

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN  
BỘ MÔN KHOA HỌC MÁY TÍNH**

----- ❁ -----

**DỰ ĐOÁN GIÁ VÀ XU HƯỚNG  
CHỨNG KHOÁN BẰNG CÁC PHƯƠNG  
PHÁP MÁY HỌC**

**ĐỒ ÁN MÔN HỌC MÁY HỌC**

**NHÓM THỰC HIỆN**

**0712133 – LÊ MINH DUY**

**0712183 – PHẠM MINH HOÀNG**

**0712228 – TRẦN TRUNG KIÊN**

**0712263 – VẠN DUY THANH LONG**

**0712394 – BÀNH TRÍ THÀNH**



**NĂM 2011**

# MỤC LỤC

|   |           |
|---|-----------|
| <b>MỤC LỤC .....</b>  | <b>2</b>  |
| <b>Chương 1 GIỚI THIỆU .....</b>                            | <b>5</b>  |
| 1.1 TẦM QUAN TRỌNG CỦA BÀI TOÁN DỰ ĐOÁN CHỨNG KHOÁN .....   | 5         |
| 1.1.1 Tổng quan.....  | 5         |
| 1.1.2 Mục tiêu .....  | 6         |
| 1.1.3 Hướng tiếp cận .....                                  | 6         |
| 1.2 PHÁT BIỂU VẤN ĐỀ.....                                   | 6         |
| 1.3 GIẢI QUYẾT VẤN ĐỀ .....                                 | 7         |
| 1.3.1 Phân tích cơ bản.....                                 | 7         |
| 1.3.2 Phân tích kỹ thuật .....                              | 7         |
| <b>Chương 2 CÁC VẤN ĐỀ CỦA HƯỚNG TIẾP CẬN MÁY HỌC .....</b> | <b>8</b>  |
| 2.1 CHỌN INPUT/ OUTPUT.....                                 | 8         |
| 2.2 TIỀN XỬ LÝ .....  | 8         |
| 2.3 CHỌN MÔ HÌNH MÁY HỌC .....                              | 9         |
| 2.4 ĐÁNH GIÁ MÔ HÌNH .....                                  | 9         |
| <b>Chương 3 NỀN TẢNG LÝ THUYẾT .....</b>                    | <b>12</b> |
| 3.1 ARTIFICIAL NEURAL NETWORK (ANN) .....                   | 12        |
| 3.1.1 Tổng quan.....  | 12        |
| 3.1.2 Mô hình ANN.....                                      | 12        |
| 3.2 SUPPORT VECTOR REGRESSION (SVR) .....                   | 16        |
| 3.2.1 Tổng quan.....  | 16        |
| 3.2.2 Mô hình SVR.....                                      | 16        |
| 3.2.3 Chọn các tham số cho mô hình.....                     | 21        |
| 3.2.3.1 Grid Search.....                                    | 21        |
| 3.2.3.2 Pattern Search .....                                | 22        |
| 3.2.3.3 Cross-Validation truyền thống.....                  | 24        |

|   |           |
|---|-----------|
| <b>Chương 4 CÁCH CẢI TIẾN</b>                           | <b>25</b> |
| 4.1 CẢI TIẾN CROSS-VALIDATION TRONG TÌM THAM SỐ CỦA SVR | 25        |
| 4.2 TĂNG ĐỘ CHÍNH XÁC VỀ HƯỚNG TRONG DỰ ĐOÁN GIÁ        | 26        |
| 4.3 Cải tiến cách tổ chức dữ liệu đưa vào mô hình:      | 27        |
| <b>Chương 5 CÀI ĐẶT</b>                                 | <b>30</b> |
| 5.1 SVR   | 30        |
| 5.1.1 Mô hình   | 30        |
| 5.1.2 Chọn tham số                                      | 30        |
| 5.1.2.1 Grid Search                                     | 30        |
| 5.1.2.2 Pattern Search                                  | 31        |
| 5.2 ANN   | 31        |
| 5.2.1 Sơ đồ tổng quan các chức năng trong mô hình       | 31        |
| 5.2.2 Training  | 31        |
| 5.2.2.1 Cài đặt Mạng                                    | 31        |
| 5.2.2.2 Tổ chức dữ liệu training:                       | 32        |
| 5.2.2.3 Lựa chọn hàm chi phí:                           | 32        |
| 5.2.2.4 Các tham số:                                    | 32        |
| 5.2.3 Quá trình training:                               | 33        |
| <b>Chương 6 THÍ NGHIỆM</b>                              | <b>34</b> |
| 6.1 DỰ ĐOÁN GIÁ   | 34        |
| 6.1.1 Mô tả dữ liệu                                     | 34        |
| 6.1.2 Thí nghiệm  | 34        |
| 6.1.3 Kết quả   | 35        |
| 6.1.4 Nhận xét  | 36        |
| 6.2 DỰ ĐOÁN XU HƯỚNG                                    | 37        |
| 6.2.1 Mô tả dữ liệu                                     | 37        |
| 6.2.2 Thí nghiệm  | 38        |
| 6.2.2.1 Với mã nước ngoài                               | 38        |
| 6.2.2.2 Với mã trong nước                               | 40        |
| 6.3 Tham số của ANN                                     | 40        |
| 6.3.1 Nhận xét  | 46        |

|   |  |           |
|---|--|-----------|
| <b>Chương 7</b>                         | <b>KẾT LUẬN VÀ CÔNG VIỆC TƯƠNG LAI .....</b> | <b>47</b> |
| 7.1                                     | KẾT LUẬN.....                                | 47        |
| 7.1.1                                   | Dự đoán giá .....                            | 47        |
| 7.1.2                                   | Dự đoán xu hướng .....                       | 47        |
| 7.1.3                                   | Toàn cảnh .....                              | 48        |
| 7.2                                     | Công việc tương lai .....                    | 49        |
| <b>PHỤ LỤC. HƯỚNG DẪN SỬ DỤNG .....</b> | <b>51</b>                                    |           |
| 7.3                                     | PHẦN THỰC NGHIỆM (EXPERIMENT) .....          | 51        |
| 7.4                                     | PHẦN ỨNG DỤNG (APPLICATION).....             | 57        |
| <b>TÀI LIỆU THAM KHẢO .....</b>         | <b>58</b>                                    |           |

### 1.1 TẦM QUAN TRỌNG CỦA BÀI TOÁN DỰ ĐOÁN CHỨNG KHOÁN

#### 1.1.1 *Tổng quan*

Ngày nay, sự phát triển mạnh mẽ của của các thị trường chứng khoán đã đem lại lợi nhuận to lớn cho nhiều nhà đầu tư. Việc nhận định đúng xu hướng của thị trường là một vấn đề cốt yếu đem lại thu nhập khổng lồ cho họ từ những khoản đầu tư nhỏ. Thế nhưng, thị trường chứng khoán tuy không thay đổi ngẫu nhiên nhưng để xác định đúng xu hướng giá hiện tại hoặc những điểm thay đổi xu hướng là một quá trình trải qua kinh nghiệm lâu dài đối với bất kỳ nhà đầu tư nào. Vì vậy, yêu cầu đặt ra là nhà đầu tư cần sự hỗ trợ tốt nhất cho những quyết định với việc giải quyết hiệu quả bài toán dự đoán xu hướng và giá chứng khoán

Không có nhiều phương pháp tốt được áp dụng cho thị trường chứng khoán Việt Nam. Một nguyên nhân đầu tiên có thể dễ dàng nhận thấy đó là sự non trẻ của thị trường dẫn đến việc một số nghiên cứu áp dụng một vài vấn đề lý thuyết trong chứng khoán vào thị trường Việt Nam chưa đạt nhiều hiệu quả. Với thời gian hoạt động 10 năm (từ năm 2000), thị trường nước ta chỉ mới nhận được sự quan tâm tin học hóa trong thời gian gần đây và phần lớn ở dạng cổng thông tin trực tuyến. Bên cạnh đó, các phương pháp dự đoán xu hướng và giá chứng khoán cũng như hỗ trợ ra quyết định chỉ dừng lại ở việc sử dụng trực tiếp và đơn thuần các chỉ số kỹ thuật. Điều này làm cho các phương pháp đó mang lại hiệu quả không cao.

Trên thế giới, ở các thị trường phát triển (như Mỹ, Anh, Hong Kong,...) đã có nhiều nghiên cứu và các hệ thống triển khai đạt được kết quả rất tốt. Các nghiên cứu này phần lớn áp dụng một vài phương pháp máy học cho quá trình

dự đoán. Phương pháp được dùng phổ biến nhất là sử dụng mạng nơ-ron nhân tạo (ANN). Gần đây, với một số kết quả khả quan đạt được, phương pháp sử dụng support vector machine cho bài toán hồi quy (SVR) cũng được coi là một hướng tiếp cận tối ưu.

### **1.1.2 Mục tiêu**

Với những vấn đề trên, mục tiêu của đề tài là áp dụng các phương pháp máy học vào quá trình dự đoán xu hướng và giá của chứng khoán để hỗ trợ cho quá trình ra quyết định của nhà đầu tư. Việc dự đoán giá chính xác trong chu kỳ ngắn sẽ hỗ trợ cho quá trình đặt lệnh của nhà đầu tư hiệu quả hơn. Trong khi đó, việc dự đoán xu hướng sẽ giúp nhà đầu tư quyết định khi nào cần bán hay mua chứng khoán.

### **1.1.3 Hướng tiếp cận**

Cụ thể, nhóm sẽ áp dụng ANN và SVR vào bài toán dự đoán dựa trên dữ liệu đầu vào là các thông số trên bảng điện tử của thị trường chứng khoán. Với từng phương pháp, nhóm cài đặt mô hình cho cả hai bài toán dự đoán xu hướng và dự đoán giá chính xác sau đó so sánh và đánh giá hiệu quả mà từng phương pháp mang lại.

## **1.2 PHÁT BIỂU VẤN ĐỀ**

Ở đây, nhóm chia vấn đề dự đoán chứng khoán thành 2 vấn đề con:

- *Dự đoán giá*: mục tiêu của ta là dự đoán giá của một số ngày tiếp theo (**ngắn hạn**). Cái ta hướng đến là dự đoán **vừa đúng về giá trị vừa đúng về xu hướng**(tăng bao nhiêu, giảm bao nhiêu.) Output là giá dự đoán.
- *Dự đoán xu hướng*: mục tiêu của ta là dự đoán xu hướng xét trong 1 khoảng thời gian nào đó, mà thường là **trung hạn và dài hạn**. Cái ta hướng tới là dự đoán **đúng xu hướng** (tăng hay giảm, không quan tâm là tăng bao nhiêu hay giảm bao nhiêu.) Output là một trong hai

giá trị ứng với 2 trường hợp: xu hướng tăng, xu hướng giảm (Ta cũng có thể đưa thêm trường hợp thứ 3: xu hướng không đổi.)

## 1.3 GIẢI QUYẾT VẤN ĐỀ

Một cách chung nhất, có hai hướng tiếp cận đối với bài toán dự đoán chứng khoán:

- Phân tích cơ bản.
- Phân tích kỹ thuật.

