BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH

CÔNG TRÌNH NGHIÊN CỨU KHOA HỌC CẤP TRƯỜNG

DỰ BÁO DỮ LIỆU CHUỖI THỜI GIAN CÓ TÍNH XU HƯỚNG HOẶC MÙA SỬ DỤNG GIẢI THUẬT *K* LÂN CẬN GẦN NHẤT

MÃ SỐ: T2015-79TĐ

Tp. Hồ Chí Minh, 2015

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO TỔNG KẾT
ĐỀ TÀI KH & CN CẤP TRƯỜNG TRỌNG ĐIỂM

DỰ BÁO DỮ LIỆU CHUỖI THỜI GIAN CÓ TÍNH XU HƯỚNG HOẶC MÙA SỬ DỤNG GIẢI THUẬT *K* LÂN CẬN GẦN NHẤT

Mã số: T2015-79TĐ

Chủ nhiệm đề tài: Nguyễn Thành Sơn

TP. HCM, 11/2015

MỤC LỤC

DANH MỤC CÁC HÌNH ẢNH

DANH MỤC CÁC TỪ VIẾT TẮT

ANN	Artificial Neuron Network
ARIMA model	Autoregressive Integrated Moving Average model
APCA	Adaptive Piecewise Constant Approximation
DTW	Dynamic Time Warping
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
ESAX	Extended Symbolic Aggregate approximation
ECG	Electrocardiogram
iSAX	indexable SAX
k-NN	*k*-Nearest Neighbors
MBR	Minimum Bounding Rectangle
MP_C	Middle Points_Clipping
MLP	Multi-layer perceptrons
MER	Mean error relative to xmean
MAE	Mean absolute error
PAA	Piecewise Aggregate Approximation
PIP	Perceptually Important Point
PSF	Pattern sequence-based forecasting
SAX	Symbolic Aggregate approximation
SBR	Skyline Bounding Region

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH
KHOA CNTT

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Tp. HCM, Ngày 20 tháng 10 năm 2015

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: Dự báo dữ liệu chuỗi thời gian có tính xu hướng hoặc mùa sử dụng giải thuật *k* lân cận gần nhất.
- Mã số: T2015-79TĐ
- Chủ nhiệm: Nguyễn Thành Sơn
- Cơ quan chủ trì: Trường Đại học SPKT Tp. HCM
- Thời gian thực hiện: 6/2014- 10/2015

2. Mục tiêu:

Ứng dụng phương pháp so trùng mẫu trong dự báo dữ liệu chuỗi thời gian có tính xu hướng hoặc mùa.

3. Tính mới và sáng tạo:

Nhiều dữ liệu chuỗi thời gian trong kinh doanh, kinh tế và các lãnh vực đời sống thường biểu hiện tính mùa hoặc tính xu hướng. Mặc dù yếu tố mùa là một thành phần quan trọng nhất trong chuỗi thời gian có tính mùa, xu hướng thường đi kèm với biến động mùa và có thể có ảnh hưởng lớn đến các phương pháp dự báo. Dự báo chính xác dữ liệu chuỗi thời gian có tính xu hướng và tính mùa là rất quan trọng để hỗ trợ ra quyết định trong các lãnh vực của đời sống. Đề tài đề xuất một phương pháp mới đơn giản và hiệu quả cho bài toán dự báo trên chuỗi thời gian có tính xu hướng hoặc theo mùa.

4. Kết quả nghiên cứu:

Đề xuất được một phương pháp mới cho bài toán dự báo trên chuỗi thời gian có tính xu hướng hoặc mùa sử dụng thuật toán *k* lân cận gần nhất.

5. Sản phẩm:

Một bài báo đăng trên tạp chí Khoa học Giáo dục Kỹ thuật, báo cáo và chương trình demo.

6. Hiệu quả, phương thức chuyển giao kết quả nghiên cứu và khả năng áp dụng:

Có thể áp dụng trong giảng dạy sau đại học về chuyên đề chuỗi thời gian, sử dụng làm cơ sở cho việc phát triển các ứng dụng trong các lĩnh vực liên quan khác.

Trưởng Đơn vị	**Chủ nhiệm đề tài**
(ký, họ và tên)	*(ký, họ và tên)*

INFORMATION ON RESEARCH RESULTS

1. General information:

Project title: Prediction in seasonal or trend time series using k nearest neighbors.
Code number: T2015-79TĐ.
Coordinator: Nguyen Thanh Son
Implementing institution: HCM City University of Technical Education.
Duration: from 6/2014 to 11/2015

2. Objective(s):

Investigate the use of pattern matching in seasonal or trend time series prediction

3. Creativeness and innovativeness:

Time series data in many applications of various life areas usually have seasonal or trend property. Although the seasonal factor is the most important element in seasonal time series data, the trend factor usually accompanies with seasonal fluctuation and can impact on predictive methods. The accuracy of seasonal or trend time series forecasting is fundamental to many decision processes. We proposed a new method which is simple and effective for forecasting seasonal or trend time series data.

4. Research results:

A new method proposed for forecasting seasonal or trend time series data.

5. Products:

A paper published in Journal of Technical Education Science, a technical report and a demo.

6. Effects, transfer alternatives of reserach results and applicability:

It can be used to lecture for the major course of time series at postgraduate level or as a base for developing application softwares in some other relevant areas

PHẦN MỞ ĐẦU

1. Tình hình nghiên cứu trong và ngoài nước.

Dự báo trên dữ liệu chuỗi thời gian đã và đang là một công việc phức tạp và thách thức đối với các nhà nghiên cứu. Tuy có một số phương pháp thường được sử dụng trên dữ liệu chuỗi thời gian như phương pháp làm trơn theo hàm mũ, mô hình ARIMA, mạng nơ ron nhân tạo. Nhưng hai phương pháp đầu chỉ có thể nắm bắt được các đặc trưng tuyến tính của chuỗi thời gian, còn việc mạng nơ ron nhân tạo có thể xử lý một cách hiệu quả dữ liệu có tính xu hướng và tính mùa hay không đang là một vấn đề gây bàn cãi vì có những nhận định trái ngược nhau trong cộng đồng nghiên cứu về dự báo dữ liệu chuỗi thời gian [49]. Mặt khác, gần đây một số phương pháp dự báo trên dữ liệu chuỗi thời gian dựa vào hướng tiếp cận so trùng mẫu đã được ứng dụng dự báo cho một số lĩnh vực cụ thể (như thời tiết, chứng khoán, giá điện và nhu cầu sử dụng điện) và là một hướng tiếp cận đáng quan tâm.

2. Tính cấp thiết của đề tài.

Dữ liệu chuỗi thời gian là loại dữ liệu được sử dụng phổ biến trong các lĩnh vực khoa học, công nghệ, y học và thương mại. Chẳng hạn, trong y khoa người ta có thể sử dụng các bài toán về chuỗi thời gian để xây dựng chương trình dò tìm tự động trên điện não đồ của bệnh nhân để phát hiện bệnh, hoặc trong lĩnh vực chứng khoán ta có thể ứng dụng các bài toán về chuỗi thời gian để xây dựng chương trình dự báo xu thế biến động của chứng khoán trong thời gian sắp tới, v.v… . Một nghiên cứu khảo sát từ 4000 hình được lấy ngẫu nhiên trong các báo tin tức trên thế giới được xuất bản trong giai đoạn từ 1974 đến 1989 cho thấy hơn 75% là các hình biểu diễn dữ liệu chuỗi thời gian [39].

Nhiều dữ liệu chuỗi thời gian trong kinh doanh, kinh tế và các lãnh vực đời sống thường biểu hiện tính mùa và tính xu hướng. Tính mùa là khuôn mẫu thường lặp lại và có tính chu kỳ do những yếu tố như thời tiết, lễ tết, những đợt khuyến mãi, v.v... . Mặc dù yếu tố mùa là một thành phần quan trọng nhất trong chuỗi thời gian có tính mùa, xu hướng thường đi kèm với biến động mùa và có thể có ảnh hưởng lớn đến các phương pháp dự báo. Một chuỗi thời gian có xu hướng được xem là một chuỗi thời gian không dừng (nonstationary) và thường phải làm cho trở thành chuỗi thời gian có tính dừng

(stationary) trước khi quá trình dự báo diễn ra. Dự báo chính xác dữ liệu chuỗi thời gian có tính xu hướng và tính mùa là rất quan trọng để hỗ trợ ra quyết định trong các lãnh vực của đời sống.

3. Ý nghĩa lý luận và thực tiễn.

3.1 Ý nghĩa lý luận.

Ứng dụng phương pháp so trùng mẫu trong dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa là một hướng tiếp cận mới cho bài toán đầy thách thức này. Một thể hiện của phương pháp so trùng mẫu là giải thuật *k-lân cận gần nhất* dùng cho dự báo chuỗi thời gian. Đề tài đề xuất sử dụng phương pháp thu giảm số chiều MP_C và cấu trúc chỉ mục đường chân trời vào giải thuật *k*-lân cận gần nhất cho công tác dự báo dữ liệu chuỗi thời gian, đặc biệt cho dữ liệu chuỗi thời gian có tính mùa và xu hướng. Kết quả thực nghiệm của cách tiếp cận *k*-lân cận gần nhất sẽ được so sánh với một mô hình thông dụng trong dự báo chuỗi thời gian là mạng nơ ron nhân tạo (ANN). Mô hình mạng nơ ron nhân tạo được dùng để so sánh vì cả hai mô hình nơ ron nhân tạo và mô hình *k*-lân cận gần nhất đều là những mô hình phi tuyến.

3.2 Ý nghĩa thực tiễn.

Nghiên cứu này sẽ là nền tảng cho những nghiên cứu tiếp theo về các bài toán khác trong khai phá dữ liệu chuỗi thời gian. Ngoài ra, còn có thể áp dụng giảng dạy như một chuyên đề cho sinh viên sau đại học.

4. Các đối tượng nghiên cứu.

Dữ liệu chuỗi thời gian và bài toán dự báo trên chuỗi thời gian.

5. Phạm vi và các phương pháp nghiên cứu.

5.1 Phạm vi nghiên cứu.

- Dự báo trên chuỗi thời gian có tính xu hướng hoặc mùa.

5.2 Các phương pháp nghiên cứu.

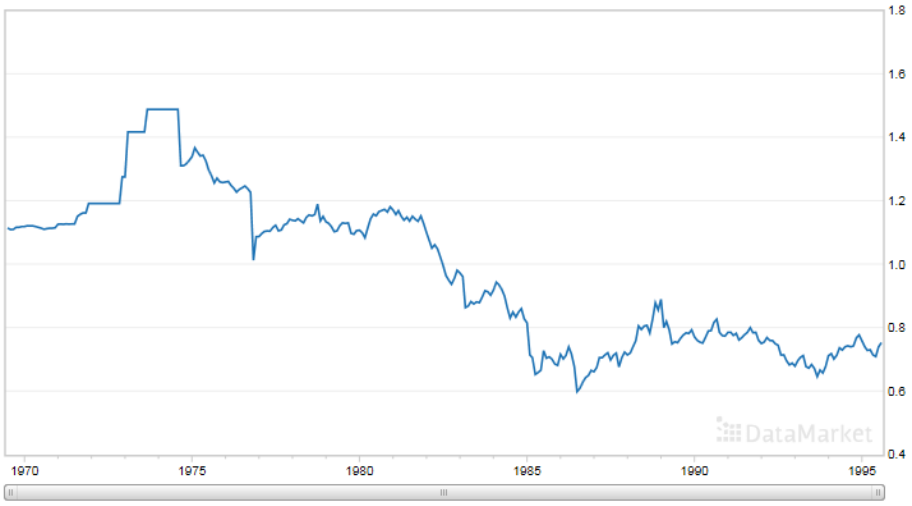
- Tổng kết các kết quả nghiên cứu liên quan trước đây. Đánh giá hiệu quả của các phương pháp. Thực nghiệm để kiểm tra kết quả.
- Nghiên cứu tài liệu, ứng dụng mô hình lý thuyết và chứng minh bằng thực nghiệm.

PHẦN NỘI DUNG

CHƯƠNG 1. Các kiến thức cơ sở.

1.1 Tổng quan về đề tài.

Một *chuỗi thời gian* (time series) là một chuỗi các điểm dữ liệu được đo theo từng khoảng thời gian liền nhau theo một tần suất thời gian thống nhất. Hình 1.1 minh họa một ví dụ về chuỗi thời gian biểu diễn tỉ giá chuyển đổi trung bình hàng tháng giữa đô la Úc và đô la Mỹ (đơn vị đô la Úc) từ 7/1969 đến 8/1995.



Hình 1.1 Đường biểu diễn một chuỗi thời gian ([17]).

Các bài toán thường được nghiên cứu trong khai phá dữ liệu chuỗi thời gian gồm *tìm kiếm tương tự* (similarity search), *gom cụm* (clustering), *phân lớp* (classification), *phát hiện motif* (motif discovery), *khai phá luật* (rule discovery), *phát hiện bất thường* (anomaly detection), *trực quan hóa* (visualization), *dự báo* (forecast).

Những khó khăn và thách thức khi nghiên cứu về dữ liệu chuỗi thời gian [25]:

- Dữ liệu thường rất lớn. Chẳng hạn, trong 1 giờ, dữ liệu điện tâm đồ (ECG) có thể lên đến 1GB.
- Phụ thuộc nhiều vào yếu tố chủ quan của người dùng và tập dữ liệu khi đánh giá mức độ tương tự giữa các chuỗi thời gian.
- Dữ liệu không đồng nhất: định dạng của dữ liệu khác nhau, tần số lấy mẫu khác nhau. Ngoài ra, dữ liệu có thể bị nhiễu, thiếu một vài giá trị hoặc không sạch.

Bài toán tìm kiếm tương tự (so trùng) trong cơ sở dữ liệu chuỗi thời gian đã được nhiều nhà nghiên cứu quan tâm trong những năm qua vì đây là *bài toán cơ bản* và là một thành phần nền tảng của nhiều bài toán khác trong khai phá dữ liệu chuỗi thời gian. Đây là bài toán khó vì kích thước dữ liệu chuỗi thời gian thường lớn và vì chúng ta không thể lập chỉ mục dữ liệu chuỗi thời gian một cách dễ dàng như trong hệ thống cơ sở dữ liệu truyền thống. Một vài thí dụ về ứng dụng của tìm kiếm tương tự trên chuỗi thời gian có thể nêu ra như sau:

- Tìm trong quá khứ, những giai đoạn mà số lượng sản phẩm bán được như tháng vừa rồi.
- Tìm những sản phẩm có chu kỳ doanh số giống nhau.
- Tìm những đoạn nhạc trong một bài hát giống một đoạn nhạc đã có bản quyền.
- Tìm những tháng trong quá khứ mà có lượng mưa giống như tháng vừa rồi.
- Tìm những năm khô hạn mà mực nước các sông đều ở mức thấp.

