại Dương tốt nghiệp cử nhân công nghệ thông tin tại Trường Đại Học Giao Thông Vận Tải Tp. Hồ Chí Minh năm 2013 và tốt nghiệp thạc sĩ ngành Khoa Học Máy Tính tại Trường Đại Học Bách Khoa Tp. Hồ Chí Minh năm 2022. Anh là lập trình viên công tác tại Công Ty Gento Tech từ 2012 đến 2014 và là chuyên viên kỹ thuật tại Công Ty Merkle Viet Nam từ 2015 đến nay. Lãnh vực nghiên cứu chính của anh là khai phá dữ liệu chuỗi thời gian.