### 1.3.1 *Phân tích cơ bản*

- Phân tích bản cân đối tài khoản và bản báo cáo lợi tức của công ty để tìm ra giá trị nội tại (tình hình phát triển của công ty) đó.
- Phân tích cơ bản cho rằng trong ngắn hạn, thị trường có thể đánh giá sai về giá trị nội tại của cổ phiếu của công ty đó nhưng về dài hạn, giá cổ phiếu sẽ hội tụ về giá trị nội tại.
- Như vậy, ta có thể thu lợi nhuận bằng cách mua cổ phiếu khi bị thị trường đánh giá thấp giá trị nội tại và bán ra khi thị trường đánh giá đúng giá trị nội tại.

### 1.3.2 *Phân tích kỹ thuật*

- Phân tích kỹ thuật cho rằng tất cả các thông tin về công ty đều được phản ánh qua giá cả → Phân tích cơ bản chỉ mất công.
- Cách thức: kết hợp dữ liệu quá khứ với các mô hình sóng.

Gần đây, cộng đồng phân tích kỹ thuật có xu hướng sử dụng các mô hình máy học như là ANN, SVM, Gaussian Process, Hidden Markov Model, các mô hình lai, ...

Ở đây, nhóm giải quyết vấn đề theo hướng phân tích kỹ thuật mà cụ thể là sử dụng 2 mô hình máy học được cho là mạnh nhất hiện nay trong lĩnh vực dự đoán chứng khoán: ANN (Artificial Neural Network) và SVR (Support Vector Regression.)

## Chương 2      CÁC VẤN ĐỀ CỦA HƯỚNG TIẾP CẬN MÁY HỌC

Để giải được bài toán khó này, trước hết ta cần phải nhận ra được các vấn đề cần phải giải quyết của nó. Với hướng tiếp cận sử dụng các mô hình máy học, ta có 4 vấn đề cơ bản như sau:

### 2.1 CHỌN INPUT/ OUTPUT

Đây là bước rất quan trọng. Ta phải chọn input/ output sao cho nó cung cấp đủ thông tin để mô hình của ta có thể nhận ra được các mẫu tiềm ẩn. Nhưng nếu ta chọn quá nhiều input thì cũng có thể gây ra nhiễu, làm giảm khả năng dự đoán của mô hình.

Nhìn chung, các input có thể sử dụng gồm có: Giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa, các indicator.

Ở đây, để khởi đầu, nhóm chọn input chỉ là giá đóng cửa.

### 2.2 TIỀN XỬ LÝ

Nhìn chung, bước tiền xử lý thực hiện hai nhiệm vụ: khử nhiễu và chuẩn hóa.

Ở đây, với input chỉ là giá đóng cửa, ở bước tiền xử lý nhóm chỉ thực hiện nhiệm vụ chuẩn hóa.

- Dự đoán giá: đơn giản là scale về  $[0, 1]$  theo công thức:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- Dự đoán xu hướng: scale về dạng return theo công thức

$$y_t = 100 \times (\log I_{t+1} - \log I_t)$$

Với  $I_t$  là giá thời điểm  $t$



## 2.3 CHỌN MÔ HÌNH MÁY HỌC

Nhìn chung, các mô hình máy học được xây dựng lên dựa trên các giả sử nào đó về sự phân bố của dữ liệu. Một mô hình có thể phù hợp ở ứng dụng này nhưng khi đưa qua ứng dụng khác thì chưa chắc. Vì vậy, cách duy nhất để chọn được mô hình máy học phù hợp là thử với các mô hình máy học khác nhau.

Ở đây, nhóm khởi đầu với hai mô hình được coi là mạnh nhất hiện nay trong lĩnh vực dự đoán chứng khoán: ANN và SVR.

Khi đã chọn được mô hình máy học rồi thì lại nảy sinh một vấn đề con là **chọn các tham số cho mô hình**.

## 2.4 ĐÁNH GIÁ MÔ HÌNH

Ở đây, ta có thể chia các độ đo để đánh giá mô hình thành 2 nhóm:

- Nhóm các độ đo về giá trị: càng nhỏ càng tốt.

- MSE (Mean Square Error):

$$MSE = \frac{1}{N} \sum_{n=1}^N (y'_n - y_n)^2$$

Với  $y'_n$  và  $y_n$  lần lượt là giá trị dự đoán và giá trị thực.

- MAE (Mean Absolute Error):

$$MAE = \frac{1}{N} \sum_{n=1}^N |y'_n - y_n|$$

- MAPE (Mean Absolute Percentage Error):

$$MAPE = \frac{100}{N} \sum_{n=1}^N \left| \frac{y'_n - y_n}{y_n} \right|$$

- NMSE (Normalized Mean Square Error):

$$NMSE = 100 \times \frac{MSE}{var(y)} = 100 \times \frac{\frac{1}{N} \sum_{n=1}^N (y'_n - y_n)^2}{\frac{1}{N-1} \sum_{n=1}^N (y_n - \bar{y})^2}$$

Với  $\bar{y}$  là giá trị trung bình của  $y_1, \dots, y_n$

- Nhóm các độ đo về hướng:

- DS (Directional Symmetry): càng lớn càng tốt.

$$DS = \frac{100}{N} \sum_{n=1}^N d_n$$

Với:

$$d_n = \begin{cases} 1 & \text{nếu } (y_n - y_{n-1})(y'_n - y'_{n-1}) \geq 0 \\ 0 & \text{nếu ngược lại} \end{cases}$$

- Sign:

$$sign(\%) = \frac{100}{N} \times \sum_{n=1}^N \alpha_n \text{ vi } \alpha_n = \begin{cases} 1 & \text{nếu } y_n y'_n > 0 \\ 0 & \text{nếu ngược lại} \end{cases}$$

- DM4Price (Direction Measure For Price): đây là độ đo do nhóm tự đặt ra cho bài toán dự đoán giá. Nó xuất phát từ hai lý do:

- Như đã nói, mục tiêu của ta trong bài toán dự đoán giá là dự đoán đúng cả về giá trị lẫn xu hướng. Giá trị thì ta đã có rất nhiều độ đo, nhưng xu hướng thì chỉ có độ đo DS. Tuy nhiên, việc xét dấu của tích  $(y_n - y_{n-1})(y'_n - y'_{n-1})$  không cho ta nhiều thông tin lắm. Ở đây, ta cần phải xét dấu của tích  $(y_n - y_{n-1})(y'_n - y'_{n-1})$
- Các cổ phiếu Việt Nam có đặc điểm là giá thường không đổi, dẫn đến tích trên = 0. Nếu ta cho  $d_n = 1$  khi tích  $\geq 0$  như ở độ đo DS thì với các cổ phiếu Việt Nam sẽ cho DS rất cao, trong khi điều đó không nói lên rằng mô hình của ta là tốt.

Từ đây, ta có công thức DM4Price như sau:

Xét tích  $(y_n - y_{n-1})(y'_n - y'_{n-1})$

- Nếu tích  $> 0$ : dự đoán đúng.
- Nếu tích  $< 0$ : dự đoán sai.
- Nếu tích  $= 0$ : không đánh giá được.

Độ đo DM4Price sẽ cho ra 3 kết quả lần lượt ứng với phần trăm số phần tử dự đoán đúng, dự đoán sai và không đánh giá được.

## CHƯƠNG 3 NỀN TẢNG LÝ THUYẾT

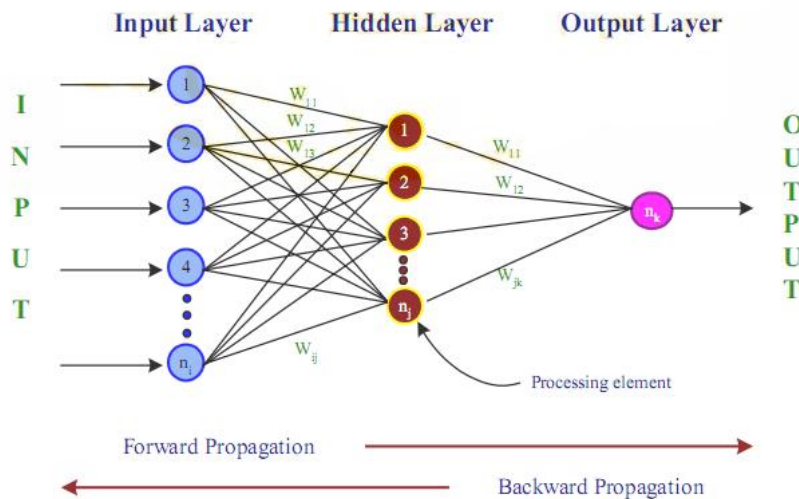
### 3.1 ARTIFICIAL NEURAL NETWORK (ANN)

#### 3.1.1 Tổng quan

Ý tưởng của ANN là mô phỏng một mạng neural nhân tạo có thể hoạt động giống với mạng neural sinh học bằng cách kết hợp tuyến tính các hàm truyền phi tuyến lại với nhau để xây dựng mô hình hồi quy.

Mục đích của ANN là tối thiểu hóa hàm độ lỗi thông qua việc xác định một bộ tham số phù hợp cho các hàm truyền cơ bản.

#### 3.1.2 Mô hình ANN



Mô hình ANN có thể xem như là một hàm phi tuyến nhận vào một tập các giá trị đầu vào  $\{x_i\}$  và trả ra tập các giá trị  $\{y_k\}$  thông qua sự chi phối của bộ trọng số  $w$ .

$$y_k(w, x) = g \left( \sum_{j=1}^M w_{kj}^{(2)} f \left( \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)} \right)$$

Để đơn giản hóa công thức, ta giả sử có thêm biến đầu vào  $x_o = 1$ , chúng ta sẽ có công thức tương đương như sau:

$$y_k(w, x) = g \left( \sum_{j=0}^M w_{kj}^{(2)} f \left( \sum_{i=0}^D w_{ji}^{(1)} x_i \right) \right)$$

Hướng tiếp cận của vấn đề là ta phải tối thiểu hóa hàm độ lỗi sum-of-squares thông qua việc xác định bộ tham số cho mạng. Cho một bộ huấn luyện với tập các giá trị đầu vào và nhãn của nó là  $\{x_n, t_n\}$ , ta có hàm độ lỗi là:

$$E(w) = \frac{1}{2} \sum_{n=1}^N \|y_n(x_n, w) - t_n\|^2$$

Ta thấy rằng do  $y(x_n, w)$  là một hàm phi tuyến nên  $E(w)$  không thể là bao lồi (nonconvex) do đó một khuyết điểm của ANN là dễ bị mắc vào các giá trị tối ưu cục bộ.

Vì hàm lỗi  $E(w)$  là một hàm liên tục theo biến  $w$ , nên các giá trị cực tiểu của nó sẽ xuất ở điểm mà có đạo hàm bằng 0:  $\nabla E(w) = 0$ .

Tuy nhiên, như đã nói hàm độ lỗi phụ thuộc phi tuyến vào bộ trọng số  $w$ , nên sẽ có rất nhiều điểm trong không gian trọng số có giá trị đạo hàm bằng 0 (các cực tiểu cục bộ). Việc tối ưu hóa phải hướng đến tìm giá trị nhỏ nhất của hàm độ lỗi, hay nói cách khác chính là phải tìm được cực tiểu toàn cục.