Dự báo trên dữ liệu chuỗi thời gian đã và đang là một công việc phức tạp và thách thức đối với các nhà nghiên cứu. Tuy có một số phương pháp thường được sử dụng trên dữ liệu chuỗi thời gian như phương pháp làm trơn theo hàm mũ, mô hình ARIMA, mạng nơ ron nhân tạo. Nhưng hai phương pháp đầu chỉ có thể nắm bắt được các đặc trưng tuyến tính của chuỗi thời gian, còn việc mạng nơ ron nhân tạo có thể xử lý một cách hiệu quả dữ liệu có tính xu hướng và tính mùa hay không đang là một vấn đề gây bàn cãi vì có những nhận định trái ngược nhau trong cộng đồng nghiên cứu về dự báo dữ liệu chuỗi thời gian [49]. Mặt khác, gần đây một số phương pháp dự báo trên dữ liệu chuỗi thời gian dựa vào hướng tiếp cận so trùng mẫu đã được ứng dụng dự báo cho một số lĩnh vực cụ thể (như thời tiết, chứng khoán, giá điện và nhu cầu sử dụng điện) và là một hướng tiếp cận đáng quan tâm.

1.2 Lý thuyết cơ sở và các công trình liên quan.

Trong phần này, chúng tôi giới thiệu tóm tắt cơ sở lý thuyết về các độ đo tương tự, các phương pháp thu giảm số chiều, các cấu trúc chỉ mục thường dùng và các công trình liên quan tới bài toán được nghiên cứu.

1.2.1 Các độ đo tương tự.

Trong các bài toán về chuỗi thời gian, để so sánh 2 chuỗi người ta sử dụng các độ đo tương tự. Hai đối tượng được xem là giống nhau khi độ đo tương tự giữa chúng bằng 0, được xem là tương tự nếu độ đo tương tự giữa chúng nhỏ hơn một giá trị ε được qui ước trước đó. Để có thể tính toán và so sánh, độ đo này được biểu diễn thành các số thực và phải thỏa các tính chất sau:

- $D(x,y) = 0$ nếu và chỉ nếu $x = y$
- $D(x, y) = D(y, x)$
- $D(x, y) \geq 0$ với mọi x, y
- $D(x, y) < D(x, z) + D(y, z)$

Dưới đây là các độ đo thường được sử dụng

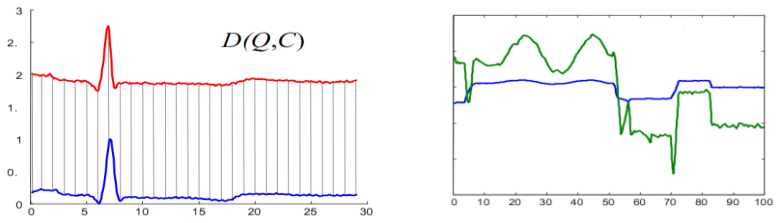
- ***Độ đo Minkowski.***

Ký hiệu là $Sim(X,Y)$ (độ tương tự giữa hai chuỗi X và Y có chiều dài n) và được định nghĩa như sau:

$$Sim(X,Y) = \sqrt[P]{\sum_{i=1}^{n}(x_i - y_i)^P} \text{ , với } x_i \in X, y_i \in Y, i = 1, \ldots, n \qquad (1.1)$$

Trong đó, $p = 2$ (Euclid) là độ đo thường được sử dụng.

Độ đo này có ưu điểm tính toán dễ dàng. Tuy nhiên nó cũng có một số nhược điểm là do phương pháp này tính toán dựa trên các cặp giá trị tương ứng trong hai chuỗi nên đối với các trường hợp tính chất của hai mẫu là giống nhau nhưng giá trị khác nhau (có đường căn bản khác nhau hay có biên độ dao động khác nhau) thì khoảng cách hai mẫu sẽ rất khác nhau. Hình 1.2 minh họa trường hợp này.



(a) (b)

Hình 1.2 Minh họa hai chuỗi thời gian giống nhau.
nhưng (a) đường cơ bản khác nhau và (b) biên độ giao động khác nhau ([26]).

Để khắc phục trường hợp này trước khi áp dụng các giải thuật ta cần thực hiện chuẩn hóa dữ liệu. Các phương pháp chuẩn hóa thường được dùng là:

- Chuẩn hóa trung bình zero (Zero-Mean normalization) [18]

Chuỗi Q được biến đổi thành chuỗi Q' theo công thức

$$Q'[i] = (Q[i] - mean(Q)) / var(Q) \quad (1.2)$$

Với $mean(Q)$ là giá trị trung bình của Q và $var(Q)$ là độ lệch chuẩn của Q.

- Chuẩn hóa nhỏ nhất-lớn nhất (Min-Max normalization) [18]

Chuỗi Q được biến đổi thành chuỗi Q' theo công thức

$$Q'[i] = \frac{Q[i] - Min_{old}}{Max_{old} - Min_{old}}(Max_{new} - Min_{new}) + Min_{new} \quad (1.3)$$

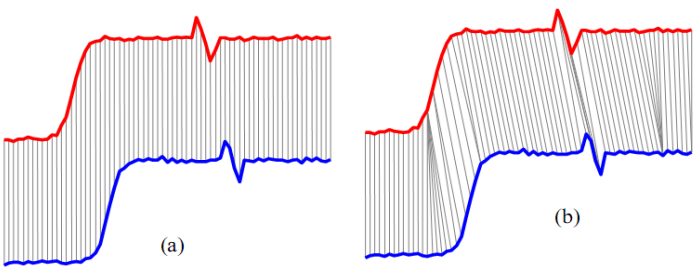
Với Min_{old} và Max_{old} là giá trị nhỏ nhất và lớn nhất của chuỗi ban đầu.

Min_{new} và Max_{new} là giá trị nhỏ nhất và lớn nhất của chuỗi sau khi được chuẩn hóa.

- ***Độ đo xoắn thời gian động.***

Trong trường hợp hai mẫu cần so sánh có hai đường biểu diễn không hoàn toàn giống nhau nhưng hình dạng biến đổi rất giống nhau thì khi so sánh độ tương tự giữa hai mẫu bằng cách so sánh từng cặp điểm 1-1 (so điểm thứ i của đường thứ nhất và điểm thứ i của đường thứ hai) là không phù hợp. Hình 1.3 minh họa hai đường biểu diễn rất giống nhau về hình dạng nhưng lệch nhau về thời gian.

Trong trường hợp này, nếu tính khoảng cách bằng cách ánh xạ 1-1 giữa hai đường thì kết quả rất khác nhau và có thể dẫn đến kết quả cuối cùng không giống như mong muốn. Vì vậy để khắc phục nhược điểm này, một điểm có thể ánh xạ với nhiều điểm và ánh xạ này không thẳng hàng. Phương pháp này gọi là *xoắn thời gian động* (Dynamic Time Warping - DTW) [5].



Hình 1.3 Khoảng cách giữa hai đường biểu diễn rất giống nhau về hình dạng nhưng lệch nhau về thời gian.
(a) tính theo độ đo Euclid và (b) tính theo độ đo DTW ([26]).

Cách tính DTW

Cách đơn giản nhất để tính DTW của hai đường X và Y là ta xây dựng ma trận $D_{m\,x\,n}$ với $m = |X|$ và $n = |Y|$. Khi đó, $D_{ij} = d(x_i , y_j)$.

Sau khi xây dựng ma trận D , ta tìm đường đi từ ô *(0,0)* đến ô *(m,n)* thỏa mãn những ràng buộc sau:

- Không được đi qua trái hay đi xuống
- Đường đi phải liên tục
- Ô *(i,j)* thuộc đường đi phải thỏa $|i - j| <= w$

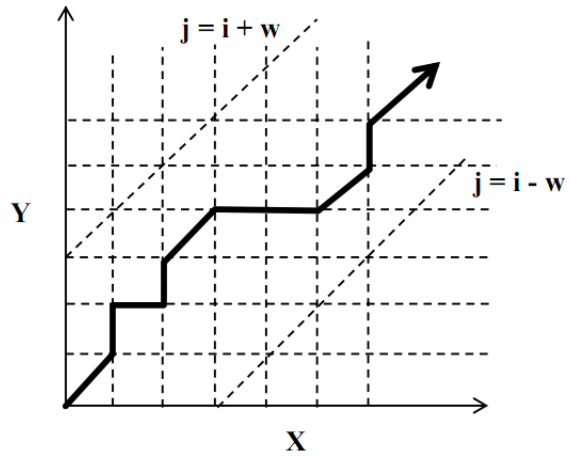
Giả sử có K ô đi từ ô (0,0) đến ô (m,n) thỏa mãn những điều kiện trên,

khi đó: $$DTW(Q,C) = \min\left\{ \sqrt{\sum_{k=1}^{K} w_k} \Big/ K \right.$$

Tuy nhiên, ta có thể dùng quy hoạch động để giải quyết bài toán này. Trong đó, công thức truy hồi để tính *D(i, j)*:

$$D(i,j) = |x_i - y_j| + \min\{ D(i-1, j), D(i-1, j-1), D(i, j-1) \}$$

Độ đo tương tự DTW có ưu điểm là cho kết quả chính xác hơn so với độ đo Euclid và cho phép nhận dạng mẫu có hình dạng giống nhau nhưng chiều dài hình dạng về thời gian có thể khác nhau. Độ đo tương tự này có nhược điểm là thời gian chạy lâu, tuy nhiên gần đây đã có những công trình tăng tốc độ tìm kiếm tương tự dùng độ đo DTW, tiêu biểu nhất là công trình của Keogh và các cộng sự, năm 2002 [27].
Hình 1.4 minh họa cách tính khoảng cách theo DTW.



Hình 1.4 Minh họa cách tính khoảng cách theo DTW.

1.2.2 Thu giảm số chiều chuỗi thời gian.

Thu giảm số chiều là phương pháp biểu diễn chuỗi thời gian n chiều $X = \{x_1, x_2, ..., x_n\}$ thành chuỗi thời gian có N chiều $Y = \{y_1, y_2, ..., y_N\}$ với $N << n$, nhưng vẫn phải giữ được các đặc trưng của chuỗi thời gian ban đầu. Với N càng lớn thì sự khôi phục càng chính xác.

Dữ liệu chuỗi thời gian thường rất lớn nên việc tìm kiếm trực tiếp trên dữ liệu chuỗi thời gian gốc sẽ không hiệu quả. Để khắc phục vấn đề này, cách tiếp cận chung thường được sử dụng bao gồm các bước sau:

1. Áp dụng một số phương pháp biến đổi xấp xỉ để thu giảm độ lớn của dữ liệu sao cho vẫn giữ được các đặc trưng của dữ liệu. Các phương pháp biến đổi xấp xỉ này thường được gọi là những phương pháp *thu giảm số chiều* (dimensionality reduction).
2. Thực hiện bài toán trên dữ liệu xấp xỉ, ta thu được tập kết quả xấp xỉ.
3. Dựa trên tập kết quả xấp xỉ này, thực hiện truy cập đĩa để thực hiện hậu kiểm trên dữ liệu gốc nhằm loại bỏ các chuỗi tìm sai trong tập kết quả xấp xỉ.

- ***Điều kiện chặn dưới.***

Do khi xấp xỉ dữ liệu sẽ gây ra mất mát thông tin, nên khi thực hiện trên dữ liệu xấp xỉ có thể xảy ra *lỗi tìm sót* (false dismissal) và/hoặc *tìm sai* (false alarm). Để đảm bảo có kết quả chính xác, lỗi tìm sót không được phép xảy ra. Mặt khác, lỗi tìm sai cũng nên thấp để giảm chi phí trong quá trình hậu kiểm.

Một kết quả quan trọng đã được Faloutsos và các cộng sự chứng minh là để không xảy ra lỗi tìm sót thì độ đo khoảng cách sử dụng trong không gian xấp xỉ (đặc trưng) phải là chặn dưới của độ đo khoảng cách sử dụng trong không gian gốc [11]. Nghĩa là, $d_{feature}(X', Y') \leq d(X, Y)$ với $d_{feature}(X', Y')$ là độ đo khoảng cách giữa hai chuỗi xấp xỉ của hai chuỗi ban đầu X, Y và $d(X, Y)$ là độ đo khoảng cách giữa hai chuỗi X, Y. Điều kiện này được gọi là *bổ đề chặn dưới* (lower bounding lemma).

1.2.3 Các phương pháp thu giảm số chiều dựa vào rút trích đặc trưng.

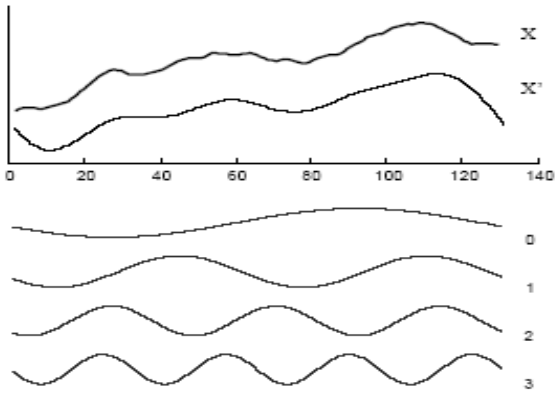
Có nhiều phương pháp thu giảm số chiều đã được đề xuất. Dưới đây chúng tôi sẽ trình bày một số phương pháp tiêu biểu.

- **Phương pháp biến đổi Fourier rời rạc.**

Kỹ thuật thu giảm số chiều áp dụng phương pháp DFT do Agrawal và các cộng sự đề xuất đầu tiên năm 1993 [1]. Ý tưởng cơ bản của phương pháp này là để thu giảm số chiều một chuỗi thời gian *X* có chiều dài *n* vào không gian đặc trưng *N* chiều (*N* << *n*), chuỗi thời gian ban đầu được biến đổi thành tập các hệ số (gọi là hệ số Fourier), các hệ số này có dạng sóng hình sin (và/hoặc cosin) và được tính theo công thức sau:

$$C_k = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t e^{-j2\pi kt} \tag{1.4}$$

Trong đó, C_k là số phức với $k = 0, \ldots, n\text{-}1$, x_t là giá trị thứ t của chuỗi thời gian, $t = 0, \ldots, n\text{-}1$ và $j = \sqrt{-1}$



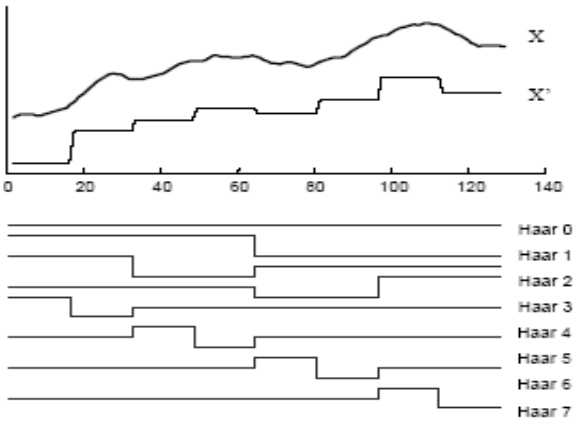
Hình 1.5 Minh họa phương pháp DFT ([28]).

Sau đó tổ hợp tuyến tính các sóng này ta có được dạng biểu diễn mong muốn (Hình 1.5). Một chuỗi thời gian được biến đổi theo cách này gọi là *biến đổi vào miền tần số*. Độ phức tạp của phép biến đổi *Fourier nhanh* (Fast Fourier Transform-FFT) là *O*(*n*log*n*) với *n* là số lượng điểm và phương pháp này thích hợp với các loại đường biểu diễn dữ liệu khác nhau, tuy nhiên chúng cũng có nhược điểm là khó giải quyết khi các chuỗi có chiều dài khác nhau.

- **Phương pháp biến đổi Wavelet rời rạc.**

Phương pháp DWT do Chan và Fu đề xuất năm 1999 [7]. Phương pháp này giống như DFT, tuy nhiên trong khi hàm cơ sở của phương pháp DFT có dạng hình sin và các hệ số Fourier luôn biểu diễn sự phân bố toàn cục của dữ liệu, thì hàm cơ sở thường được dùng trong phương pháp DWT là hàm Haar như trong Hình 1.6 và các hệ số Wavelet là những đoạn con cục bộ theo thời gian của dữ liệu được nghiên cứu.

Ngoài sử dụng hàm Haar, phương pháp DWT có thể sử dụng các hàm cơ sở khác như *Daubechies, Coiflet, Symmlet, …* . Tuy nhiên, Haar Wavalet đã được sử dụng rất nhiều trong khai phá dữ liệu chuỗi thời gian [40].



Hình 1.6 Minh họa phương pháp Haar Wavelet ([28]).

Phương pháp DWT rất hiệu quả vì nó mã hóa đơn giản và nhanh. Phương pháp này cũng thích hợp với những dữ liệu tĩnh ít thay đổi do đường *Haar* không thay đổi liên tục. Độ phức tạp của phép biến đổi DWT là $O(n)$, với n là chiều dài của chuỗi thời gian. Nhược điểm của phương pháp này là chiều dài chuỗi dữ liệu ban đầu phải là một số lũy thừa 2.

- **Phương pháp xấp xỉ gộp từng đoạn.**

Phương pháp xấp xỉ gộp từng đoạn (PAA) do Keogh và cộng sự đề xuất năm 2000 [28]. Theo phương pháp này, chuỗi thời gian ban đầu được chia thành N đoạn con có kích thước bằng nhau, sau đó tính trung bình của các điểm dữ liệu nằm trong mỗi đoạn con. Như vậy, chuỗi thời gian được xấp xỉ bằng N giá trị trung bình đó. Kết quả cuối cùng là đường thẳng có dạng bậc thang.

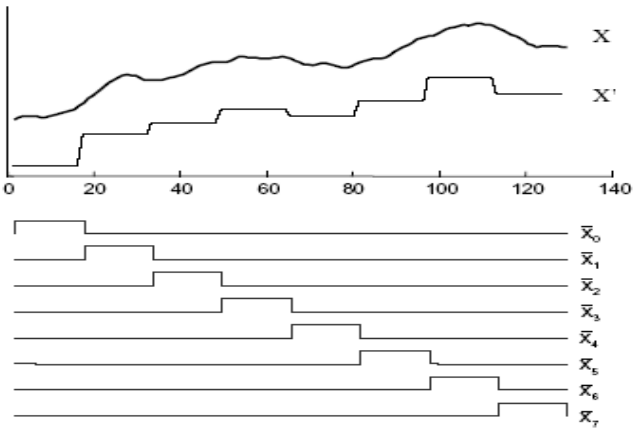
Cho chuỗi dữ liệu thời gian $X = (x_1, x_2, \ldots, x_n)$, phương pháp PAA sẽ biến đổi chuỗi này thành chuỗi $\overline{X} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_N)$ với $(N << n)$ theo công thức sau:

$$\overline{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j \qquad (1.5)$$

Ưu điểm của phương pháp này là đơn giản, thời gian tính toán rất nhanh và cách biểu diễn của nó hỗ trợ nhiều phương pháp tính khoảng cách (*Euclid, DTW*). Nhưng nhược điểm của nó là phương pháp có thể bỏ qua những điểm đặc biệt trong từng đoạn

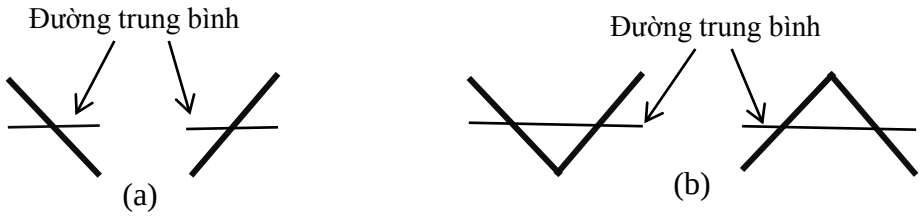
xấp xỉ của chuỗi thời gian. Vì vậy, trong nhiều trường hợp các đoạn có giá trị trung bình bằng nhau nhưng về khoảng cách Euclid rất khác nhau.

Hình 1.7 minh họa phương pháp này.



Hình 1.7 Minh họa phương pháp PAA ([28]).

Hình 1.8 là hai ví dụ minh họa cho các trường hợp này. Nhược điểm này làm cho PAA không thích hợp với một số dữ liệu chuỗi thời gian trong lĩnh vực tài chính [33]. Ngoài ra, chặn dưới của phương pháp PAA cũng chưa thật sự chặt.



Hình 1.8 Các trường hợp hai đoạn có cùng giá trị trung bình nhưng khoảng cách Euclid khác nhau.

Năm 2001, Keogh và các cộng sự đưa ra một cách tiếp cận tổng quát hơn so với PAA. Phương pháp này được gọi là *xấp xỉ hằng số từng đoạn thích nghi* (APCA – Adaptive Piecewise Constant Approximation) [29], nó cho phép các đoạn con có chiều dài khác nhau nhằm xấp xỉ tốt hơn chuỗi thời gian.

- **Phương pháp điểm cực trị.**

Năm 2003, Fink and Pratt đã đề xuất một kỹ thuật thu giảm số chiều dựa trên việc trích các điểm quan trọng trong chuỗi thời gian [12]. Các điểm quan trọng được lấy là các điểm cực đại và cực tiểu quan trọng và bỏ qua các điểm biến đổi nhỏ. Tỉ số nén được kiểm soát bằng tham số $R > 1$. Khi tăng R sẽ có ít điểm được lấy hơn. Các điểm cực trị quan trọng được định nghĩa như sau:

Điểm a_m trong chuỗi $a_1, \ldots, a_n$ được gọi là một *cực tiểu quan trọng* nếu có một cặp chỉ số i, j sao *cho* $i \le m \le j$, mà: a_m là cực tiểu trong đoạn $a_i \ldots a_j$ và $a_i / a_m \ge R$ và $a_j / a_m \ge R$.

Tương tự, điểm a_m trong chuỗi $a_1, \ldots, a_n$ được gọi là một *cực đại quan trọng* nếu có một cặp chỉ số i, j sao *cho* $i \le m \le j$, mà: a_m là cực đại trong đoạn $a_i \ldots a_j$ và $a_m / a_i \ge R$ và $a_m / a_j \ge R$.

Fink và Gandhi [13] đã đề xuất giải thuật trích ra những điểm cực trị quan trọng, giải thuật này có độ phức tạp $O(n)$. Nó quét qua chuỗi thời gian một lần và không cần qua giai đoạn tiền xử lý.

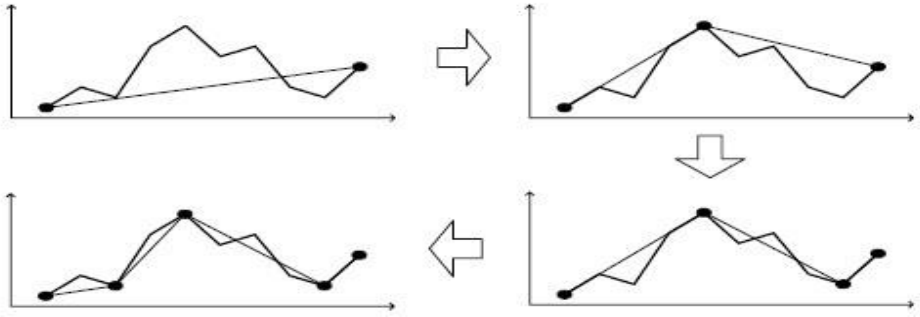
- **Phương pháp PIP.**

Năm 2001, Chung và các cộng sự đưa ra kỹ thuật thu giảm số chiều dựa vào các điểm PIP (Perceptually Important Points) [8]. Giải thuật xác định các điểm PIP như sau:

Với một chuỗi thời gian T đã được chuẩn hóa, hai điểm PIP đầu tiên được chọn là điểm đầu tiên và điểm cuối cùng của chuỗi T. Điểm PIP thứ ba được chọn là điểm trong T có khoảng cách lớn nhất so với hai điểm PIP đầu tiên. Điểm PIP thứ tư được chọn là điểm trong T có khoảng cách lớn nhất so với hai điểm PIP kế cận đã chọn (có thể là điểm đầu và điểm thứ ba hoặc điểm thứ ba và điểm cuối). Tiến trình xác định các điểm PIP tiếp tục cho đến khi số điểm PIP đạt được số điểm yêu cầu. Khoảng cách giữa một điểm trong T với hai điểm PIP kế cận đã chọn là *khoảng cách thẳng đứng* (Vertical Distance) từ điểm cần tính tới đường nối hai điểm PIP kế cận đã chọn.

Những ưu điểm của phương pháp thu giảm số chiều dựa vào điểm quan trọng là (1) phù hợp với trực giác, (2) các chuỗi thời gian có chiều dài khác nhau có thể so trùng và (3) có thể thu giảm số chiều ở nhiều mức phân giải khác nhau. Thông qua thực nghiệm các tác giả cho thấy rằng cách tiếp cận dựa vào các điểm quan trọng là hiệu quả. Tuy nhiên, họ chưa chứng minh về mặt lý thuyết tính chính xác của phương pháp này, tức là thỏa được điều kiện chặn dưới. Ngoài ra, các phương pháp thu giảm số chiều dựa vào điểm quan trọng còn có một nhược điểm khác là không đề xuất được cấu trúc chỉ mục đa chiều nào hỗ trợ.

Hình 1.9 minh họa quá trình nhận dạng các điểm PIP trên một chuỗi thời gian.



Hình 1.9 Minh họa quá trình nhận dạng các điểm PIP ([8]).

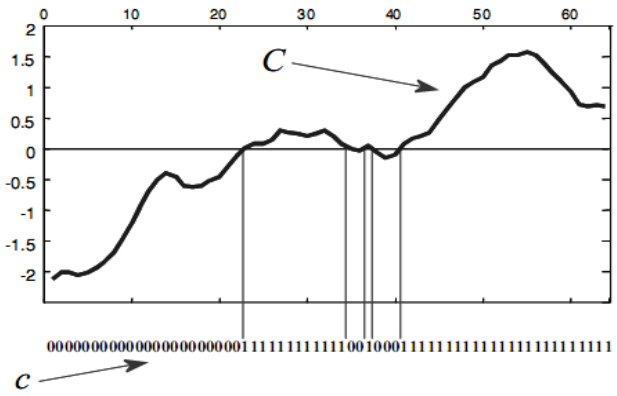
- **Phương pháp xén dữ liệu.**

Phương pháp *xén dữ liệu* (Clipping) do Ratanamahatana và các cộng sự đề xuất năm 2005 [42]. Xén dữ liệu là một tiến trình biến đổi các giá trị số thực của một chuỗi thời gian $C = (c_1, \ldots, c_n)$ thành một chuỗi bit b tùy thuộc giá trị đó nằm trên hay dưới đường trung bình của chuỗi. Quá trình biến đổi được thực hiện theo công thức sau:

$$b_t = \begin{cases} 1 & \text{nếu } c_t > \mu \\ 0 & \text{ngược lại} \end{cases} \qquad (1.6)$$

trong đó, μ là giá trị trung bình của chuỗi. Không mất tính tổng quát, tác giả giả định rằng $\mu = 0$. Hình 1.10 minh họa kỹ thuật xén dữ liệu một chuỗi thời gian.

Ưu điểm của kỹ thuật xén dữ liệu là (1) giữ được đặc trưng về hình dạng xấp xỉ của chuỗi thời gian, (2) có tỉ số nén cao tối thiểu là 32:1, (3) cho phép so sánh trực tiếp giữa chuỗi truy vấn gốc và biểu diễn xấp xỉ đồng thời vẫn thỏa điều kiện chặn dưới, (4) có thể sử dụng các phép toán chuyên dụng trên chuỗi bit. Tuy nhiên, kỹ thuật này có một số nhược điểm là (1) không hỗ trợ người dùng tùy chọn tỉ lệ thu giảm số chiều, (2) không có cấu trúc chỉ mục đa chiều hỗ trợ cho bài toán tìm kiếm tương tự trong cơ sở dữ liệu chuỗi thời gian lớn.



Hình 1.10 Minh họa kỹ thuật xén dữ liệu một chuỗi thời gian có chiều dài 64 ([42]).

- **Phương pháp MP_C (Middle points_clipping).**

Phương pháp MP_C do Sơn và Anh đề xuất năm 2011 [44]. Phương pháp này dựa trên việc chia chuỗi có chiều dài n thành N đoạn (segment) với $N << n$. Một số điểm trong mỗi đoạn sẽ được chọn. Việc chọn các điểm này nhằm mục đích tăng độ chặt chặn dưới của phương pháp đề xuất so với phương pháp thông dụng PAA đồng thời có thể lưu trữ hình dạng xấp xỉ của chuỗi. Số đoạn càng lớn và số điểm trong mỗi đoạn được chọn càng nhiều thì độ chặt chặn dưới càng cao. Để tiết kiệm không gian lưu trữ, các điểm được chọn này được biến đổi thành chuỗi nhị phân, trong đó mỗi bit được lưu trữ là 0 hay 1 tùy thuộc giá trị của điểm nằm trên hay dưới đường trung bình của đoạn chứa điểm đó. Chuỗi bit cùng với giá trị trung bình của các đoạn sẽ được lưu giữ làm đặc trưng của chuỗi.