Không có hy vọng trong cách giải quyết vấn đề tìm cực tiểu toàn cục với  $\nabla E(w) = 0$ . Một vài thuật toán khác tập trung vào việc lựa chọn giá trị khởi tạo  $w^{(0)}$  và việc cập nhật vector trọng số  $\nabla w^{(r)}$

$$w^{(r+1)} = w^{(r)} + \nabla w^{(r)}$$

Trong đó thuật toán Gradient descent được áp dụng phổ biến trong việc huấn luyện mô hình ANN với bộ dữ liệu lớn. Ta có hàm độ lỗi cho một tập các biến quan sát được độc lập nhau là:

$$E(w) = \sum_{n=1}^N E_n(w)$$

Ý tưởng của thuật toán Gradient descent là từ việc khởi tạo một giá trị ngẫu nhiên cho bộ trọng số  $w$ , qua việc chạy thuật toán nhiều lần ta sẽ tiến hành cập nhật dần bộ trọng số cho đến khi hội tụ, các lần chạy sau cho kết quả tối ưu hơn các kết quả trước đó.

$$w^{(r+1)} = w^{(r)} - \eta \nabla E_n(w^{(r)})$$

Ta sẽ thấy được rằng hệ số  $\eta$  nhỏ sẽ làm thuật toán lâu hội tụ, và nếu nó quá lớn thì hàm lỗi bị xoay quanh các giá trị biên của điểm hội tụ.

Tóm lại mô hình ANN sẽ qua các bước xử lý sau:

- Cho vector input  $I = (i_1, i_2, \dots, i_n)$  được truyền vào lớp input của mạng. Sau quá trình tiền xử lý (khử nhiễu, chuẩn hóa), mỗi node ở lớp input sẽ tạo ra vector  $O^{(i)} = (o_1, o_2, \dots, o_n)$  truyền đến lớp ẩn.
- Giá trị đầu vào  $i_j^{(h)}$  của neuron thứ  $j$  của lớp ẩn ( $h$ ) sẽ được tính như sau:

$$i_j^{(h)} = b + \sum_{i=1}^N w_{ij}^{(h)} o_i^{(i)}$$

- Giá trị output của neuron thứ  $j$  trong lớp ẩn là  $o_j^{(h)}$  được tính theo công thức:

$$o_j^{(h)} = f_j(net)$$

Ở đây  $net$  chính là  $i_j^{(h)}$  và  $f_j(net)$  được gọi là hàm truyền phi tuyến.

$$f_j(net) = \frac{1}{1 - e^{-net}}$$

- Giá trị đầu vào  $i_k^{(o)}$  của neuron thứ k trong lớp output (o) được tính theo công thức:

$$i_k^{(o)} = c + \sum_{j=1}^M w_{jk}^{(o)} o_j^{(h)}$$

- Giá trị đầu ra của neuron thứ k trong lớp output là  $o_k^{(o)}$  được tính theo công thức:

$$o_k^{(k)} = g_k(net)$$

Ở đây net chính là  $i_k^{(o)}$  và  $g_k(net)$  là hàm truyền phi tuyến cho lớp output.

$$g_k(net) = net$$

- Tiếp theo khi đã có kết quả ở lớp output, ta cần tính độ lỗi  $\delta_k^{(o)}$  giữa giá trị dự đoán với giá trị thực để lan truyền ngược lại cập nhật trọng số theo công thức:

$$\delta_k^{(o)} = (d_k - o_k^{(o)}) \cdot g'_k(net)$$

Ở đây  $d_k$  là giá trị thực và  $g'_k(net)$  là đạo hàm của hàm truyền phi tuyến ở lớp xuất.

$$g'_k(net) = 1$$

- Ta tính tiếp độ lỗi ở các neuron lớp ẩn  $\delta_j^{(h)}$  theo công thức:

$$\delta_j^{(h)} = f'_j(net) \cdot \sum_{k=1}^P \delta_k^{(o)} w_{jk}^{(o)}$$

Ở đây  $f'_j(net)$  là đạo hàm của hàm truyền phi tuyến ở lớp ẩn.

$$f'_j(net) = f_j(net) \cdot [1 - f_j(net)]$$

- Độ thay đổi trọng số ở lớp output  $\Delta w_{jk}^{(o)}$  và ở lớp ẩn  $\Delta w_{ij}^{(h)}$  ở vòng lặp t được tính như sau:

$$\Delta w_{jk}^{(o)}(t) = \eta \delta_k^{(o)} o_j^{(h)} + \alpha [\Delta w_{jk}^{(o)}(t-1)]$$

$$\Delta w_{ij}^{(h)}(t) = \eta \delta_j^{(h)} o_i^{(i)} + \alpha [\Delta w_{ij}^{(h)}(t-1)]$$

Ở đây  $\eta$  là hệ số học và  $\alpha$  là nhân tố hướng để hạn chế bị rơi vào bẫy tối ưu cục bộ.

- Trọng số mới được cập nhật theo công thức sau:

$$w_{jk}^{(o,new)} = w_{jk}^{(o)} + \Delta w_{jk}^{(o)}(t)$$

$$w_{ij}^{(h,new)} = w_{ij}^{(h)} + \Delta w_{ij}^{(h)}(t)$$

## 3.2 SUPPORT VECTOR REGRESSION (SVR)

### 3.2.1 Tổng quan

Ý tưởng cơ bản của SVR là ta sẽ ánh xạ không gian đầu vào (mà nếu ta áp dụng hồi qui tuyến tính thì không hiệu quả) sang một không gian mới cao chiều hơn mà ở đó, ta có thể áp dụng được hồi qui tuyến tính.

Đặc điểm của SVR là cho ta một giải pháp thưa; nghĩa là để xây dựng được hàm hồi qui, ta không cần phải sử dụng hết tất cả các điểm dữ liệu trong bộ huấn luyện. Những điểm có đóng góp vào việc xây dựng hàm hồi qui được gọi là những Support Vector.

Điểm mạnh của SVR là sử dụng tối ưu hóa rủi ro cấu trúc (structural risk minimization), nhờ đó mà khả năng tổng quát hóa cao, tránh overfit (ANN thì dễ bị overfit.) Hơn nữa, hàm mục tiêu của SVR là hàm “convex”, do đó điểm cực trị tìm được sẽ là cực trị toàn cục (hàm mục tiêu của ANN thì không như vậy, nó có nhiều điểm cực trị và dẫn đến cực trị tìm được thường là cực trị cục bộ.)

### 3.2.2 Mô hình SVR

Với bài toán hồi qui tuyến tính đơn giản, ta phải minimize hàm lỗi chuẩn hóa:

$$\frac{1}{2} \sum_{n=1}^N \{y_n - t_n\}^2 + \frac{\lambda}{2} \|w\|^2$$

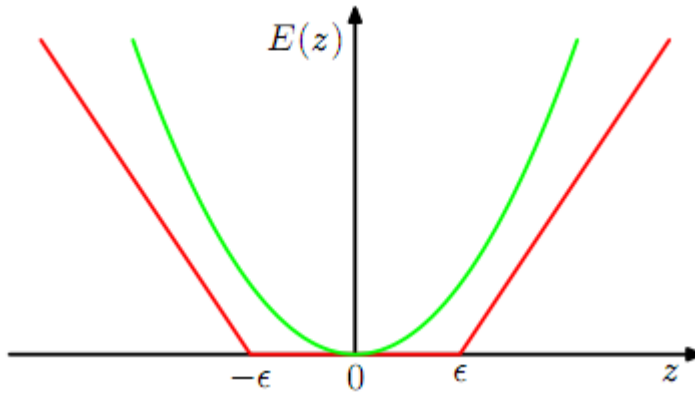
Để có được một giải pháp thưa, ta sẽ thay hàm lỗi trên bằng hàm lỗi  $\epsilon$ -insensitive. Đặc điểm của hàm lỗi này là nếu trị tuyệt đối của sự sai khác giữa



giá trị dự đoán  $y(x)$  và giá trị đích nhỏ hơn epsilon (với  $\epsilon > 0$ ) thì nó coi như là độ lỗi bằng 0.

$$E_{\epsilon}(y(x) - t) = \begin{cases} 0 & \text{nếu } |y(x) - t| < \epsilon \\ |y(x) - t| - \epsilon & \text{nếu ngược lại} \end{cases}$$

Để hiểu thêm, ta hãy ngó qua hình vẽ dưới đây:



Trong đó, đường màu xanh là hàm lỗi bậc hai thông thường; đường màu đỏ là hàm lỗi  $\epsilon$ -insensitive.

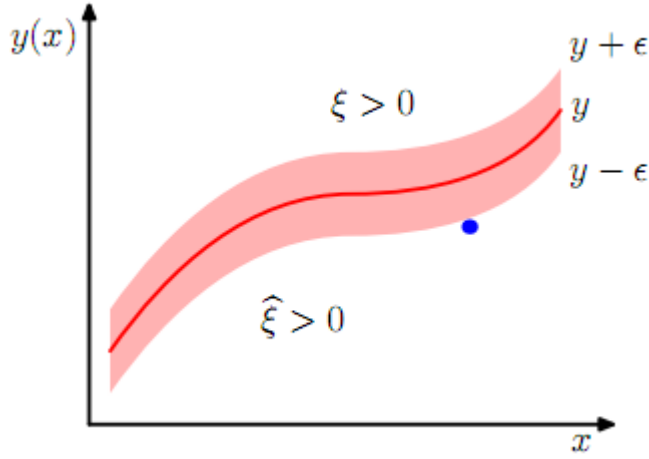
Như vậy bây giờ, ta phải minimize hàm lỗi chuẩn hóa sau:

$$C \sum_{n=1}^N E_{\epsilon}(y(x_n) - t_n) + \frac{1}{2} \|w\|^2$$

Với:

$$y(x_n) = w^T \phi(x_n) + b$$

Để cho phép một số điểm nằm ngoài ống epsilon, ta sẽ đưa thêm các biến “slack” vào. Đối với mỗi điểm dữ liệu  $x_n$ , ta cần hai biến slack  $\xi_n \geq 0$  và  $\widehat{\xi}_n \geq 0$ ; trong đó  $\xi_n > 0$  ứng với điểm mà  $t_n > y(x_n) + \epsilon$  (nằm ngoài và phía trên ống) và  $\widehat{\xi}_n > 0$  ứng với điểm mà  $t_n < y(x_n) - \epsilon$  (nằm ngoài và phía dưới ống.)



Điều kiện để một điểm đích nằm trong ống là:  $y_n - \epsilon \leq t_n \leq y_n + \epsilon$  với  $y_n = y(x_n)$ . Với việc sử dụng các biến slack, ta cho phép các các điểm đích nằm ngoài ống (ứng với các biến slack  $> 0$ ) và như thế thì điều kiện của ta bây giờ sẽ là:

$$t_n \leq y_n + \epsilon + \xi_n \quad [4.2.2.1]$$

$$t_n \geq y_n - \epsilon - \widehat{\xi}_n \quad [4.2.2.2]$$

Như vậy, ta có hàm lỗi cho SVR:

$$C \sum_{n=1}^N (\xi_n + \widehat{\xi}_n) + \frac{1}{2} \|w\|^2 \quad [4.2.2.3]$$

Mục tiêu của ta là minimize hàm lỗi này với các ràng buộc:

- $\xi_n \geq 0, \widehat{\xi}_n \geq 0$
- [4.2.2.1]
- [4.2.2.2]

Ta hãy gọi đây là vấn đề tối ưu hóa A.