Các điểm trong mỗi đoạn có thể được chọn theo một qui luật nào đó theo trục thời gian, chẳng hạn để lấy l điểm trong mỗi đoạn, ta có thể lấy l điểm đầu hoặc cuối mỗi đoạn hoặc chọn l điểm quan trọng (như của phương pháp PIP) hay chia đoạn thành l đoạn con rồi chọn điểm giữa của mỗi đoạn con. Nếu chọn l điểm đầu hay cuối trong mỗi đoạn tuy làm cho tăng độ chặt chặn dưới nhưng vì các điểm dữ liệu được chọn không phân bố đều trên đoạn nên không thể hiện được hình dạng xấp xỉ của đoạn đó. Nếu chọn theo l điểm PIP, tuy có thể thể hiện hình dạng xấp xỉ của đoạn nhưng phí tổn về thời gian sẽ tăng cao khi số điểm được chọn tăng (thu giảm số chiều theo cách này có độ phức tạp là $O(n\log_2 k)$, với n là chiều dài chuỗi thời gian và k là số điểm quan trọng được chọn trong mỗi đoạn). Nếu chọn theo điểm giữa của l đoạn con, ngoài việc tăng độ chặt của chặn dưới ta có thể lưu được hình dạng xấp xỉ của đoạn, nhưng độ phức tạp tính toán vẫn là $O(n)$, tương đương với độ phức tạp tính toán của phương pháp PAA hay xén dữ liệu. Trong đề tài này, chúng tôi thực hiện theo cách chọn điểm giữa của l đoạn con.

Với phương pháp MP_C, khi xem xét độ tương tự giữa hai chuỗi, ta có thể xem xét độ tương tự về mặt giá trị kết hợp với tương tự về mặt hình dạng bằng cách tịnh tiến cho hai đường trung bình của các đoạn tương ứng trùng nhau rồi so sánh các điểm được chọn dựa trên các bit biểu diễn và cộng thêm khoảng cách giữa các trung bình đoạn.

Để biểu diễn chuỗi thời gian dựa vào phương pháp MP_C ta thực hiện như sau:

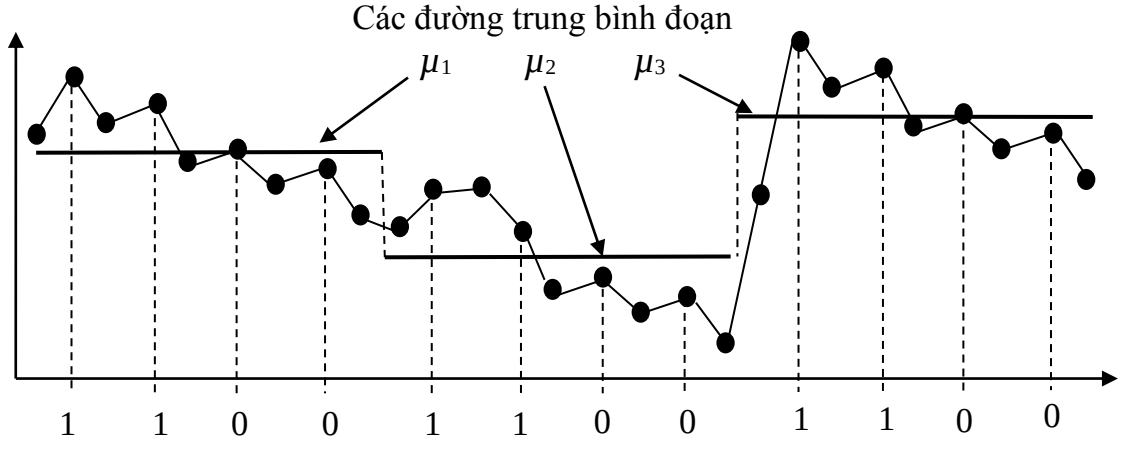
Cho một cơ sở dữ liệu S gồm k chuỗi thời gian $S = \{C_1, ..., C_k\}$ và một chuỗi truy vấn $Q = q_1, ..., q_n$. Không mất tính tổng quát, giả sử các chuỗi trong cơ sở dữ liệu S có cùng chiều dài n và mỗi chuỗi coi như một đoạn. Chia đoạn $C_i = (c_1, ..., c_n)$ thành l đoạn con bằng nhau ($l \leq n$), chọn ra các điểm giữa của l đoạn con và tính trung bình của đoạn. Để giảm thiểu dung lượng bộ nhớ cần lưu các đặc trưng cho đoạn, kỹ thuật MP_C lưu các điểm giữa được chọn dưới dạng chuỗi bit b theo công thức tương tự công thức (1.6)

$$b_t = \begin{cases} 1 & \text{Nếu } c_t > \mu \\ 0 & \text{ngược lại} \end{cases}$$

Trong đó, μ là giá trị trung bình của đoạn.

c_t là giá trị điểm giữa của đoạn con t, với $t = 1, \ldots, l$.

Ý tưởng chính khi so sánh chuỗi truy vấn Q với một chuỗi thời gian C trong cơ sở dữ liệu là biến đổi Q vào cùng không gian đặc trưng như C ngoại trừ các điểm giữa không cần chuyển sang chuỗi bit. Sau đó di chuyển đường trung bình của Q trùng với đường trung bình của C nhằm so sánh sự giống nhau về hình dạng của hai chuỗi. Ngoài ra, cần cộng thêm khoảng cách giữa hai đường trung bình của hai chuỗi để tính toán sự sai biệt về mặt giá trị. Hình 1.11 minh họa trực quan phương pháp này với số đoạn $N = 3$ và số điểm giữa được chọn trong mỗi đoạn $l = 4$. Trong ví dụ này ta sẽ lưu μ_1 và 1100 ; μ_2 và 1100; μ_3 và 1100.



Hình 1.11 Minh họa phương pháp MP_C.

1.3 Rời rạc hóa chuỗi thời gian.

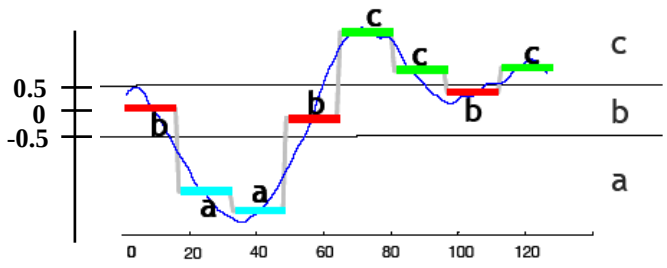
Rời rạc hóa (discretization) chuỗi thời gian là quá trình biến đổi chuỗi thời gian thành một chuỗi các ký tự. Phương pháp rời rạc hóa tiêu biểu là *phương pháp xấp xỉ*

gộp ký hiệu hóa (Symbolic Aggregate approXimation - SAX) [34] và các biến thể của nó như *phương pháp xấp xỉ gộp ký hiệu hóa mở rộng* (Extended SAX - ESAX) [33], *phương pháp xấp xỉ gộp ký hiệu có thể được lập chỉ mục* (indexable SAX - iSAX) [43].

Phương pháp xấp xỉ gộp ký hiệu hóa do Lin và cộng sự đã đề xuất năm 2003. Phương pháp này được thực hiện như sau: đầu tiên dữ liệu chuỗi thời gian được thu giảm số chiều theo phương pháp PAA. Sau đó, dựa trên giá trị trung bình cộng của từng đoạn, phương pháp này sẽ ánh xạ chúng thành một chuỗi các ký hiệu rời rạc bằng cách sử dụng các *điểm ngắt* (breakpoint). Các giá trị điểm ngắt được lựa chọn dựa trên bảng xác suất của phân bố Gauss nhằm có một xác suất bằng nhau cho mỗi ký hiệu được sử dụng trong bộ ký hiệu được dùng để rời rác hóa chuỗi thời gian. Giả sử, gọi a là kích thước bộ ký hiệu được dùng để rời rạc hóa chuỗi thời gian, cho α_i là ký hiệu thứ i trong bộ ký hiệu và ta đã tìm được các điểm ngắt có giá trị $\beta_1, \beta_2, \ldots, \beta_{a-1}$ với $\beta_1 < \beta_2 < \ldots < \beta_{a-1}$. Chuỗi thời gian $T = t_1, \ldots, t_w$ sẽ được rời rạc hóa thành chuỗi ký hiệu $C = c_1 c_2 \ldots c_w$. Trong đó mỗi phần tử c_i được ánh xạ thành một ký hiệu trong bộ ký hiệu theo công thức sau:

$$c_i = \begin{cases} \alpha_1 & t_i \le \beta_1 \\ \alpha_a & t_i > \beta_{a-1} \\ \alpha_k & \beta_{k-1} < t_i \le \beta_k \end{cases}$$

Phương pháp này biểu diễn dữ liệu chuỗi thời gian thành dạng chuỗi nên từ đó có thể áp dụng các kỹ thuật xử lý trên dữ liệu chuỗi ký tự để thực hiện xử lý, phân tích dữ liệu chuỗi thời gian. Tuy nhiên phương pháp này không hỗ trợ tốt việc tính khoảng cách Euclid và dữ liệu chuỗi thời gian được giả định là phải thỏa phân bố xác suất Gauss. Hình 1.12 minh họa phương pháp SAX.



Hình 1.12 Minh họa phương pháp SAX với $a = 3$ ([34]).

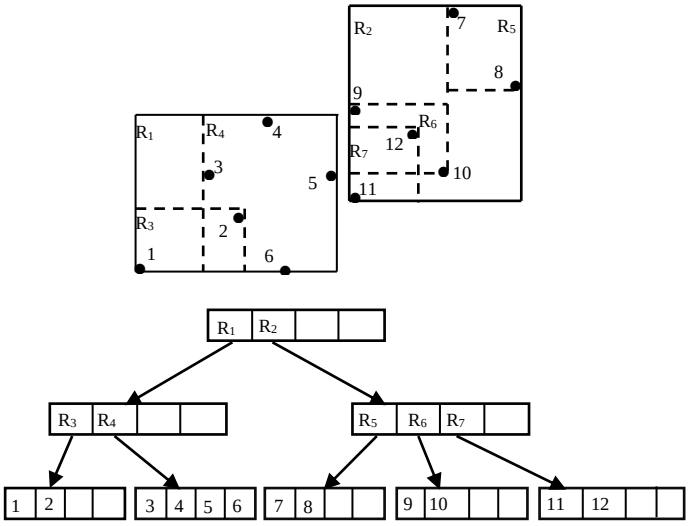
1.4 Cấu trúc chỉ mục đa chiều.

- **R-tree.**

Việc sử dụng cấu trúc chỉ mục cho phép chúng ta tìm kiếm các chuỗi tương tự nhau một cách nhanh chóng và hiệu quả nhằm đáp ứng yêu cầu về độ phức tạp tính toán thấp của các giải thuật khai phá dữ liệu chuỗi thời gian.

Cấu trúc chỉ mục đa chiều thông dụng cho chuỗi thời gian là R-tree và các biến thể của nó ([14], [4]). R-tree là một cây cân bằng cao tương tự như B-tree.

Trong một cấu trúc chỉ mục R-tree, mỗi nút trong cây chứa từ *m* đến *M* phần tử trừ khi nút đó là nút gốc (nút gốc có thể có ít nhất 2 phần tử). Chặn dưới *m* được sử dụng nhằm tránh sự suy biến của cây. Khi số phần tử trong một nút nhỏ hơn *m*, nút đó sẽ bị xóa và các phần tử của nút sẽ được cấp phát lại cho các nút kế cận. Chặn trên *M* nhằm mục đích đảm bảo mỗi một nút có thể lưu trữ được một trang dữ liệu đĩa (disk page). Mỗi phần tử trong một nút không phải lá chứa *một vùng bao chữ nhật nhỏ nhất* (Minimum Bounding Rectangle – MBR) và một con trỏ đến nút con của nó. Một MBR tại phần tử trong một nút là một vùng nhỏ nhất bao các MBR của các nút con của nó. Mỗi phần tử trong nút lá chứa một MBR của chuỗi thời gian và một con trỏ đến đối tượng dữ liệu nguyên thủy được bao bởi MBR.



Hình 1.13 Minh họa R-tree.

Điểm yếu của R-tree là các MBR trong các nút trên cùng một mức có thể phủ lấp nhau. Sự phủ lấp (overlap) này có thể làm giảm hiệu quả thực thi của việc tìm kiếm dựa vào chỉ mục. Hình 1.13 minh họa các MBR và R-tree tương ứng.

Tác vụ tìm kiếm trong R-tree tương tự như tác vụ tìm kiếm trong B-tree. Tại mỗi nút nội, các phần tử cùng với nút con của nó sẽ được kiểm tra xem MBR của phần tử đó có giao với vùng bao MBR của chuỗi truy vấn không.

Để chèn một chuỗi mới vào R-tree, giải thuật sẽ chèn vùng bao MBR của chuỗi và con trỏ tới nó vào cây. Giải thuật sẽ duyệt cây dọc theo một lối đi từ nút gốc đến nút lá. Tại mỗi mức, giải thuật lựa chọn phần tử cần mở rộng vùng bao ít nhất khi chèn MBR của chuỗi mới vào. Khi đến nút lá, nếu nút còn đủ chỗ trống giải thuật sẽ chèn vùng bao MBR của chuỗi và con trỏ đến nó vào nút. Ngược lại, giải thuật sẽ tiến hành tách nút. Tiến trình tách nút có thể được lan truyền ngược từ nút lá lên trên nếu nút cha của nút bị tách không còn chỗ trống.

R*-tree là một biến thể của R-tree, do Beckmann và các cộng sự đề xuất năm 1990. Các tác giả đã cải tiến tác vụ chèn thêm đối tượng mới vào cây của R-tree bằng cách sử dụng kỹ thuật tách nút dựa trên các tiêu chuẩn tối ưu hóa [4].

- **Chỉ mục đường chân trời (Skyline index).**

Năm 2004, Li và các cộng sự đã đề xuất một kỹ thuật lập chỉ mục mới gọi là *chỉ mục đường chân trời* (Skyline index) [35]. nhằm khắc phục tình trạng phủ lấp (overlap) giữa các hình chữ nhật chặn bên trong các MBR của các chuỗi bằng cách định nghĩa một vùng bao mới gọi là *vùng bao đường chân trời* (Skyline Bounding Region - SBR) thay cho MBR. Vùng bao SBR dùng để xấp xỉ và biểu diễn một nhóm các chuỗi thời gian theo hình dạng chung của chúng.

Một SBR được định nghĩa trong cùng không gian *thời gian-giá trị* như chuỗi thời gian. SBR cho phép chúng ta định nghĩa một hàm khoảng cách là chặn dưới của khoảng cách giữa một câu truy vấn và một nhóm các chuỗi thời gian.

Vùng bao đường chân trời được định nghĩa như sau:

Cho một nhóm S gồm n chuỗi thời gian có chiều dài l, $S = \{s_1, s_2, \ldots, s_n\}$, vùng bao đường chân trời (*SBR*) của S xác định một vùng hai chiều được bao bởi hai đường chân trời trên, dưới và hai đường thẳng đứng nối hai đường chân trời tại hai điểm đầu và cuối của các chuỗi thời gian trong S. Hai đường chân trời trên (*Tsky*) và dưới (*Bsky*) của S được định nghĩa như sau:

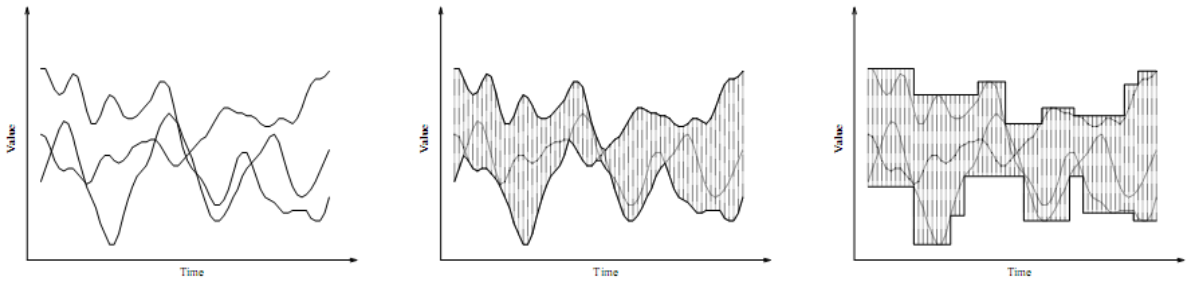
$Tsky = \{ts_1, ts_2, \ldots, ts_l\}$, $Bsky = \{bs_1, bs_2, \ldots, bs_l\}$,

trong đó, với mọi $1 \leq i \leq l$,

$ts_i = \max\{s_1[i], \ldots, s_n[i]\}$ và $bs_i = \min\{s_1[i], \ldots, s_n[i]\}$

với $s_j[i]$ là giá trị thứ i của chuỗi thời gian thứ j trong S.

Chú ý là theo định nghĩa trên thì vùng bao SBR chỉ bao gồm một vùng duy nhất và không xảy ra tình trạng phủ lấp.



(a) 3 chuỗi thời gian (b) *SBR* (c) *SBR* xấp xỉ

Hình 1.14 Minh họa *SBR* và *SBR* xấp xỉ của ba chuỗi thời gian. ([35])

Hình 1.14 minh họa SBR của ba chuỗi thời gian. Tuy nhiên, điều dễ nhận thấy là chi phí để biểu diễn SBR cho các chuỗi thời gian dài là rất cao. Vì thế tác giả chỉ sử dụng biểu diễn xấp xỉ của *Tsky* và *Bsky* cho mỗi *SBR* trong cấu trúc chỉ mục. Khi sử dụng *SBR* xấp xỉ cần phải đảm bảo là *SBR* xấp xỉ phải bao *SBR* gốc nhằm đảm bảo tính chất chặn dưới của nhóm các chuỗi thời gian. Các tác vụ tìm kiếm và chèn trên chỉ mục đường chân trời tương tự như tác vụ tìm kiếm và chèn trên R-tree.

1.5 Dự báo trên dữ liệu chuỗi thời gian có tính xu hướng hoặc mùa.

1.5.1 Tổng quan về một số phương pháp dự báo trên dữ liệu chuỗi thời gian.

Nhiều phương pháp dự báo chuỗi thời gian đã được giới thiệu và đưa vào ứng dụng trong thực tế. Một số phương pháp thường được sử dụng cho bài toán dự báo dữ liệu chuỗi thời gian như phương pháp làm trơn theo hàm mũ (exponential smoothing) ([15]), mô hình ARIMA (autoregressive integrated moving average) ([9], [30], [31]), mạng nơ ron nhân tạo (artificial neural network – ANN) ([6], [10], [16], [20], [48], [49]), logic mờ ([41]) và máy véc tơ hỗ trợ ([41], [32], [38]). Trong đó, phương pháp làm trơn theo hàm mũ và mô hình ARIMA là các mô hình tuyến tính vì chúng chỉ có thể nắm bắt được các đặc trưng tuyến tính của chuỗi thời gian, còn ANN là một mô hình phi tuyến đã được sử dụng cho bài toán dự báo dữ liệu chuỗi thời gian. Tuy nhiên vấn đề mô hình ANN có thể xử lý một cách hiệu quả dữ liệu có tính xu hướng và tính

mùa hay không đang là một vấn đề gây bàn cãi vì có những nhận định trái ngược nhau trong cộng đồng nghiên cứu về dự báo dữ liệu chuỗi thời gian [49].

Năm 2007, Nayak và te Braak đã đề xuất phương pháp dự báo cho dữ liệu thị trường chứng khoán sử dụng thuật toán gom cụm [39]. Phương pháp này dựa trên ý tưởng là một cụm được hình thành quanh một biến cố có thể được dùng để ước lượng cho biến cố ở tương lai. Cụm đó cần được xác định với bán kính nhỏ nhất có thể.

Năm 2004, Lora và các cộng sự đã đề xuất một phương pháp dự báo được gọi là phương pháp dự báo dựa vào chuỗi mẫu (pattern sequence-based forecasting – PSF) [36]. Phương pháp này sử dụng thuật toán k-Means để gom cụm dữ liệu và phát sinh ra một chuỗi các nhãn phân cụm. Cuối cùng phương pháp thực hiện dự báo dựa trên các nhãn này. Cách tiếp cận này đã giới thiệu một phương pháp luận mới có thể cung cấp các qui luật dự báo dựa trên các nhãn dữ liệu thu được một cách tự động từ thuật toán gom cụm. Năm 2011, phương pháp này đã được ứng dụng dự báo giá thị trường điện và nhu cầu sử dụng điện [2]. Tuy nhiên, qua thực nghiệm chúng tôi thấy rằng kết quả dự báo phụ thuộc vào số cụm và việc xác định số cụm tốt nhất bằng cách gom cụm nhiều lần để chọn ra số cụm tốt nhất sẽ tốn nhiều thời gian. Ngoài ra, trong một số trường hợp bất thường, nếu các mẫu tìm kiếm không có trong tập huấn luyện, phương pháp này không thể dự báo các biến cố ở tương lai ngay cả khi chiều dài của mẫu là 1.

Năm 2009, Jiang và các cộng sự đề nghị một phương pháp dự báo chuỗi thời gian chứng khoán dựa vào thông tin motif [24]. Sau khi phát hiện ra motif quan trọng nhất trong một chuỗi thời gian, motif đó được chia làm hai phần: tiền tố (prefix) và hậu tố (postfix). Nếu mẫu hiện hành của dữ liệu chuỗi thời gian khớp với tiền tố của motif, thì ta có thể dự đoán trị của bước thời gian kế tiếp dựa vào hậu tố của motif. Do giải thuật phát hiện motif được dùng trong công trình này không được hữu hiệu, nên độ chính xác dự báo và độ hữu hiệu về thời gian tính toán của phương pháp dự báo dựa vào motif chưa cao.

Một số phương pháp dự báo dựa vào k-lân cận gần nhất cũng đã được đề xuất. Năm 2005, Sorjamaa và các cộng sự đề xuất phương pháp sử dụng thông tin hỗ tương (mutual information) giữa các đối tượng và k-lân cận gần nhất để dự báo dài hạn trên dữ liệu chuỗi thời gian [45]. Năm 2007, Lora và các cộng sự đã sử dụng kỹ thuật lân cận gần nhất có trọng số (weighted nearest neighbors) để dự báo dữ liệu chuỗi thời

gian về thị trường giá điện Tây Ban Nha [37]. Năm 2010 và 2011, Huang và các cộng sự đề xuất một chiến lược kết hợp k-lân cận gần nhất với mô hình máy véc tơ hỗ trợ bình phương tối thiểu (least square support vector machine – LS-SVM) để dự báo dài hạn trên dữ liệu chuỗi thời gian ([21], [22]). Sau đó, các tác giả này đã cải tiến phương pháp trên bằng cách kết hợp thêm với mô hình tự hồi quy (autoregressive model – AR), theo đó kỹ thuật k-lân cận gần nhất và mô hình máy véc tơ hỗ trợ bình phương tối thiểu được dùng để phát sinh ra các giá trị dự báo, rồi sau đó, mô hình tự hồi quy được sử dụng để kết hợp các giá trị dự báo thu được ở bước trước nhằm tạo ra giá trị dự báo cuối cùng [23].

Năm 2013, Huong và các cộng sự đề xuất cải tiến quá trình học của mạng nơ ron nhân tạo bằng cách kết hợp phương pháp lan truyền ngược (*back-propagation training*) với thuật toán DE (*Differential Evolution*) nhằm khắc phục nhược điểm của phương pháp lan truyền ngược [19]. Cũng trong năm này, Truong và các cộng sự đã kết hợp thông tin về motif phát hiện được trong chuỗi thời gian và mạng nơ ron nhân tạo và dùng cho bài toán dự báo trên chuỗi thời gian[47].

1.5.2 Xu hướng và tính mùa trong dữ liệu chuỗi thời gian.

Xu hướng trong một chuỗi thời gian là sự thay đổi dài hạn về biên độ của dữ liệu. Nếu trong một giai đoạn tương đối dài, biên độ dữ liệu lớn dần, ta bảo dữ liệu có xu hướng tăng. Nếu biên độ của dữ liệu giảm dần theo thời gian, ta bảo dữ liệu có xu hướng giảm. Xu hướng trong chuỗi thời gian thường vi phạm điều kiện về *tính dừng* (stationarity) của dữ liệu chuỗi thời gian.

Tính mùa được định nghĩa như là một khuynh hướng của dữ liệu mà có hành vi lặp lại chính nó cứ sau một chặng *s* điểm dữ liệu. Do đó, *s* được gọi là *chiều dài một mùa* của dữ liệu.

1.5.3 Dự báo chuỗi thời gian bằng mạng nơ ron nhân tạo.

Khác với mô hình ARIMA là cách tiếp cận dự báo *điều khiển bởi mô hình* (model-driven) thường đòi hỏi phải xác định loại quan hệ giữa các biến trong mô hình và sau đó phải xác định các thông số của mô hình, hai cách tiếp cận mạng nơ ron nhân tạo và *k*-lân cận gần nhất là những cách tiếp cận dự báo *điều khiển bởi dữ liệu* (data-driven) mà không đòi hỏi phải biết trước những tính chất nền tảng của dữ liệu chuỗi

thời gian đang xét là gì. Do vậy, cả hai cách tiếp cận mạng nơ ron nhân tạo và *k*-lân cận gần nhất có những đặc điểm hấp dẫn để được sử dụng trong dự báo dữ liệu chuỗi thời gian.

Có nhiều mô hình mạng nơ ron nhân tạo đã được đề nghị từ những năm 1980. Những mô hình mạng nổi bật nhất là mô hình mạng nơ ron nhiều tầng (multi-layer perceptrons – MLP), mạng Hopfield, và mạng tự tổ chức Kohonen. Trong phạm vi nghiên cứu này, chúng tôi tập trung vào mô hình mạng nơ ron nhiều tầng.

Một mạng nơ ron MLP bao gồm nhiều tầng. Tầng thứ nhất là tầng nhập, dùng để tiếp nhận thông tin nhập. Tầng cuối cùng là tầng xuất, là nơi nhận được lời giải của bài toán. Giữa tầng nhập và tầng xuất là một số tầng trung gian, được gọi là các tầng ẩn. Các nút giữa các tầng kế tiếp nhau được nối bằng các *cung liên kết* (link arc) từ một tầng thấp hơn sang một tầng cao hơn.

Đối với công tác dự báo dữ liệu chuỗi thời gian, các trị nhập là các giá trị quan sát được trong quá khứ của chuỗi thời gian và trị kết xuất là một giá trị dự báo ở tương lai. Mô hình mạng nơ ron nhân tạo thực hiện một ánh xạ hàm như sau:

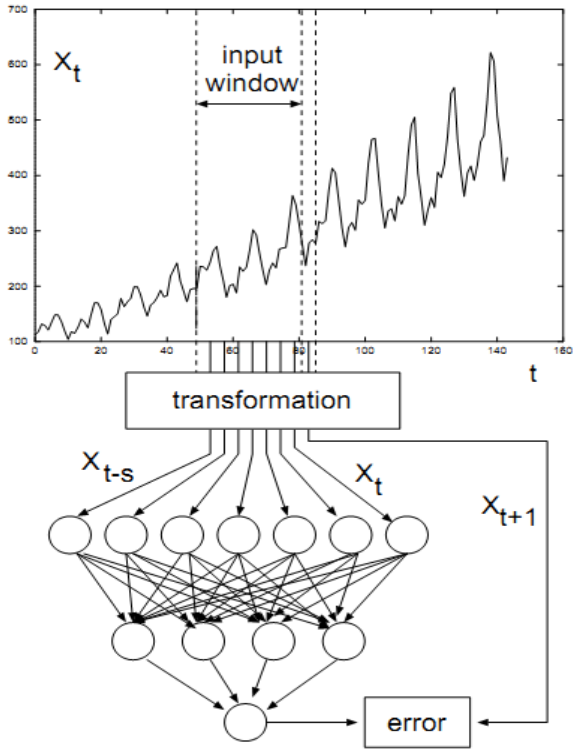
$$X_{t+1} = f(X_t, X_{t-1}, \ldots, X_{t-p})$$

trong đó X_t là giá trị quan sát tại thời điểm t.

Trước khi một mạng nơ ron có thể được dùng cho một ứng dụng nào đó, ta phải huấn luyện mạng để nó có thể thực hiện ứng dụng đó. Huấn luyện là quá trình xác định trọng số tại các cung liên kết. Các trọng số này là những thành tố quan trọng của một mạng nơ ron. Tri thức mà một mạng nơ ron học được sẽ được lưu trữ tại các cung liên kết và các nút dưới hình thức của các *trọng số cung liên kết* (link arc weights) và các *độ lệch* (bias) tại các nút. Chính nhờ thông qua các cung liên kết này mà mạng nơ ron có thể thực hiện được những ánh xạ phi tuyến phức tạp từ các nút nhập đến các nút xuất. Một quá trình huấn luyện mạng nơ ron nhiều lớp là một quá trình học có giám sát trong đó quá trình học được lặp lại cho đến khi kết quả (dữ liệu đầu ra) của ANN đạt được giá trị mong muốn (desired value) đã biết [50]. Điển hình cho kỹ thuật này là *giải thuật lan truyền ngược* (backpropagation algorithm).