Có ngay hàm Lagrange:

$$L = C \sum_{n=1}^N (\xi_n + \widehat{\xi}_n) + \frac{1}{2} \|w\|^2 - \sum_{n=1}^N (\mu_n \xi_n + \widehat{\mu}_n \widehat{\xi}_n) - \sum_{n=1}^N a_n (\epsilon + \xi_n + y_n - t_n) - \sum_{n=1}^N \widehat{a}_n (\epsilon + \widehat{\xi}_n - y_n + t_n) \quad [4.2.2.4]$$

Với

$$y_n = y(x_n) = w^T \phi(x_n) + b$$

Lấy đạo hàm theo  $w$ ,  $b$ ,  $\xi_n$ ,  $\widehat{\xi_n}$  và cho bằng 0, ta được:

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{n=1}^N (a_n - \widehat{a_n}) \phi(x_n) \quad [4.2.2.5]$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{n=1}^N (a_n - \widehat{a_n}) = 0 \quad [4.2.2.6]$$

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow a_n + \mu_n = C \quad [4.2.2.7]$$

$$\frac{\partial L}{\partial \widehat{\xi_n}} = 0 \Rightarrow \widehat{a_n} + \widehat{\mu_n} = C \quad [4.2.2.8]$$

Dùng 4 kết quả này thế vào hàm Lagrange, ta sẽ loại bỏ được  $w$ ,  $b$ ,  $\xi_n$ ,  $\widehat{\xi_n}$ ,  $\mu_n$ ,  $\widehat{\mu_n}$ :

$$\tilde{L}(a, \widehat{a}) = -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N (a_n - \widehat{a_n})(a_m - \widehat{a_m}) k(x_n, x_m) - \epsilon \sum_{n=1}^N (a_n + \widehat{a_n}) \quad [4.2.2.9]$$

Với  $k$  là hàm kernel:  $k(x, x') = \phi(x)^T \phi(x')$

Như vậy, ta đã chuyển từ vấn đề tối ưu hóa A sang vấn đề tối ưu hóa B:

Maximize [4.2.2.9] với các ràng buộc:

- $0 \leq a_n \leq C$
- $0 \leq \widehat{a_n} \leq C$
- [4.2.2.6]

Lợi ích chính của việc chuyển đổi từ vấn đề tối ưu hóa A sang vấn đề tối ưu hóa B là vấn đề tối ưu hóa B có sử dụng hàm kernel. Điều này sẽ giúp cho việc tính toán trong không gian cao chiều trở nên rất hiệu quả.

Thế [4.2.2.5] vào hàm hồi qui ban đầu, ta có sự dự đoán cho một mẫu mới  $x$ :

$$y(x) = \sum_{n=1}^N (a_n - \widehat{a}_n) k(x, x_n) + b \quad [4.2.2.10]$$

Theo điều kiện KKT, có ngay:

$$a_n(\epsilon + \xi_n + y_n - t_n) = 0 \quad [4.2.2.11]$$

$$\widehat{a}_n(\epsilon + \widehat{\xi}_n - y_n + t_n) = 0 \quad [4.2.2.12]$$

$$(C - a_n)\xi_n = 0 \quad [4.2.2.13]$$

$$(C - \widehat{a}_n)\widehat{\xi}_n = 0 \quad [4.2.2.14]$$

Từ đây, ta có thể rút được những thông tin quan trọng như sau:

- Nếu  $a_n > 0$  thì  $\epsilon + \xi_n + y_n - t_n = 0$ : điểm nằm ở biên trên của ống ( $\xi_n = 0$ ) hoặc nằm ngoài về phía trên của ống ( $\xi_n > 0$ )
- Nếu  $\widehat{a}_n > 0$  thì  $\epsilon + \widehat{\xi}_n - y_n + t_n = 0$ : điểm nằm ở biên dưới của ống ( $\widehat{\xi}_n = 0$ ) hoặc nằm ngoài về phía dưới của ống ( $\widehat{\xi}_n > 0$ )
- $a_n$  và  $\widehat{a}_n$  không thể cùng dương vì nếu vậy thì ta có:  $\epsilon + \xi_n + y_n - t_n = 0$  và  $\epsilon + \widehat{\xi}_n - y_n + t_n = 0$ , cộng lại ta sẽ thấy ngay vế trái luôn dương, trong khi vế phải bằng 0: vô lý!
- Những điểm Support Vector là những điểm đóng góp vào hàm dự đoán [4.2.2.10], nghĩa là những điểm có  $a_n > 0$  hoặc  $\widehat{a}_n > 0$ : những điểm nằm trên biên ống hoặc nằm ngoài ống.
- Những điểm nằm trong ống sẽ có  $a_n = \widehat{a}_n = 0$  và do đó không đóng góp gì vào quá trình dự đoán.

*Tính b:*

Thấy ngay ta dễ tính được b bằng cách xét một điểm  $x_n$  có  $0 < a_n < C$ . Từ [4.2.2.13] ta có  $\xi_n = 0$ . Từ [4.2.2.11] ta có  $\epsilon + y_n - t_n = 0$ . Kết hợp với [4.2.2.10] có ngay:

$$b = t_n - \epsilon - \sum_{m=1}^N (a_m - \widehat{a}_m) k(x_n, x_m)$$

Ta cũng sẽ được kết quả tương tự nếu xét điểm có  $0 < \widehat{a}_n < C$ .

Để vững chắc hơn, ta nên lấy trung bình của tất cả các giá trị của b lại.

### 3.2.3 Chọn các tham số cho mô hình

#### 3.2.3.1 Grid Search

##### 3.2.3.1.1 Sơ bộ về ý tưởng

Grid Search chẳng qua là vét cạn không gian tìm kiếm. Đầu tiên, ta cần phải có miền giá trị của các biến cần tìm kiếm. Chẳng hạn, trong không gian hai chiều, đã biết  $x \in [2^2, 2^{10}]$ ,  $y \in [2^{-5}, 2^5]$ . Kế đến, ta cần phải chọn một hệ số Delta nào đó để rời rạc hóa các miền giá trị này. Chẳng hạn,  $\Delta = 2$ , ta sẽ xét  $x = 2^2, 2^4, 2^6, \dots, 2^{10}$  và  $y = 2^{-5}, 2^{-3}, 2^{-1}, \dots, 2^5$ . Sau đó, ta sẽ xét hết tất cả các trường hợp có thể (ở đây là  $5 \times 6$  trường hợp) để chọn ra cặp biến tốt nhất.

Ta nhận thấy hệ số rời rạc hóa Delta càng lớn thì sẽ làm giảm thời gian tính toán, nhưng ngược lại, độ chính xác của nó cũng sẽ giảm đi.

Grid Search là phương pháp truyền thống dùng để chọn bộ tham số cho SVM. Từ nhược điểm cố hữu của nó là quá tốn chi phí, người ta mới đề ra cải tiến như sau:

- Đầu tiên, ta sẽ dùng một grid thưa (Delta lớn) để chọn ra bộ tham số tốt nhất.
- Sau đó, ta sẽ làm kỹ hơn: ta sẽ dùng một grid dày hơn (Delta nhỏ hơn) xung quanh bộ tham số vừa tìm được ở trên.

##### 3.2.3.1.2 Thuật toán

**Bước 1: Khởi tạo**

- Giá trị bắt đầu của C, gamma, epsilon.
- Số interval (số này thể hiện số điểm grid), chẳng hạn nếu chọn 7 thì sẽ có  $7 \times 7 \times 7$  điểm cần search.
- Delta

**Bước 2: Search thưa**

- Ứng với từng bộ ba C, gamma, epsilon, thực hiện cross validation để tìm giá trị lỗi

- Cập nhật bộ ba mới nếu độ lỗi thấp hơn độ lỗi nhỏ nhất hiện tại.
- Tăng giá trị của từng tham số theo delta
- Quay lại bước 1 nếu chưa hết interval của cả 3 tham số.

**Bước 3: Thay đổi giá trị khởi tạo**

Để chuẩn bị cho việc search kỹ:

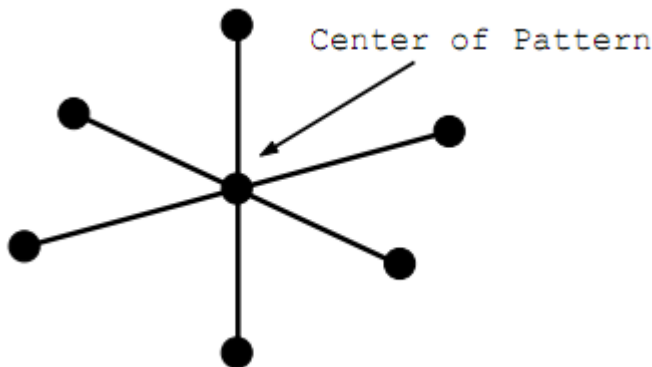
- Gán lại giá trị bắt đầu bằng giá trị lân cận của bộ ba có độ lỗi nhỏ nhất
- Giảm delta

**Bước 4: Thực hiện search kỹ với các công đoạn như bước 2.**

### 3.2.3.2 Pattern Search

#### 3.2.3.2.1 Sơ bộ về ý tưởng

Đầu tiên, ta cần hiểu thế nào là pattern? Pattern đơn giản chỉ là một nhóm các điểm kề cận nhau trong không gian, trong đó có một điểm ở giữa được gọi là trung tâm của pattern.



Ta sẽ bắt đầu thuật toán từ một điểm bất kỳ trong không gian tìm kiếm. Điểm này được coi là trung tâm của pattern và là điểm “tốt nhất” cho đến thời điểm hiện tại.

Từ điểm trung tâm này, ta sẽ tính giá trị của hàm mục tiêu tại các điểm kề cận nhằm tìm ra được một điểm tốt hơn điểm hiện tại. Các điểm kề cận được với điểm trung tâm được định bởi **hướng** (luôn không đổi) và một thành phần để xác định **khoảng cách** gọi là bước tìm kiếm (search step.)

Nếu tìm được, ta sẽ gán lại trung tâm của pattern là điểm tốt hơn đó. Ngược lại, ta sẽ giảm bước tìm kiếm và lặp lại quá trình trên. Thuật toán sẽ ngừng khi bước tìm kiếm nhỏ hơn một ngưỡng nào đó.