Dữ liệu nhập để huấn luyện có dạng véc tơ của các biến nhập hay là các mẫu huấn luyện (training pattern). Tương ứng với mỗi thành phần trong một véc tơ nhập là một nút nhập tại tầng nhập. Do đó số nút nhập bằng với số chiều của véc tơ nhập. Bất

luận số chiều đó ra sao, véc tơ nhập dùng trong một hệ thống dự báo chuỗi thời gian sẽ gồm một *cửa sổ trượt* (sliding window) có chiều dài cố định đi dọc qua chuỗi thời gian (xem Hình 1.15)



Hình 1.15 Quá trình huấn luyện mạng nơ ron dùng cho dự báo dữ liệu chuỗi thời gian ([126]).

Quá trình huấn luyện mạng nơ ron để dự báo dữ liệu chuỗi thời gian được thực hiện như sau. Dùng một cửa sổ trượt kích thước p trượt qua chuỗi thời gian và mạng nơ ron coi chuỗi thời gian $X_1, X_2, ..., X_n$ như thể gồm nhiều ánh xạ chuyển từng véc tơ nhập thành một trị xuất. Một chuỗi con gồm s điểm được cửa sổ trượt trích ra từ chuỗi dữ liệu để đưa vào các nút ở tầng nhập. Các giá trị nhập này được đánh trọng số và cộng tích lũy tại mỗi nút của tầng ẩn đầu tiên. Trị tổng này sẽ được biến đổi bằng một *hàm truyền*, thí dụ như hàm sigmoid $f(x) = 1/(1+e^{-x})$, thành ra trị kết xuất tại nút ấy. Trị này đến lượt nó trở thành trị nhập đi vào những nút ở tầng kế tiếp để cuối cùng tạo thành trị của nút xuất. Sai số giữa giá trị tại nút xuất với giá trị của chuỗi thời gian tại thời điểm $t+1$ (tức X_{t+1}) sẽ là giá trị lỗi được dùng cho *giải thuật lan truyền ngược* (backpropagation algorithm). Sai số này được truyền ngược đến các cung liên kết giữa tầng xuất và tầng ẩn, rồi đến các cung liên kết giữa tầng ẩn và tầng nhập. Sau khi tất cả các trọng số của tất cả các cung liên kết trong mạng được cập nhật, quá trình huấn

luyện coi như đã hoàn tất một mẫu huấn luyện nạp vào mạng. Khi toàn bộ chuỗi thời gian được cửa sổ trượt duyệt qua và nạp vào mạng, quá trình huấn luyện coi như đã hoàn tất một *chuyến lặp* (epoch). Quá trình huấn luyện bằng lan truyền ngược có thể phải lặp lại nhiều chuyến lặp như vậy trước khi có thể thỏa mãn điều kiện dừng của giải thuật lan truyền ngược.

- **Xác định kiến trúc của mạng nơ ron MLP cho công tác dự báo chuỗi thời gian.**

Về kiến trúc của mạng nơ ron được dùng trong nghiên cứu này, chúng tôi phải quyết định về 4 lựa chọn: số tầng ẩn, số nút trong tầng nhập, số nút trong tầng xuất và số nút trong tầng ẩn.

- Về số tầng ẩn, trong nghiên cứu này chúng tôi dùng cấu trúc mạng nơ ron một tầng ẩn. Có nhiều nghiên cứu thực nghiệm và lý thuyết đã chứng tỏ rằng chỉ cần với một tầng ẩn, mạng nơ ron vẫn có thể xấp xỉ được bất kỳ hàm phi tuyến phức tạp nào ([50]).
- Về số nút của tầng nhập, đối với công tác dự báo dữ liệu chuỗi thời gian, vẫn chưa có công trình nghiên cứu lý thuyết nào nêu cách xác định con số thích hợp cho số nút của tầng nhập. Tuy vậy, bài báo năm 2008 của Hamzaçebi có khuyến cáo rằng số nút của tầng nhập nên chọn bằng chiều dài s của mùa khi dùng mạng nơ ron để dự báo dữ liệu chuỗi thời gian có tính mùa [50]. Thí dụ, ta nên chọn số nút tầng nhập là 12 khi chiều dài của mùa là ứng với một năm gồm 12 tháng, và nên chọn số nút tầng nhập là 4 khi chiều dài của mùa ứng với một năm gồm 4 quý. Trong nghiên cứu này, chúng tôi áp dụng khuyến cáo của Hamzaçebi khi xác định số nút tầng nhập cho mạng nơ ron dùng để dự báo dữ liệu chuỗi thời gian có tính mùa.
- Về số nút của tầng xuất, con số này tùy thuộc vào tầm dự báo (prediction horizon) là bao nhiêu. Mạng nơ ron nên có một nút ở tầng xuất trong trường hợp công tác dự báo là nhằm dự báo một bước về phía tương lai (one-step ahead prediction) và nên có *k* nút ở tầng xuất trong trường hợp công tác dự báo là nhằm dự báo *k* bước về phía tương lai (*k*-step ahead prediction) áp dụng phương pháp dự báo nhiều bước theo cách trực tiếp (direct multi-step method) (theo đề xuất của Zhang và các cộng sự trong [50]).

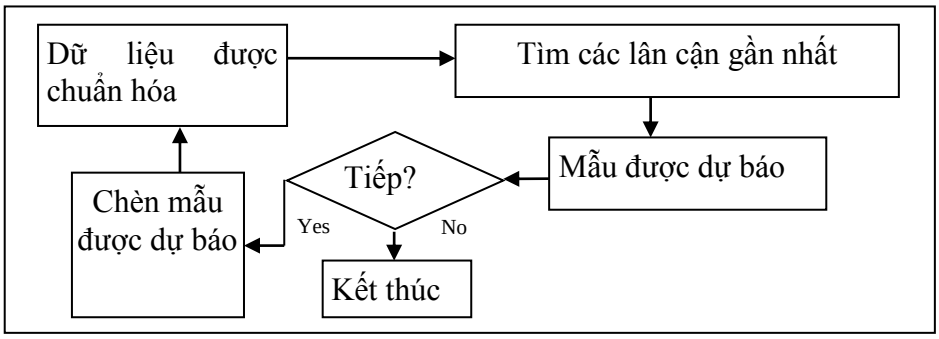
- Về số nút của tầng ẩn duy nhất trong mạng nơ ron, chúng tôi áp dụng một phương pháp xây dựng mạng nơ ron được đề xuất bởi Ash ([3]) để xác định số nút, sẽ được mô tả chi tiết trong phần thực nghiệm (chương 3).

Mặc dù mạng nơ ron là một mô hình xấp xỉ hàm phổ quát (universal function approximator), chúng không thể mô hình được một cách trực tiếp những biến đổi có tính mùa và tính xu hướng trong dữ liệu chuỗi thời gian. Theo Zhang và các cộng sự, 2005, lý do của hiện tượng này là vì cũng giống như các mô hình thống kê truyền thống, mạng nơ ron không thể đồng thời xử lý nhiều thành phần khác nhau nằm tiềm tàng trong một chuỗi thời gian ([49]).

CHƯƠNG 2. Phương pháp đề xuất.

Trong đề tài này, bài toán dự báo chuỗi thời gian có tính xu hướng hoặc biến đổi theo mùa được thực hiện dựa trên việc so trùng mẫu. Chúng tôi sử dụng thuật toán tìm *k* lân cận gần nhất hoặc tìm lân cận trong phạm vi một ngưỡng cho trước dựa trên một cấu trúc chỉ mục đa chiều.

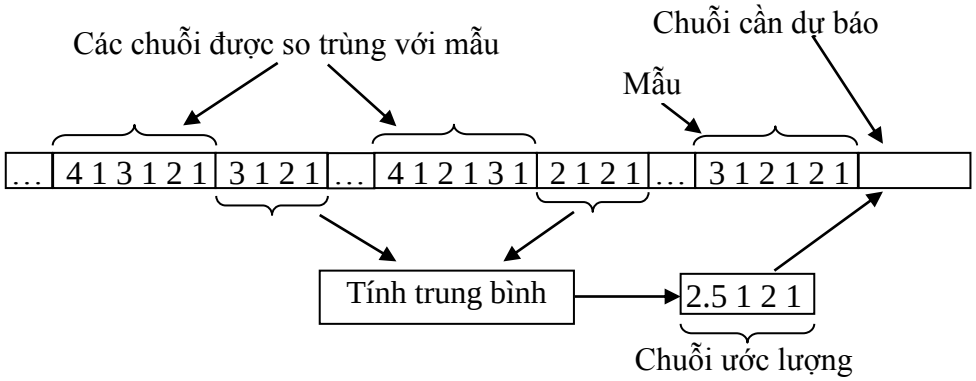
Cách tiếp cận *k*-lân cận gần nhất là một trong những kỹ thuật dự báo *phi tham số* (non-parametric), hiểu theo nghĩa người dùng không phải biết trước mối quan hệ lý thuyết nào giữa các trị xuất và các trị nhập trong bài toán dự báo, do đó nó rất tự nhiên và trực giác. Ý tưởng chính của cách tiếp cận này là nhận dạng các mẫu trong quá khứ khớp với mẫu hiện hành và dùng tri thức về cách mà chuỗi thời gian biến đổi trong quá khứ trong những tình huống tương tự để dự báo về biến đổi trong tương lai. Ngoài ra, với cách tiếp cận *k*-lân cận gần nhất này, các mẫu dự báo có thể được hồi tiếp trở lại vào tập dữ liệu để sử dụng cho các lần dự báo sau, nhờ vậy tầm (horizon) của dự báo có thể được kéo dài theo yêu cầu (kỹ thuật này được gọi là *dự báo lặp* – iterated prediction). Hình 2.1 trình bày ý tưởng cơ bản của cách tiếp cận này.



Hình 2.1 Ý tưởng cơ bản của cách tiếp cận dựa trên phương pháp so trùng mẫu.

Cho một trạng thái (mẫu) hiện hành có chiều dài *w* trong chuỗi thời gian có chiều dài *n* ($w << n$) và chúng ta phải dự đoán chuỗi có chiều dài *m* ($m \leq w$) sẽ xảy ra ở bước kế tiếp theo thời gian (tức là dự báo *m* bước về phía tương lai). Đầu tiên, thuật toán sẽ tìm kiếm *k* lân cận gần nhất hay các lân cận trong một ngưỡng *T* cho trước đối với mẫu đó. Sau đó, thuật toán lấy các chuỗi có chiều dài *m* nằm kế cận bên phải của các lân cận gần nhất tìm được ở bước trên. Cuối cùng, chuỗi dự báo được ước lượng bằng cách tính trung bình cộng các chuỗi vừa thu được. Trong trường hợp cần dự báo cho

các chuỗi khác nữa, chuỗi ước lượng có thể được chèn vào cuối tập dữ liệu để dự báo cho các mẫu tiếp theo.



Hình 2.2 Minh họa thuật toán dự báo dựa trên phương pháp so trùng mẫu.

Hình 2.2 minh họa bằng thí dụ thuật toán được đề xuất và Hình 2.3 trình bày các bước chính của thuật toán này. Trong Hình 2.3, D là chuỗi thời gian có chiều dài n_1, TS là tập kiểm tra có chiều dài n_2, w là chiều dài của mẫu, và m là chiều dài của chuỗi dự báo ($m \leq w < n_2$ và $w << n_1$).

Chú ý là trong trường hợp $m < w$, chúng ta có thể dùng một biến để lưu tích lũy các chuỗi ước lượng cho tới khi m bằng với w. Khi đó, chúng ta có thể chèn chuỗi tích lũy được vào trong cấu trúc chỉ mục mà không cần phải xây dựng lại cấu trúc chỉ mục khi quay lại thực hiện bước 1.

1. Thu giảm số chiều các chuỗi con có chiều dài w trong D và chèn chúng vào trong một cấu trúc chỉ mục đa chiều (nếu cần).
2. Lấy chuỗi S (mẫu) có chiều dài w nằm trước vị trí chuỗi ta phải dự báo trong TS.
3. Tìm k lân cận gần nhất (hay các lân cận nằm trong phạm vi ngưỡng T) của S.
4. Với mỗi lân cận gần nhất tìm được ở bước 3, khôi phục chuỗi có chiều dài m nằm kế cận nó trong D.
5. Tính trung bình cộng các chuỗi tìm được ở bước 4.
6. Trả lại kết quả ước lượng ở bước 5.
7. Chèn chuỗi ước lượng ở bước 5 vào D để dự báo các mẫu tiếp sau và quay lại bước 1 (nếu cần).

Hình 2.3 Các bước chính của thuật toán dự báo dựa trên phương pháp so trùng mẫu.

Từ giải thuật ở Hình 2.3 ta có thể thấy khác với các mô hình thống kê và cả mô hình ANN thường phải xây dựng mô hình từ tập dữ liệu có sẵn (tức là quá trình học),

phương pháp *k*-lân cận gần nhất coi tập huấn luyện chính là mô hình, do vậy nó tiến hành dự báo trực tiếp dựa vào tập huấn luyện mà không qua một quá trình học nào cả.

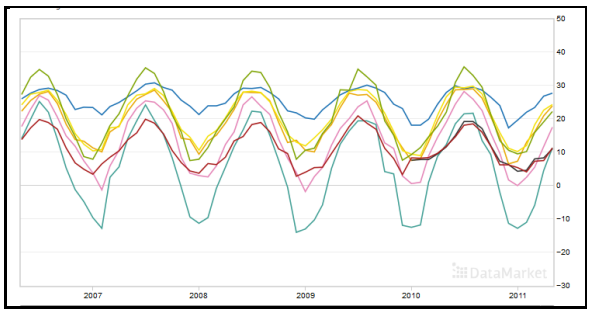
Trong giải thuật ở Hình 2.3 có ba tham số phải xác định: độ đo được dùng để xác định độ tương tự của hai chuỗi con, số lân cận gần nhất *k* (hay ngưỡng tương tự *T*) cần tìm và chiều dài *w* của mẫu để so trùng. Độ đo được chọn để dùng là độ đo Euclid. Việc xác định giá trị của *k* có ảnh hưởng đến chất lượng của dự báo của giải thuật *k*-lân cận gần nhất. Trong thực tế, giá trị tốt nhất của *k* thường nhỏ đối với dữ liệu chuỗi thời gian không có nhiễu. Về chiều dài *w* của mẫu, việc xác định *w* là tùy thuộc vào tính mùa của dữ liệu, nếu dữ liệu chuỗi thời gian có chiều dài của mùa là *s* thì ta nên chọn *w* bằng với *s*.

CHƯƠNG 3. Kết quả thực nghiệm.

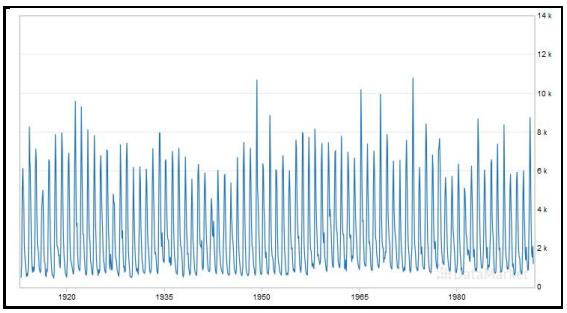
Trong thực nghiệm, các giải thuật do chúng tôi đề xuất được viết bằng ngôn ngữ *C#* và chạy trên máy *Core 2 Duo 1.60 GHz, 1.00 GB RAM*.

Thực nghiệm được thực hiện trên bốn tập dữ liệu thực có tính xu hướng hoặc biến đổi theo mùa: Temperatures at Savannah International Airport, Fraser river (FR), Milk production (MP) and Carbon dioxide (CD). Chúng tôi so sánh sự thực thi của cách tiếp cận này với sự thực thi của phương pháp mạng nơ ron nhân tạo (ANN).

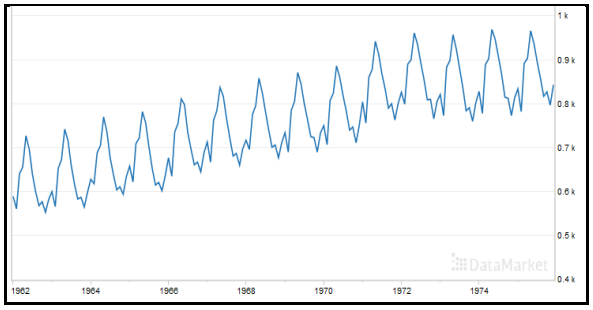
- **Mô tả các tập dữ liệu thử nghiệm.**



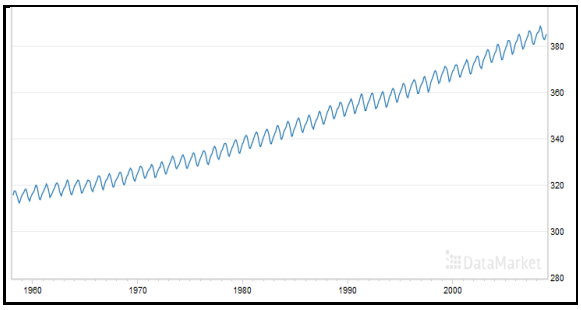
Tập dữ liệu Temperature



Tập dữ liệu Fraser river



Tập dữ liệu Milk production



Tập dữ liệu Carbon Dioxide

Hình 3.1 Minh họa bốn tập dữ liệu dùng trong thực nghiệm ([17]).

Các tập dữ liệu dùng trong thực nghiệm được mô tả như sau.

- Tập dữ liệu Temperatures at Savannah International Airport, từ 1/1910 đến 12/2010. Tập dữ liệu huấn luyện được lấy từ 1/1910 đến 12/2000 và tập dữ liệu dùng để kiểm tra được lấy từ 1/2001 đến 12/2010.
- Tập dữ liệu Fraser River, từ 1/1913 đến 12/1990. Tập dữ liệu huấn luyện được lấy từ 1/1913 đến 12/1982 và tập dữ liệu dùng để kiểm tra được lấy từ 1/1983 đến 12/1990.

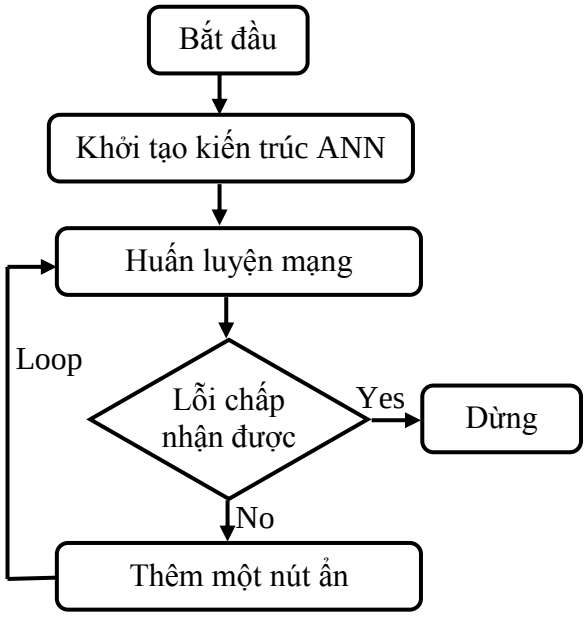
- Tập dữ liệu Milk production, từ 1/1962 đến 12/1975. Tập dữ liệu huấn luyện được lấy từ 1/1962 đến 12/1971 và tập dữ liệu dùng để kiểm tra được lấy từ 1/1972 đến 12/1975.

Tập dữ liệu Carbon dioxide, từ 1/1959 đến 12/2008. Tập dữ liệu huấn luyện được lấy từ 1/1959 đến 12/1998 và tập dữ liệu dùng để kiểm tra được lấy từ 1/1999 đến 12/2008.

Tất cả các tập dữ liệu này là những chuỗi thời gian có tính xu hướng và tính mùa, được lấy từ nguồn [17]. Hình 3.1 minh họa các tập dữ liệu trên dưới dạng đồ họa.

- **Xác định kiến trúc mạng ANN dùng trong thực nghiệm.**

Chương trình hiện thực mô hình ANN sử dụng phần mềm mạng nơ ron chuyên dụng Spice-Neuro [46]. Hàm truyền được sử dụng trong tầng ẩn và tầng xuất của mạng nơ ron là hàm sigmoid. Giải thuật huấn luyện mạng là giải thuật lan truyền ngược. Cấu trúc mạng nơ ron được xác định như sau: 12 nút nhập và 3 nút xuất vì mọi tập dữ liệu đều là dữ liệu quan sát hàng tháng (monthly time series) với chiều dài của mùa là 12 và dự báo ở đây là dự báo 3 bước về phía tương lai nên cần 3 nút xuất. Để xác định số nút trong tầng ẩn duy nhất chúng tôi áp dụng một phương pháp xác định được đề xuất bởi Ash ([3]). Hình 3.2 mô tả phương pháp này.



Hình 3.2 Giải thuật xây dựng mạng nơ ron của Ash.

Giải thuật xây dựng cấu trúc mạng nơ ron bao gồm các bước:

- Bước 1: Tạo một ANN ban đầu bao gồm ba tầng: tầng nhập, tầng ẩn và tầng xuất. Số lượng các nút trong tầng nhập và tầng xuất tùy thuộc vào chiều dài của mùa và số bước dự báo như đã nêu. Ban đầu tầng ẩn chỉ có 1 nút. Ngẫu nhiên khởi tạo trọng số các cung liên kết trong một phạm vi giá trị nhất định.
- Bước 2: Sử dụng tập huấn luyện để huấn luyện mạng bằng giải thuật lan truyền ngược cho đến khi tỉ lệ lỗi nhỏ hơn một ngưỡng θ cho trước.
- Bước 3: Tính toán lỗi của ANN dựa vào một tập kiểm tra. Nếu tỉ lệ lỗi được tìm thấy không thể chấp nhận được (quá lớn) có nghĩa là kiến trúc hiện tại của ANN là không phù hợp, ta chuyển qua bước tiếp theo.
- Bước 4: Thêm một nút ẩn vào tầng ẩn. Khởi tạo ngẫu nhiên trọng số của nút mới thêm và chuyển sang bước 2.

Bằng cách áp dụng giải thuật xây dựng mạng vừa nêu, chúng tôi đã xác định được số nút tầng ẩn thích hợp cho các mạng nơ ron làm việc với các tập dữ liệu thử nghiệm: 3 nút ẩn đối với tập dữ liệu Temperatures và 5 nút ẩn đối với các tập dữ liệu khác.

- **Tiêu chuẩn đánh giá.**

Chúng tôi so sánh sự thực thi của hai phương pháp dự báo trên các đoạn trong tập dữ liệu kiểm tra và tính toán lỗi trung bình trong khoảng thời gian dự báo.

Hai tiêu chuẩn đánh giá được sử dụng cho bài toán này là Lỗi trung bình tương đối so với x_{mean} (Mean error relative to x_{mean} - MER) và Lỗi trung bình tuyệt đối (Mean absolute error - MAE). Các tiêu chuẩn đánh giá này được định nghĩa như sau [2]:

- Lỗi trung bình tương đối so với to x_{mean}

$$MER = 100 \times \frac{1}{N} \sum_{i=1}^{N} \frac{\left| x_{\text{model},i} - x_{obs,i} \right|}{x_{mean}} \tag{3.1}$$

- Lỗi trung bình tuyệt đối

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| x_{\text{model},i} - x_{obs,i} \right| \tag{3.2}$$

Trong đó $x_{\text{obs},i}$ là giá trị quan sát được, $x_{\text{model},i}$ là giá trị dự báo được tại thời điểm/vị trí i, x_{mean} là giá trị trung bình trong khoảng thời gian quan tâm (ngày, tháng hay năm) và N là chiều dài của chuỗi được dự báo. Việc dùng hai độ đo lỗi này thể

hiện hai góc nhìn khác nhau khi đánh giá hai mô hình dự báo. Độ đo MAE là độ đo tuyệt đối và độ đo MER là độ đo tương đối.

- **Kết quả thực nghiệm.**

- **Thực nghiệm 1: Xác định *k* và *T* phù hợp.**

Trong thực nghiệm này chúng tôi xem xét ảnh hưởng của *k* và ngưỡng *T* đối với độ chính xác của dự báo. Lưu ý rằng với cả 4 bộ dữ liệu mẫu, chúng tôi đều chọn chiều dài của mẫu $w = 12$ vì chiều dài mùa của dữ liệu là 12 tháng. Có hai cách so trùng mẫu : dùng *k*-lân cận gần nhất hay các lân cận nằm trong phạm vi ngưỡng *T*. Bảng 3.1 trình bày các lỗi của dự báo được thực nghiệm trên tập dữ liệu Frazer river với *k* thay đổi từ 1 đến 10. Kết quả thực nghiệm cho thấy rằng các lỗi dự báo sẽ thay đổi với các giá trị *k* khác nhau. Trong thực nghiệm này, chúng ta thấy lỗi dự báo là nhỏ nhất khi *k* bằng 4.

Bảng 3.1 Lỗi dự báo khi thực nghiệm trên tập dữ liệu Frazer river với *k* thay đổi từ 1 đến 10.

k	1	2	3	4	5	6	7	8	9	10
MER (%)	26.62	29.2	23.74	22.46	24.39	24.31	23.29	22.7	23	22.66
MAE	0.055	0.060	0.049	0.046	0.050	0.050	0.048	0.047	0.047	0.047

Bảng 3.2 trình bày các lỗi của dự báo được thực nghiệm trên tập dữ liệu Frazer river với một số giá trị ngưỡng *T*. Kết quả thực nghiệm cho thấy rằng các lỗi dự báo sẽ thay đổi với các giá trị *T* khác nhau. Trong thực nghiệm này, chúng ta thấy lỗi dự báo là nhỏ nhất khi *T* bằng 0.21.

Bảng 3.2 Lỗi dự báo khi thực nghiệm trên tập dữ liệu Frazer river với một số giá trị ngưỡng *T* khác nhau.

T	0.15	0.17	0.19	0.21	0.23	0.25
MER (%)	27.94	27.05	25.64	23.11	25.29	25.91
MAE	0.056	0.055	0.052	0.047	0.051	0.052

Bảng 3.3 trình bày các lỗi của dự báo được thực nghiệm trên tập dữ liệu Frazer river với giá trị *k* tốt nhất được chọn bằng 4 cho trường hợp sử dụng thuật toán tìm *k* lân cận gần nhất và *T* tốt nhất được chọn là 0.21 cho trường hợp tìm kiếm các lân cận trong phạm vi ngưỡng cho trước. Các lỗi dự báo được tính toán cho mỗi năm. Dòng cuối của bảng là trung bình lỗi trong tám năm. Kết quả thực nghiệm cho thấy độ chính

xác dự báo của hai cách (*k* lân cận gần nhất hay các lân cận nằm trong phạm vi ngưỡng *T*) là xấp xỉ nhau.

Bảng 3.3 Lỗi dự báo của phương pháp sử dụng thuật toán *k* lân cận gần nhất so sánh với phương pháp sử dụng thuật toán tìm lân cận trong phạm vi ngưỡng *T* cho trước với giá trị *k* và *T* tốt nhất.

Year	MER (%)		MAE	
	k-NN	Range search	k-NN	Range search
1	24.27	21.87	0.06	0.06
2	18.94	16.75	0.04	0.03
3	28.48	22.39	0.06	0.05
4	15.15	26.86	0.03	0.05
5	25.77	22.66	0.05	0.05
6	32.20	28.52	0.06	0.05
7	18.57	20.86	0.04	0.04
8	21.12	25.02	0.04	0.05
Mean	23.06	24.16	0.05	0.05

- **Thực nghiệm 2: So sánh hai phương pháp k-lân cận gần nhất và ANN.**

Bảng 3.4 Lỗi dự báo của phương pháp sử dụng thuật toán *k* lân cận gần nhất so sánh với phương pháp ANN. Thực nghiệm được thực hiện trên tập dữ liệu Temperature.

Year	MER(%)		MAE	
	k-NN	ANN	*k*-NN	ANN
1	7.555	17.814	0.043	0.065
2	6.779	11.666	0.039	0.059
3	8.316	11.523	0.047	0.039
4	6.288	10.239	0.035	0.036
5	7.652	8.921	0.042	0.039
6	8.329	10.053	0.047	0.040
7	7.570	9.590	0.044	0.044
8	7.767	11.335	0.045	0.053
9	5.004	8.298	0.029	0.035
10	14.542	14.394	0.081	0.049
Mean	7.980	11.383	0.045	0.046

Trong thực nghiệm này, chúng tôi so sánh độ chính xác dự báo của hai phương pháp k-lân cận gần nhất và mô hình ANN. Bảng 3.4 trình bày các lỗi dự báo được thực nghiệm với phương pháp *k*-lân cận gần nhất trên tập dữ liệu Temperature so sánh với lỗi dự báo của phương pháp ANN. Các lỗi dự báo được tính toán cho mỗi năm. Dòng

cuối của bảng là trung bình lỗi trong mười năm. Bảng 3.5 trình bày kết quả tổng hợp từ thực nghiệm trên ba tập dữ liệu: Fraser river, Milk production và Carbon dioxide. Các giá trị trong bảng là trung bình lỗi của các năm được dự báo.

Bảng 3.5 Trung bình lỗi dự báo của phương pháp sử dụng k-NN so sánh với trung bình lỗi dự báo của phương pháp ANN.