### 3.2.3.2.2 Thuật toán

#### Bước 1: Khởi tạo

- $k = 1$
- Chọn bước tìm kiếm ban đầu  $\Delta_k$  và ngưỡng của bước tìm kiếm để dừng thuật toán  $\tau$ .
- Chọn ngẫu nhiên điểm trung tâm  $x_k$  của pattern, tính  $f(x_k)$  và gán  $\min = f(x_k)$ .
- $\text{BestPoint} = x_k$ .

$$s_k = 0.$$

#### Bước 2: Nếu $\Delta_k \leq \tau$ , dừng thuật toán!

#### Bước 3: For $i = 1, \dots, n$ // $n$ là số chiều của không gian tìm kiếm

- $s_k^i = s_k + \Delta_k e_i$  và  $x_k^i = x_k + s_k^i$ . Tính  $f(x_k^i)$  //  $e_i$  là dòng thứ  $i$  của ma trận đơn vị  $I_n$
- Nếu  $f(x_k^i) < \min$ :
  - $\min = f(x_k^i)$
  - $s_k = s_k^i$
  - $\text{BestPoint} = x_k^i$
- Ngược lại: // Ta đổi ngược hướng tìm kiếm với hy vọng sẽ tìm được điểm tốt hơn
  - $s_k^i = s_k - \Delta_k e_i$  và  $x_k^i = x_k + s_k^i$ . Tính  $f(x_k^i)$
  - Nếu  $f(x_k^i) < \min$ :
    - $\min = f(x_k^i)$
    - $s_k = s_k^i$
    - $\text{BestPoint} = x_k^i$

#### Bước 4:

- Nếu  $\text{BestPoint} == x_k$  // Các điểm kề cận không tốt hơn điểm trung tâm  $\rightarrow$  giảm bước tìm kiếm
  - $x_{k+1} = x_k$
  - $\Delta_{k+1} = \Delta_k / 2$
- Ngược lại: // Tìm được điểm tốt hơn điểm trung tâm  $\rightarrow$  Gán lại trung tâm của pattern là điểm tốt hơn đó
  - $x_{k+1} = \text{BestPoint}$
  - $\Delta_{k+1} = \Delta_k$
- $k = k + 1$
- Quay lại bước 2

### 3.2.3.2.3 Các ý quan trọng về thuật toán Pattern Search

- Thuật toán đơn giản, dễ hiểu, dễ cài đặt, chi phí thấp.
- Khác với cách truyền thống là lấy đạo hàm của hàm mục tiêu (dở khi hàm mục tiêu không “convex”, có nhiều điểm cực trị địa phương; không làm được khi hàm mục tiêu không khả vi), ở đây ta không cần biết hàm mục tiêu có liên tục và khả vi hay không.
- Người ta đã chứng minh được rằng thuật toán hội tụ về cực trị địa phương (dễ hiểu vì ta thấy rằng sau mỗi vòng lặp giá trị hàm mục tiêu tại điểm trung tâm pattern đều giảm hoặc không đổi) và hơn nữa, điểm cực trị này thường khá tốt.

### 3.2.3.3 Cross-Validation truyền thống

Để chọn các tham số cho mô hình thì Grid Search và Pattern Search sẽ được kết hợp với Cross-Validation.

|        |        |        |        |        |
|--------|--------|--------|--------|--------|
| 1      | 2      | 3      | 4      | 5 test |
| 1 test | 2      | 3      | 4      | 5      |
| 1      | 2 test | 3      | 4      | 5      |
| 1      | 2      | 3 test | 4      | 5      |
| 1      | 2      | 3      | 4 test | 5      |

Với Cross-Validation truyền thống thì ta sẽ chia bộ train thành k phần bằng nhau (k fold), chẳng hạn ở đây là 5 phần.

- Phần 5 sẽ được test dựa trên train phần 1 – 4.
- Phần 1 sẽ được test dựa trên train phần 2 – 5.
- Phần 2 sẽ được test dựa trên train phần 1, 3, 4, 5.
- Phần 3 sẽ được test dựa trên train phần 1, 2, 4, 5.
- Phần 4 sẽ được test dựa trên train phần 1, 2, 3, 5.

Độ lỗi ứng với 5 lần test sẽ được lấy trung bình lại. Bộ tham số nào cho độ lỗi trung bình này nhỏ nhất sẽ được chọn.



## CHƯƠNG 4 CÁC CẢI TIẾN

### 4.1 CẢI TIẾN CROSS-VALIDATION TRONG TÌM THAM SỐ CỦA SVR

Ở đây, nhóm sử dụng hai phương pháp tìm tham số cho SVR là Grid Search và Pattern Search. Nhóm nhận thấy Grid Search với bản chất là vét cạn cho kết quả tốt hơn và ổn định hơn; tuy nhiên, nhược điểm của Grid Search là thời gian chạy quá lâu, đặc biệt là khi bộ huấn luyện lớn.

Điểm nữa là hiện nay trong hàm Cross-Validation của LIBSVM sử dụng phương pháp shuffle (xáo trộn các phần tử trong bộ huấn luyện rồi mới chia fold.) Phương pháp này có 2 nhược điểm:

- Tính không ổn định: mỗi lần ra một kết quả khác nhau, thời gian chạy khi nhanh khi lâu.
- Theo [2] thì phương pháp này không phù hợp đối với dữ liệu time series. Dữ liệu time series có đặc trưng là tính có hướng; nghĩa là ta chỉ có thể dùng dữ liệu được tạo ra sớm hơn để dự đoán dữ liệu được tạo ra trễ hơn mà không thể làm theo chiều ngược lại. Như vậy thì rõ ràng phương pháp shuffle sẽ làm mất tính chất này của dữ liệu time series.

Từ đây, [2] đề xuất phương pháp Cross-Validation cải tiến dành cho dữ liệu time series như sau:

|   |           |           |           |           |
|---|-----------|-----------|-----------|-----------|
| 1 | 2         | 3         | 4         | 5 predict |
| 1 | 2         | 3         | 4 predict |           |
| 1 | 2         | 3 predict |           |           |
| 1 | 2 predict |           |           |           |

Ta chia bộ huấn luyện làm 5 phần bằng nhau mà không xáo trộn bộ huấn luyện.

- Phần 5 sẽ được test dựa trên train phần 1-4.
- Phần 4 sẽ được test dựa trên train phần 1-3.
- Phần 3 sẽ được test dựa trên train phần 1-2.
- Phần 2 sẽ được test dựa trên train phần 1.

Nhận thấy với cách chia này thì ngoài việc đảm bảo tính có hướng của dữ liệu time series thì nó còn có 1 tác dụng nữa là làm giảm thời gian chạy cho Grid Search (Vì kích thước bộ train giảm dần thay vì luôn cố định như trong Cross-Validation truyền thống.)

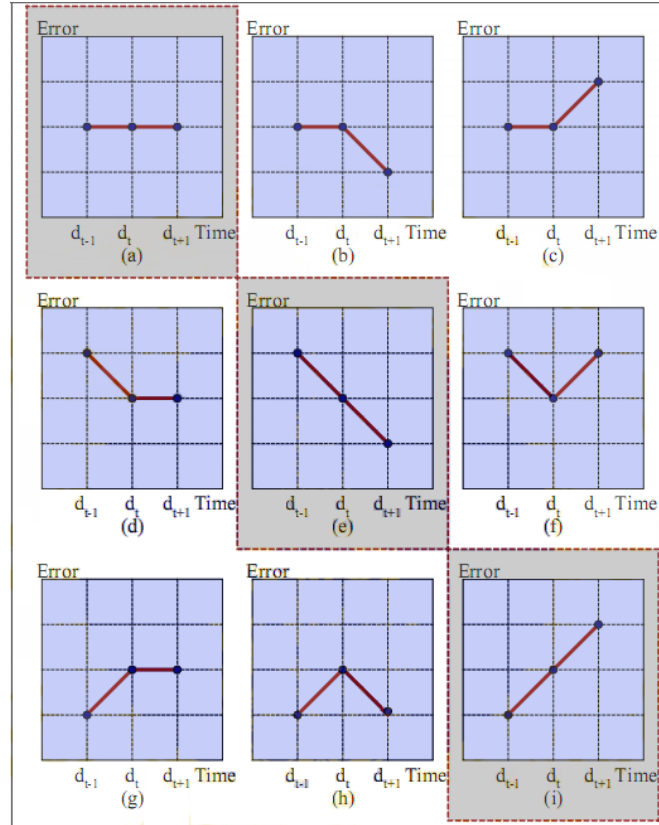
## 4.2 TĂNG ĐỘ CHÍNH XÁC VỀ HƯỚNG TRONG DỰ ĐOÁN GIÁ

Trong quá trình thí nghiệm, nhóm nhận thấy những trường hợp cho độ lỗi về giá trị nhỏ hơn chưa chắc đã cho sự đúng về xu hướng cao hơn.

Như đã nói mục tiêu của bài toán dự đoán giá là dự đoán đúng cả về giá trị lẫn xu hướng. Nhưng hiện nay quá trình huấn luyện chỉ tập trung vào việc minimize độ lỗi về giá trị. Như vậy, để đạt được mục tiêu đã nói ta cần đưa thêm thông tin về xu hướng vào trong quá trình huấn luyện.

[3] đề xuất một phương pháp như sau: với bộ huấn luyện, xét các output  $t_n$  thỏa:  $(t_n - t_{n-1}) * (t_{n+1} - t_n) > 0$  (giá tăng hoặc giảm 2 ngày liên tiếp).

Để cải thiện xu hướng ta thay đổi giá trị của output là  $t_{n+1}$  (thay vì trước đây là  $t_n$ ) với hy vọng là output sẽ thể hiện xu hướng rõ hơn.



Ở đây  $d_t$  chính là giá trị thực của ngày dự đoán,  $d_{t+1}$  là giá trị của ngày tiếp theo. Chúng ta chỉ quan tâm đến trường hợp (a), (e), (i) vì nó biểu lộ xu hướng rõ ràng tăng, giảm hoặc giữ nguyên.

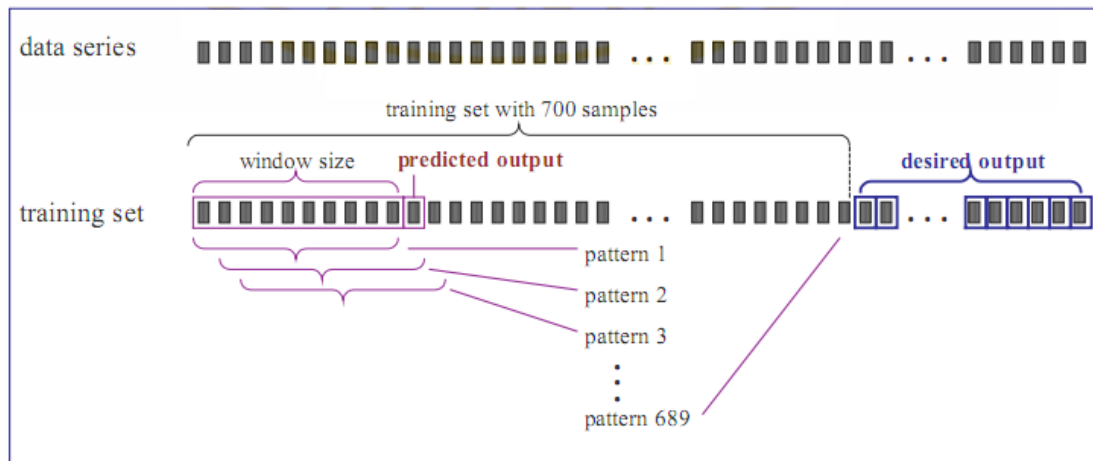
Như vậy giá trị  $t_n$  sẽ được tính lại theo công thức sau:

$$t_n = \begin{cases} d_{k,t} & \text{nếu } (d_{k,t} - d_{k,t-1}) = 0 \text{ và } (d_{k,t+1} - d_{k,t}) = 0 \\ d_{k,t+1} & \text{nếu } (d_{k,t} - d_{k,t-1}) * (d_{k,t+1} - d_{k,t}) > 0 \\ d_{k,t} & \text{trường hợp còn lại} \end{cases}$$