Tập dữ liệu	MER (%)		MAE	
	k-NN	ANN	*k*-NN	ANN
FR	23.06	24.16	0.05	0.06
MP	8.06	14.73	0.09	0.10
CD	3.38	3.61	0.037	0.032

Ngoài việc so sánh về độ chính xác của kết quả dự báo, chúng tôi còn thực nghiệm so sánh về thời gian thực hiện của hai phương pháp. Bảng 3.6 trình bày kết quả thực nghiệm trên bốn tập dữ liệu về thời gian thực hiện của hai phương pháp (tính theo giây). Thời gian thực hiện đối với mô hình ANN bao gồm thời gian huấn luyện và thời gian dự báo. Chúng ta có thể thấy là phương pháp sử dụng *k*-lân cận gần nhất luôn thực hiện nhanh hơn rất nhiều so với phương pháp ANN.

Bảng 3.6 Thời gian thực hiện của hai phương pháp thực nghiệm trên bốn tập dữ liệu khác nhau.

Tập dữ liệu	ANN	*k*-NN
Temperatures	50	0.262
Milk production	4	0.464
Carbon dioxide	37	1.261
Frazer river	58	0.199

- Nhận xét.

- Các kết quả thực nghiệm cho thấy mặc dù các lỗi dự báo (MER và MAE) của phương pháp sử dụng *k*-NN trong một vài năm lớn hơn lỗi dự báo của phương pháp ANN, trung bình các lỗi MER và MAE trong các năm dự báo của phương pháp *k*-lân cận gần nhất thường nhỏ hơn so với trung bình lỗi dự báo của phương pháp ANN. Chỉ riêng trường hợp thực nghiệm trên tập dữ liệu Carbon dioxide, trung bình lỗi MAE của phương pháp sử dụng *k*-NN lớn hơn một ít so với trung bình lỗi tương

ứng của phương pháp ANN. Nhưng trung bình lỗi MER của phương pháp sử dụng *k*-NN thì nhỏ hơn trung bình lỗi tương ứng của phương pháp ANN.

- Giá trị *k* trong bài toán tìm *k*- lân cận gần nhất và ngưỡng *T* trong bài toán tìm lân cận trong phạm vi ngưỡng *T* đều có ảnh hưởng đến kết quả dự báo.
- Kết quả thực nghiệm cho thấy với giá trị *k* thích hợp, phương pháp dự báo dựa trên *k*- lân cận gần nhất có thể cho kết quả có độ chính xác tốt hơn so với phương pháp ANN khi thực hiện trên dữ liệu chuỗi thời gian có tính xu hướng hoặc biến đổi theo mùa.
- Thời gian thực hiện của phương pháp dự báo dựa trên so trùng mẫu luôn nhanh hơn rất nhiều so với phương pháp ANN khi thực hiện trên dữ liệu chuỗi thời gian có tính xu hướng hoặc biến đổi theo mùa.

CHƯƠNG 4. Kết luận và hướng phát triển.

Chương này sẽ trình bày các đóng góp trong nghiên cứu của đề tài này, một số hạn chế và hướng phát triển trong tương lai.

- **Đóng góp của đề tài.**

Đề tài đã đề xuất đề xuất một phương pháp mới cho bài toán dự báo dữ liệu chuỗi thời gian có tính xu hướng hoặc biến đổi theo mùa dựa vào cách tiếp cận so trùng mẫu (sử dụng *k*-lân cận gần nhất). Kết quả đánh giá bằng thực nghiệm cho thấy phương pháp so trùng mẫu hữu hiệu hơn ANN về cả hai phương diện độ chính xác dự báo và thời gian thực thi trong bài toán dự báo dữ liệu chuỗi thời gian có tính xu hướng hoặc biến đổi theo mùa.

- **Hạn chế của đề tài.**

Hầu hết các giải thuật khai phá dữ liệu chuỗi thời gian thường đòi hỏi phải xác định giá trị một số thông số đầu vào và việc xác định các thông số này thường không dễ dàng đối với người dùng. Việc xác định các thông số đầu vào thường đòi hỏi ở người dùng một quá trình thử-và-sửa sai (try-and-error) bằng thực nghiệm rất tốn thời gian. Giải thuật được đề xuất trong báo cáo này cũng không tránh khỏi những hạn chế nêu trên. Đó là viêc dự báo dữ liệu chuỗi thời gian bằng giải thuật k-NN hoặc lân cận trong phạm vi ngưỡng *T* cho trước, người dùng phải xác định tham số *k* và ngưỡng *T* phù hợp.

- **Hướng phát triển.**

Từ những nghiên cứu và kết quả đạt được của đề tài này, chúng tôi đề nghị hướng nghiên cứu tiếp theo như sau:

Lai ghép phương pháp *k*-lân cận gần nhất với mô hình ANN để phát huy những điểm mạnh của cả hai phương pháp này trong công tác dự báo dữ liệu chuỗi thời gian.

TÀI LIỆU THAM KHẢO

[1] R. Agrawal, C. Faloutsos, A. Swami , "Efficient similarity search in sequence databases," in *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, Chicago, 1993, pp. 69-84.

[2] F. M. Álvares, A. T. Lora, J.C. Riquelme, J.S. Aguilar Ruiz, "Energy Time Series Forecasting Based on Pattern Sequence Similarity," *Knowledge and Data Engineering, IEEE Transaction*, vol. 23, no. 8, pp. 1230-1243, Aug. 2011.

[3] T. Ash, "Dynamic node creation in backpropagation networks," *Computer Science*, vol. 1, no. 4, pp. 365-375, 1989.

[4] N. Beckmann, H. Kriegel, R. Schneider, B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," in *Proc. of 1990 ACM SIGMOD Conf.*, Atlantic City, NJ, 1990, pp. 322-331.

[5] D. Berndt and J. Clifford, "Finding Patterns in time series: a dynamic programming approach," Journal of advances in Knowledge Discovery and Data Mining, pp. 229-248, 1996.

[6] S. D. Balkin and J. K. Ord, "Automatic neural network modeling for univariate time series," *International Journal of Forecasting*, vol. 16, pp. 509-515, 2000.

[7] K. Chan and A. W. Fu, "Efficient Time Series Matching by Wavelets," in Proceedings of the 15th IEEE Int'l Conference on Data Engineering, Sydney, Australia, 1999, pp. 126-133.

[8] F.L. Chung, T.C. Fu, R. Luk, V. Ng, "Flexible Time Series Pattern Matching Based on Perceptually Important Points," in International Joint Conference on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data, 2001, pp. 1-7.

[9] C. Chatfield, *Time-series forecasting*. New York, NY: Chapman and Hall, Inc., 2000.

[10] E. Cadenas and W. Rivera, "Short-term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural network," *Renewable Energy*, vol. 34, pp. 274-278, 2009.

[11] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, "Fast Subsequence Matching in Time Series Databases," in Proceedings of the ACM SIGMOD International Conference on Management of Data, Minneapolis, NM, 1994, pp. 419-429.

[12] E. Fink, K. B. Pratt, "Indexing of compressing time series," in Mark Last, Abraham Kandel and Horst Bunke, editors. Data mining in time series Databases, World Scientific, Singapore., 2003.

[13] E. Fink, H. S. Gandhi, "Compression of time series by extracting major extrema," Journal of Experimental & Theoretical Artificial Intelligence, vol. 23, no. 2, pp. 255-270, Jun. 2011.

[14] A. Guttman, "R-trees: a Dynamic Index Structure for Spatial Searching," in *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, 1984, pp. 47-57.

[15] S. Gelper, R. Fried, C. Croux, "Robust forecasting with exponential and Hold-Winters smoothing," *Journal of Forecasting*, vol. 29, no. 3, pp. 285-300, 2010.

[16] M. Ghiassi, H. Saidane and D.K. Zimbra, "A dynamic artificial neural networkfor forecasting series events," *International Journal of Forecasting*, vol. 21, pp. 341-362, 2005.

[17] R. Hyndman. Time Series Data Library. [Online]. http://www.datamarket.com.

[18] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Second Edition ed. Morgan Kaufmann publishers, 2006.

[19] Đinh Thi Thu Huong, Cao Thi Phuong Anh and Bui Thu Lam, (2013). An Evolutionary Ensemble-based approach for Exchange Rate Forcasting. *In Proc. of 2013 World Congress on Information and Communication Technologies (WICT 2013)*, Hanoi, Vietnam, December, 15-18, 2013, pp. 111-116.

[20] S. Heravi and C. R. Birchenhall, "Linear versus neural network forecasting for European industrial production series," *International Journal of Forecasting*, vol. 20, pp. 435-446, 2004.

[21] Z. Huang and M. L. Shyu, "k-NN Based LS-SVM Framework for Long-Term Time Series Prediction," in *The 11th IEEE International Conference on Information Reuse and Integration (IRI 2010)*, Tuscany Suites & Casino, Las Vegas, Nevada, USA, 2010, pp. 69-74.

[22] Z. Huang and M.-L. Shyu, "Long-Term Time Series Prediction using k-NN Based LS-SVM Framework with Multi-Value Integration," in *Recent Trends in Information Reuse and Integration*, K. K. a. M. T. Tansel Ozyer, Ed. Springer Vienna, 2012, ch. 9, pp. 191-209.

[23] Z. Huang, M. L. Shyu, J. M. Tien, "Multi-Model Integration for Long-Term Time Series Prediction," in *The 13th IEEE International Conference on Information Reuse and Integration (IRI 2012)*, Tuscany Suites & Casino, Las Vegas, Nevada, USA, 2012.

[24] Y. Jiang, C. Li, J. Han, "Stock temporal prediction based on time series motifs," in *Proc. of 8th Int. Conf. on Machine Learning and Cybernetics*, 2009.

[25] E. Keogh, "A Tutorial on Indexing and Mining Time Series Data," in The IEEE International Conference on Data Mining (ICDM 2001), San Jose, USA, November 29, 2001.

[26] E. Keogh, "Mining Shape and Time Series Databases with Symbolic Representations," in Tutorial of the 13rd ACM International Conference on Knowledge Discovery and Data mining (KDD 2007), 2007, pp. 12-15.

[27] E. Keogh and C. A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping," in VLDB '02 Proceedings of the 28th international conference on Very Large Data Bases, 2002, pp. 406-417.

[28] E. Keogh, K. Chakrabarti , M. Pazzani , S. Mehrotra , "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," in Proceedings of Conference on Knowledge and Information Systems, 2000, pp. 263-286.

[29] E. Keogh, K. Chakrabarti, S. Mehrotra, M. Pazzani, "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases," in Proceedings of ACM SIGMOD Conference on Management of Data, Santa Barbara, CA, 2001, pp. 151-162.

[30] I. .-B. Kang, "Multi-period forecasting using different models for different horizons: An application to U.S. economic time series data," *International Journal of Forecasting*, vol. 19, pp. 387-400, 2003.

[31] J. H. Kim, "Forecasting autoregressive time series with bias corrected parameter estimators," *International Journal of Forecasting*, vol. 19, pp. 493-502, 2003.

[32] K. J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, pp. 307-319, 2003.

[33] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "New Time Series Data Representation ESAX for Financial Applications," in Proceedings of the International Special Workshop on Databases for Next-Generation Researchers (SWOD 2006) in conjunction with International Conference on Data Engineering, ICDE 2006 , Georgia, USA, 2006, pp. 17-22.

[34] J. Lin, E. Keogh, S. Leonardi, B. Chiu, "A symbolic Representation of Time Series with Implications for Streaming Algorithms," in Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, CA, 2003, pp. 2-11.

[35] Q. Li, I. López, B. Moon, "Skyline Index for Time Series Data," in *IEEE Trans. on Knowledge and Data Engineering*, vol. 16, 2004, pp. 669-684.

[36] A. T. Lora, J. R. Santos, J. C. R. Santos, A. G. Expósito, J. L. M. Ramos, "Time series prediction: Application to the short term electric energy demand," in *Lecture Notes in Artificial Intelligence*, 2004, pp. 577-586.

[37] A.T. Lora, J.M.R. Santos, A.G. Exposito, J.L.M. Ramos, J.C.R. Santos, "Electricity Market Price Forecasting Based on Weighted Nearest Neighbors Techniques," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1294-1301, Aug. 2007.

[38] M. A. Mohandes, T. O. Halawani, S. Rehman,A. A. Hussain, "Support vector machine for wind speed prediction," *Renewable Energy*, vol. 29, pp. 938-947, 2004.

[39] R. Nayak and P. te Braak, "Temporal Pattern Matching for the Prediction of Stock Prices," in *Ong, K.-L. and Li, W. and Gao, J., Eds Proceedings 2nd International Workshop on Integrating Artificial Intelligence and Data Mining (AIDM 2007) CRPIT, 84*, Gold Coast, 2007, pp. 99-107.

[40] I. Popivanov, R. J. Miller, "Efficient Similarity Queries Over Time Series Data Using Wavelets," in Proceedings of the 18th International Conference on Data Engineering, San Jose, California, USA, 2002, pp. 212-221.

[41] A. K. Palit and D. Popovic, *Computational intelligence in time series forecasting – Theory and Engineering Applications*. Springer-Verlag London, 2005.

[42] A. Ratanamahatana, E. Keogh, A. J. Bagnall, S. Lonardi, "A Novel Bit Level Time Series Representation with Implications for Similarity Seach and Clustering," in Proc. 9th Pacific-Asian Int. Conf. on Knowledge Discovery and Data Mining (PAKDD'05), Hanoi, Vietnam, 2005, pp. 51-65.

[43] J. Shieh and E. Keogh, "iSAX: indexing and mining terabyte sized time series," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 623-631.

[44] Nguyen Thanh Son, Duong Tuan Anh, (2011)**.** Time Series Similarity Search based on Middle Points and Clipping. *Proceedings of the 3rd Conference on Data Mining and Optimization (DMO 2011), Putrajaya, Malaysia***,** *June 28-29, 2011,* IEEE, pp.13-19.

[45] A. Sorjamaa, J. Hao and A. Lendasse, "Mutual information and k-nearest neighbors approximator for time series prediction," in *Artificial Neural Networks: Biological Inspirations – ICANN 2005: 15th International Conference*, Warsaw, Poland, 2005, pp. 553-558.

[46] Spice-Neuro Neural Network Program. [Online]. http://www.spice.ci.ritsumei.ac.jp/~thang/programs

[47] Cao Duy Truong, Huynh Nguyen Tin, Duong Tuan Anh, 2013, Combining Motif Information and Neural Network for Time Series Prediction. *International Journal of Business Intelligence and Data Mining*, Vol. 7, No. 4, 2012, pp. 318-339.

[48] G. Tkacz, "Neural network forecasting of canadian GDP growth," *International Journal of Forecasting*, vol. 17, pp. 57-69, 2001.

[49] G. P. Zhang and M. Qi, "Neural Network Forecasting for Seasonal and Trend Time Series," *European Journal of Operational Research*, vol. 160, pp. 501-514, 2005.

[50] G. Zhang, B. E. Patuwo, M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, pp. 35-62, 1998.


TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
www.hcmute.edu.vn