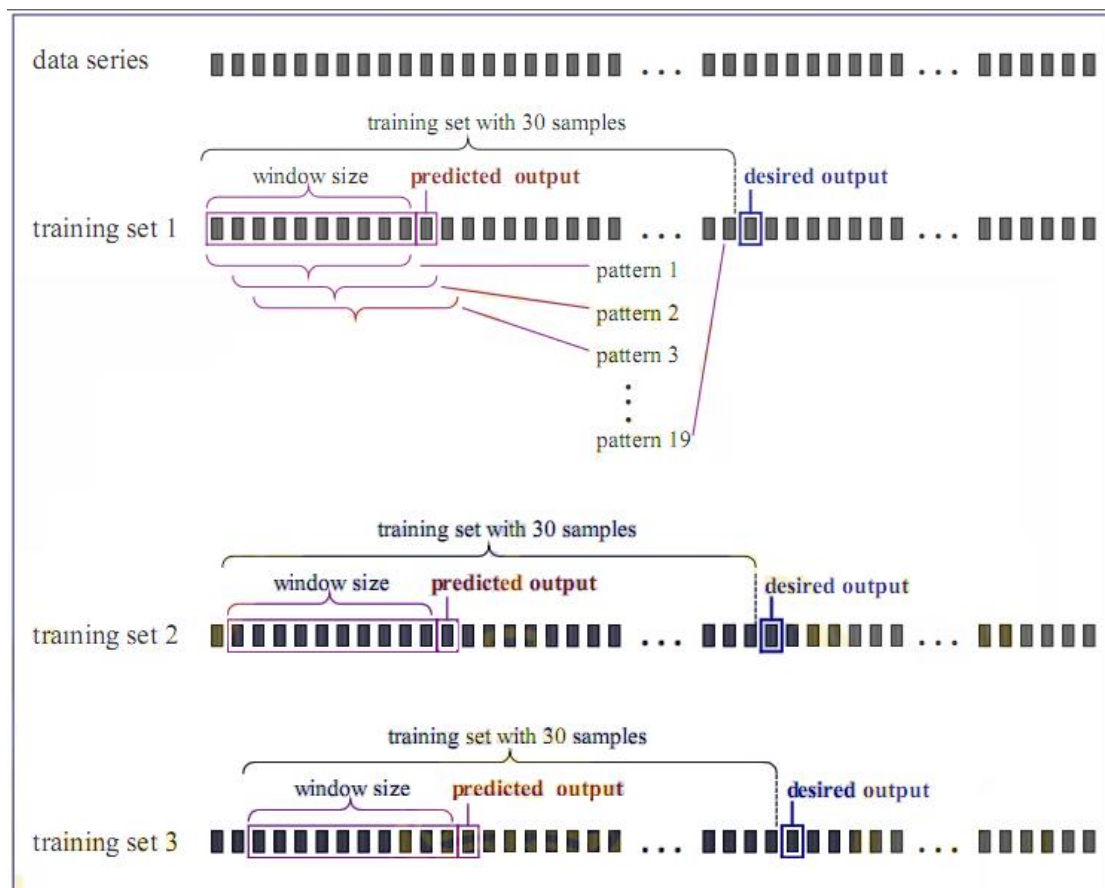
### 4.3 Cải tiến cách tổ chức dữ liệu đưa vào mô hình:

Hiện tại có 2 cách tổ chức dữ liệu đưa vào mô hình như sau:

- Cách truyền thống: phân dữ liệu có được ra làm 2 bộ dữ liệu con là training và test theo một tỉ lệ cho trước, yêu cầu bộ dữ liệu cho training nên có kích thước lớn.



- Cách cải tiến: Mỗi lần dự đoán một giá trị mới ta phải huấn luyện lại toàn bộ mô hình (một lần huấn luyện chỉ để dự đoán duy nhất 1 giá trị liền sau nó, để dự đoán giá trị tiếp theo đó ta phải xây dựng lại mô hình), nhưng bộ dữ liệu huấn bây giờ có kích thước nhỏ hơn rất nhiều so với cách truyền thống



Như đã nói, một trong những nhược điểm lớn của mô hình hiện tại là cần một lượng lớn dữ liệu mẫu cho quá trình huấn luyện, đó là một trong những nguyên nhân làm quá trình huấn luyện chậm đi, dẫn đến không thể xử lý trực tuyến khi có thông tin dữ liệu mới do quá trình huấn luyện mất nhiều thời gian, cho kết quả không tốt cho các mã chứng khoán mới ra vì bộ dữ liệu thu được ít.

Một lý do khác chúng ta tiến hành xử lý dữ liệu chuỗi thời gian nên sẽ có một số mẫu sau một thời gian nó không còn mang tính hợp lệ (như giai đoạn đầu mới mở giá có thể là 20\$, nhưng sau một thời gian hoạt động công ty làm ăn phát triển giá trị cổ phiếu khó có thể trở về 20\$ như ban đầu). Đó cũng là nhược điểm ở cách truyền thống

### 5.1 SVR

#### 5.1.1 Mô hình

Sử dụng thư viện LIBSVM làm nền tảng cho mô hình SVR. Bên cạnh đó, nhóm có chỉnh sửa phương pháp cross validation như đã trình bày, các tham số không phải là những giá trị tùy chọn mà được tối ưu bằng các phương pháp tìm tham số: Grid search và Pattern search. Quá trình huấn luyện cũng được tinh chỉnh để có thể mềm dẻo với các độ lỗi khác nhau.

#### 5.1.2 Chọn tham số

Ở đây, ta chỉ xét SVR với hàm lỗi  $\epsilon - insensitive$  và hàm Gaussian kernel.

Với mô hình này, ta cần tìm 3 tham số: C, gamma (của hàm Gauss), và epsilon.

Căn cứ vào các báo cáo và theo kinh nghiệm, ta chọn miền giá trị cho ba biến này như sau:

- $C \in [2^{-4}, 2^8]$
- $\text{Gamma} \in [2^{-9}, 2^3]$
- $\text{Epsilon} \in [2^{-14}, 2^{-2}]$

##### 5.1.2.1 Grid Search

Để bắt đầu, giá trị delta sẽ được gán bằng 2. Sau khi tìm được bộ ba ưng ý, delta sẽ được giảm xuống 0.25. Ở đây, interval được cho bằng 7.

Ví dụ: Giả sử đã tìm được các giá trị ở bước search thưa:  $C = 2^4$ ,  $\gamma = 2^{-2}$ ,  $\epsilon = 2^{-6}$ . Gán lại giá trị bắt đầu lân cận bộ ba này, sao cho từ vị trí đó đến giá trị hiện tại là  $7/2 = 3$  interval:

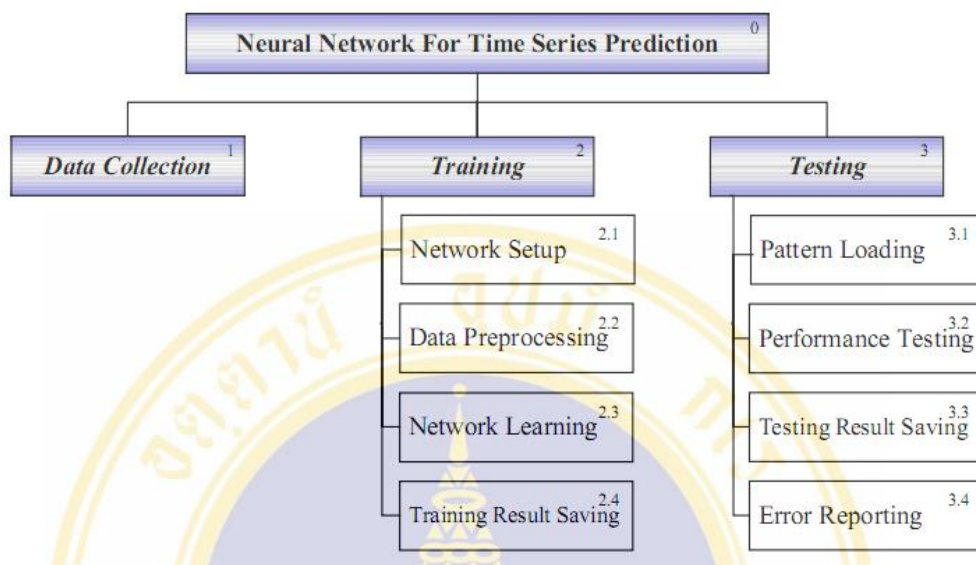
$$C = 2^{3.25}, \gamma = 2^{-1.25}, \epsilon = 2^{-5.25}.$$

#### 5.1.2.2 *Pattern Search*

- Đầu tiên, ta khởi tạo ngẫu nhiên một trung tâm của pattern nằm trong miền này.
- Khởi tạo bước tìm kiếm  $\Delta_k = 1$ , ngưỡng  $\tau = 0.1$
- Còn lại, cài đặt giống như lý thuyết ở trên.

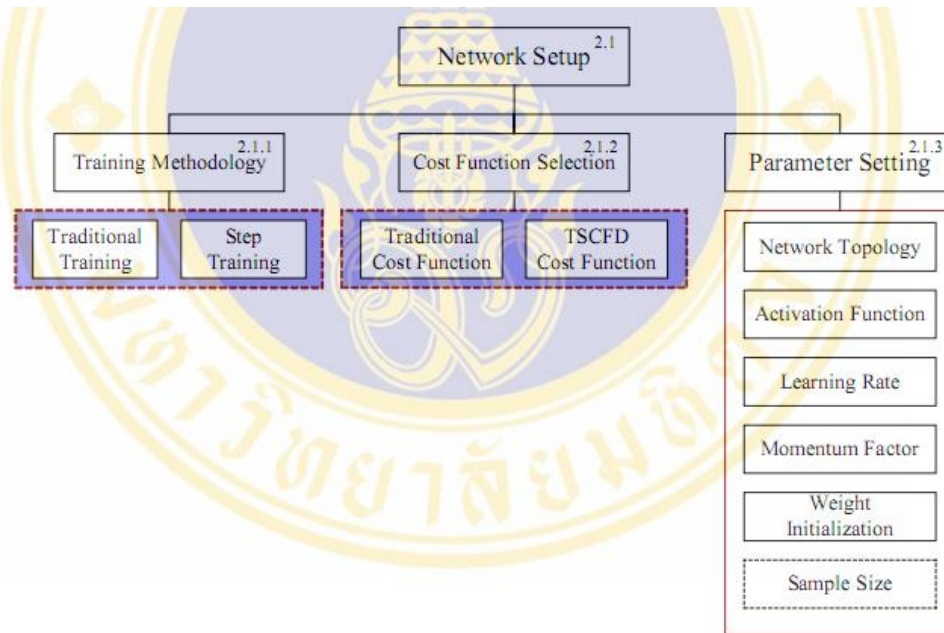
## 5.2 ANN

### 5.2.1 *Sơ đồ tổng quan các chức năng trong mô hình*



### 5.2.2 *Training*

#### 5.2.2.1 *Cài đặt Mạng*



#### 5.2.2.2 Tổ chức dữ liệu training:

Như đã trình bày ở trên, ta sẽ có 2 cách là cách truyền thống và cải tiến (step-training)

#### 5.2.2.3 Lựa chọn hàm chi phí:

Ta cũng sẽ có 2 hàm là truyền thống và hàm TSCFD.

#### 5.2.2.4 Các tham số:

Network Topology: Ở đây ta chọn mạng truyền thẳng 3 lớp với thuật toán học lan truyền ngược.

- Trong đó lớp output sẽ có 1 node xuất ra giá dự đoán.
- Số node ở lớp input và lớp ẩn là không biết trước, sẽ phải qua thực nghiệm để quyết định. (Bởi vì không có một giả thiết nào có trước đưa ra số node gần đúng cho mô hình.)
- Cấu trúc mạng mà một số bài báo đã áp dụng:
  - Input (5,6,10,15,20) – Hiden ( $\sqrt{\text{input} * \text{output}}$ )  
5-2-1, 5-4-1, 6-3-1, 10-3-1, 15-3-1, **20-2-1**
  - 64-8-1, 64-16-1, 64-32-1, **64-64-1**, 64-100-1



- 28-60-1 (Cố định số node ẩn = số node input = n, khảo sát n, rồi sau đó khảo sát nó node ẩn)
- ?-(9-14)-1

Hàm truyền phi tuyến: Nhìn chung đều là dùng hàm sigmoid, nhưng có một số dạng như sau

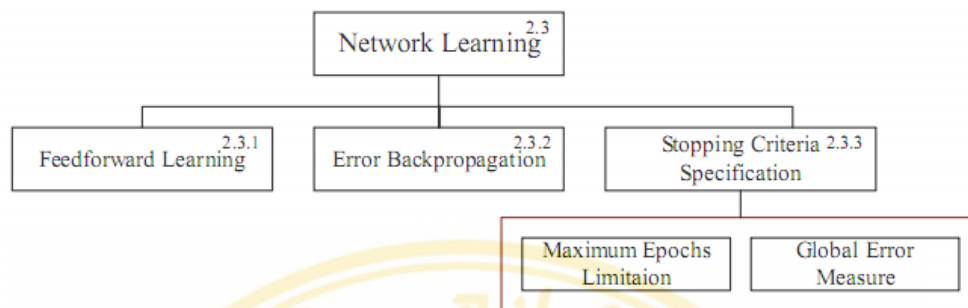
- $f(net) = \frac{1}{1+e^{-net}} f'(net) = f(net) * [1 - f(net)]$
- $f(net) = \frac{2}{1+e^{-net}} - 1 f'(net) = \frac{1}{2} [1 + f(net)][1 - f(net)]$

Hệ số học  $\eta$ : Hệ số lớn sẽ giúp đẩy nhanh tốc độ học, nhưng nếu quá cao sẽ làm cho kết quả không thể hội tụ như mong muốn (step over). Miền giá trị một số bài báo đưa ra là 0.1, 0.01-0.05

Momentum Factor  $\alpha$ : Giúp cho việc tránh rơi vào bẫy tối ưu cục bộ. Cần qua thực nghiệm để lựa chọn.

Khởi tạo bộ trọng số w ngẫu nhiên trong đoạn [0,1].

### 5.2.3 *Quá trình training:*



Global Error Measure: Ta sẽ dùng MSE, MAE, MAPE, Tolerance và so sánh kết quả xem cái nào tốt hơn.

### 6.1 DỰ ĐOÁN GIÁ

#### 6.1.1 Mô tả dữ liệu

- Nước ngoài:
  - IBM: từ 2/1/2001 đến 31/12/2002; gồm 500 điểm dữ liệu.
  - YA: từ 9/1/2004 đến 31/12/2007; gồm 1001 điểm dữ liệu.
- Việt Nam:
  - FTP: từ 13/12/2006 đến 9/8/2010; gồm 914 điểm dữ liệu.
  - DHG: từ 21/12/2006 đến 10/12/2010; gồm 994 điểm dữ liệu.

Các bộ dữ liệu được chia thành 2 bộ train và test theo tỉ lệ: 80%:20%. Số node đầu vào là 5.

Giải thích về số node đầu vào: Ta có thể phát biểu bài toán hồi qui theo giá với SVR như sau:

Tìm  $f$  sao cho:  $\text{price}_k = f(\text{price}_{k-1}, \dots, \text{price}_{k-n})$

Ở đây,  $n$  chính là số node đầu vào của SVR.

#### 6.1.2 Thí nghiệm

Ở đây, SVR sử dụng Grid Search để tìm tham số. Ta sẽ so sánh giữa các phương pháp:

- Nhóm SVR:
  - SVR + Shuffle Cross-Validation (1)
  - SVR + Improved Cross-Validation (2)
  - SVR + Improved Cross-Validation + Improved Direction (3)
  -
- Nhóm ANN:
  - ANN (4)
  - ANN + Step Training (5)

- ANN + Step Training + Improved Direction (6)

### 6.1.3 Kết quả

- Bộ nước ngoài:
  - IBM

|     | MSE      | MAPE     | DM4Price-Right | DM4Price-Wrong | DM4Price-CannotMeasure | DS       | Thời gian search tham số (phút) |
|-----|----------|----------|----------------|----------------|------------------------|----------|---------------------------------|
| (1) | 2.416134 | 2.450208 | 46.94%         | 53.06%         | 0%                     | 55.10204 | 4.466322                        |
| (2) | 2.184375 | 2.206432 | 45.92%         | 54.08%         | 0%                     | 53.06122 | 1.042342                        |
| (3) | 2.408207 | 2.283254 | 54.08%         | 45.92%         | 0%                     | 58.16327 | 0.992422                        |
| (4) | 2.242792 | 2.317366 | 57.14%         | 42.85%         | 0%                     | 54.08163 |                                 |
| (5) | 2.690727 | 2.454764 | 41.83%         | 58.16          | 0%                     | 60.20408 |                                 |
| (6) | 2.538417 | 2.348978 | 40.81%         | 59.18          | 0%                     | 54.08163 |                                 |

- NYA

|     | MSE       | MAPE     | DM4Price-Right | DM4Price-Wrong | DM4Price-CannotMeasure | DS       | Thời gian search tham số (phút) |
|-----|-----------|----------|----------------|----------------|------------------------|----------|---------------------------------|
| (1) | 5970.9    | 0.896274 | 47.24%         | 52.76%         | 0%                     | 45.72864 | 29.18472                        |
| (2) | 5650.005  | 0.831292 | 52.76%         | 47.24%         | 0%                     | 45.72864 | 4.771268                        |
| (3) | 5594.911  | 0.845839 | 54.27%         | 45.73%         | 0%                     | 46.23116 | 4.451208                        |
| (4) | 58687.825 | 3.153585 | 43.21%         | 56.78%         | 0%                     | 44.72361 |                                 |
| (5) | 6002.368  | 0.873353 | 54.77%         | 45.22%         | 0%                     | 45.72864 |                                 |
| (6) | 6448.194  | 0.884658 | 56.78%         | 43.21%         | 0%                     | 45.72864 |                                 |

- Bộ Việt Nam:
  - FPT

|     | MSE      | MAPE     | DM4Price-Right | DM4Price-Wrong | DM4Price-CannotMeasure | DS       | Thời gian search tham số (phút) |
|-----|----------|----------|----------------|----------------|------------------------|----------|---------------------------------|
| (1) | 2.822008 | 2.08442  | 41.44%         | 43.65%         | 14.92%                 | 58.56354 | 82.47512                        |
| (2) | 2.905566 | 2.107643 | 39.78%         | 45.30%         | 14.92%                 | 54.69613 | 3.874267                        |
| (3) | 2.94058  | 2.162133 | 44.75%         | 40.33%         | 14.92%                 | 59.66851 | 7.613333                        |
| (4) | 3.239410 | 2.474532 | 42.21%         | 41.70%         | 16.08%                 | 55.27638 |                                 |
| (5) | 2.620597 | 1.844288 | 45.72%         | 38.19%         | 16.08%                 | 58.29145 |                                 |
| (6) | 2.289264 | 1.739638 | 44.22%         | 39.69%         | 16.08%                 | 57.78894 |                                 |

○ DHG

|     | MSE      | MAPE     | DM4Price-<br>Right | DM4Price-<br>Wrong | DM4Price-<br>CannotMeasure | DS       | Thời gian<br>search<br>tham số<br>(phút) |
|-----|----------|----------|--------------------|--------------------|----------------------------|----------|--|
| (1) | 0.992382 | 0.871604 | 25.89%             | 30.46%             | 43.65%                     | 70.05076 | 58.47058                                 |
| (2) | 0.993292 | 0.896034 | 28.43%             | 27.92%             | 43.65%                     | 69.54315 | 6.746752                                 |
| (3) | 1.19722  | 0.976193 | 25.38%             | 30.96%             | 43.65%                     | 69.03553 | 10.80942                                 |
| (4) | 1.023388 | 0.935622 | 29.94%             | 26.39%             | 43.65%                     | 69.03553 |  |
| (5) | 0.882822 | 0.801306 | 37.56%             | 18.78%             | 43.65%                     | 62.94416 |  |
| (6) | 0.859166 | 0.803013 | 36.04%             | 20.30%             | 43.65%                     | 64.46700 |  |

#### 6.1.4 Nhận xét

- Trong nhóm SVR:
  - Phương pháp Cross-Validation cải tiến cho độ lỗi MSE thấp hơn Cross-Validation truyền thống ở 2 mã nước ngoài và cho MSE cao hơn (nhưng không đáng kể) ở 2 mã Việt Nam. Tuy nhiên, điều quan trọng là phương pháp Cross-Validation cải tiến đã giúp cho thời gian chạy Grid Search giảm đáng kể.
  - Ta đạt được kết quả tương đối tốt về giá trị. Về xu hướng, phương pháp cải tiến để tăng độ chính xác về xu hướng đã đạt được hiệu quả đối với 2 mã nước ngoài và 1 mã Việt Nam (FPT.) Tuy nhiên, sự đúng về xu hướng vẫn còn dưới 60%. Trong tương lai, ta cần đưa thêm các thông tin về xu hướng vào quá trình huấn luyện để tăng độ chính xác về xu hướng hơn nữa.
  - Kết hợp 2 điều trên và điểm nữa là: ta thấy giá các mã Việt Nam thường xuyên đi ngang (giá không đổi) dẫn đến khó đánh giá mô hình một lần nữa cho ta thấy những khó khăn trong việc dự đoán thị trường chứng khoán Việt Nam.
- Trong nhóm ANN:
  - Phương pháp Step-Training cho độ lỗi MSE thấp hơn và thể hiện xu hướng tốt hơn so với cách train truyền thống ở ¾ mã chứng khoán, riêng mã IBM chỉ cao hơn không đáng kể. Tuy nhiên điều quan trọng khi áp dụng step-training ta có thể dự đoán trực tuyến vì bộ dữ liệu huấn luyện bây giờ rất nhỏ, và

độ lỗi NMSE (chuẩn hóa của MSE) luôn cho dao động cố định.

- Khi ta kết hợp thêm cải tiến ImprovedDirection về độ lỗi MSE thì cho kết quả tốt hơn ở  $\frac{3}{4}$  mã chứng khoán, nhưng bù lại yếu tố xu hướng lại giảm đi ở các mã tương ứng ???
- SVR vs. ANN:
  - Xét về yếu tố xu hướng đúng thì ta thấy ở mô hình ANN trong cả 4 mã chứng khoán đều cho kết quả tốt hơn so với SVR, đặc biệt là với ANN-StepTraining, điều đó cho thấy StepTraining đã thể hiện đúng mục đích của nó là loại bỏ các dữ liệu lỗi thời, thể hiện đúng bản chất của dữ liệu trong thời điểm dự đoán.
  - Xét về độ lỗi MSE thì SVR cho kết quả tốt hơn ở 2 mã chứng khoán nước ngoài, còn ANN cho kết quả tốt hơn ở 2 mã chứng khoán Việt Nam

## 6.2 DỰ ĐOÁN XU HƯỚNG

### 6.2.1 Mô tả dữ liệu

- Nước ngoài:
  - NYSE: từ 9/1/2004 đến 31/12/2007; gồm 1001 điểm dữ liệu.[4]
- Việt Nam:
  - FTP: từ 13/12/2006 đến 9/8/2010; gồm 914 điểm dữ liệu.
  - DHG: từ 21/12/2006 đến 10/12/2010; gồm 994 điểm dữ liệu.
  - VIS: từ 25/12/2006 đến 10/12/2010; gồm 988 điểm dữ liệu.
  - VNM: từ 19/1/2006 đến 10/12/2010; gồm 1222 điểm dữ liệu.
  - BT6: từ 18/04/2002 đến 10/12/2010; gồm 2164 điểm dữ liệu.

Dữ liệu được chia làm 2 phần: huấn luyện và kiểm thử. Đối với mã NYSE để tương đồng với [4], tỷ lệ huấn luyện/kiểm thử là 9/1. Đối với các mã Việt Nam, tỷ lệ này là 8/1. Tất cả dữ liệu được khảo sát với số node là 5.

Đối với từng period khác nhau, giá trị từng node có sự điều chỉnh.

- Period = 1 và 5: mỗi node là return của 1 ngày trước đó.
- Period = 10: mỗi node là return của từng bộ 2 ngày trước.

- Period = 30: mỗi node là return của từng bộ 6 ngày trước.

Ví dụ với period = 10, thời điểm hiện tại là ngày 10, thời điểm cần dự đoán là ngày 20. Nhân dự đoán sẽ là return 20 – 10 (của ngày 20 so với ngày 10), các node bên trong lần lượt là:

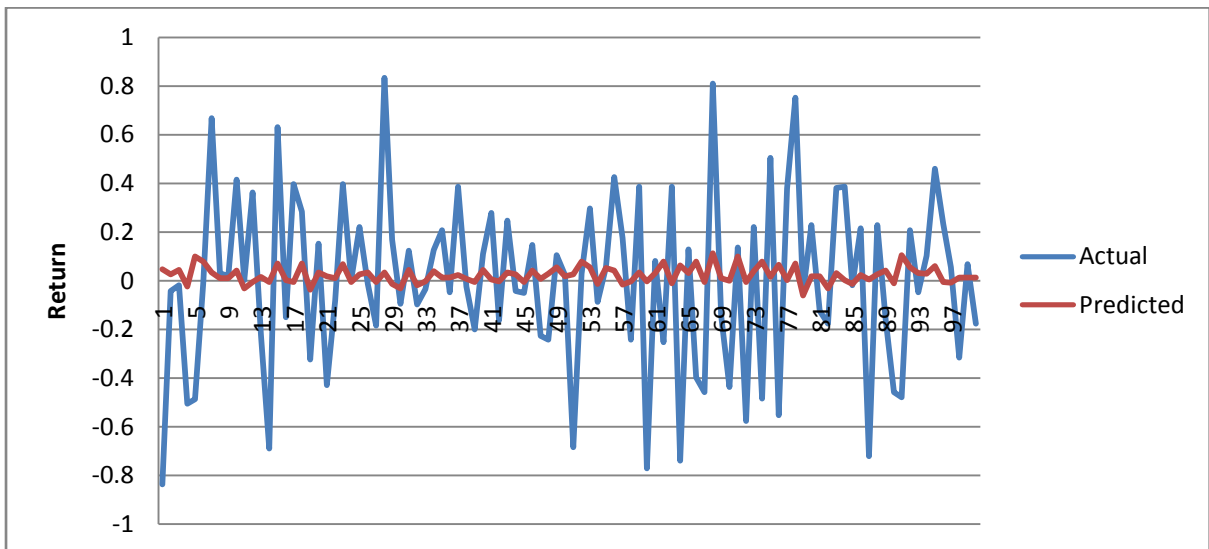
**1**: return 2 – 0, **2**: return 4 – 2, **3**: return 6 – 4, **4**: return 8 – 6, **5** : return 10 – 8.

### 6.2.2 *Thí nghiệm*

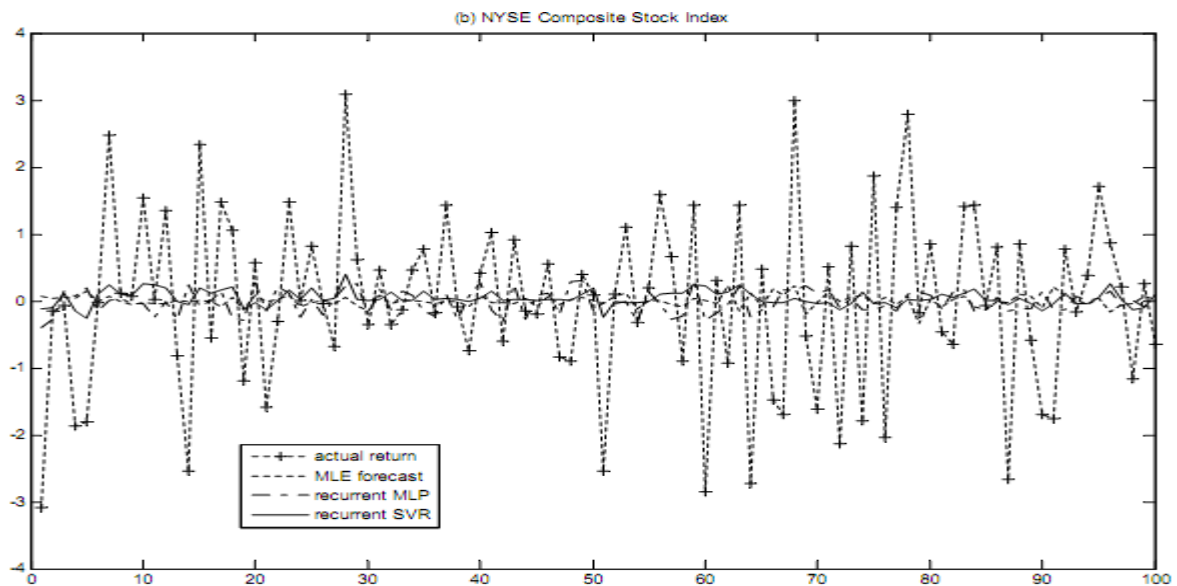
#### 6.2.2.1 *Với mã nước ngoài*

Đối với mã NYSE ta chỉ kiểm thử với period = 1 nhằm so sánh kết quả đối với mô hình [4]. Với SVR, ta dùng 2 phương pháp tìm tham số là grid search và pattern search.

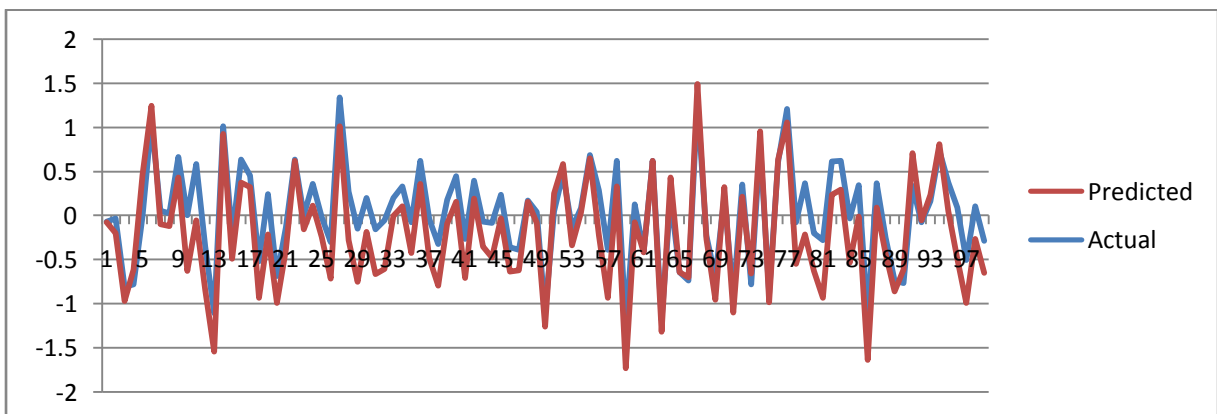
|             |                | NMSE  | Sign(%) |
|-------------|----------------|-------|---------|
| SVR         | Pattern search | 96.75 | 60      |
|             | Grid search    | 96.74 | 62      |
| ANN         |                | 103   | 54      |
| Mô hình [4] |                | 99.41 | 68.69   |



**Figure 1 Đồ thị kết quả SVR với grid search**



**Figure 2 Đồ thị kết quả của mô hình [4]**



**Figure 3 Kết quả với period = 1, hệ số học = 0,3**

#### 6.2.2.2 Với mã trong nước

### 6.3 Tham số của ANN

Các tham số của mạng như sau:

- Số node tầng nhập: 5.
- Số node tầng ẩn: 4.
- Số node tầng xuất: 1.
- Hệ số học 0.3
- Số vòng lặp tối đa 2000
- Bias 0
- Độ chính xác khi train 90%

### Tham số của SVR

Ta chọn tham số dựa trên 2 phương pháp grid search và pattern search.

Kết quả khi thực hiện kiểm thử với từng Period

- Period = 1:

| ANN        |      |         |             |       |                |                  |          |
|------------|------|---------|-------------|-------|----------------|------------------|----------|
|            | NMSE | SIGN(%) | DM4Price(%) |       |                | Độ lỗi khi train |          |
|            |      |         | Right       | Wrong | Cannot measure | NMSE             | Max Loop |
| <b>BT6</b> | 1.28 | 42      | 72.09       | 19.06 | 8.83           | 1.65             | 2000     |
| <b>DHG</b> | 1.34 | 25      | 51.02       | 13.27 | 35.71          | 1.03             | 2000     |
| <b>FPT</b> | 1.3  | 43      | 72.73       | 23.23 | 4.04           | 1.18             | 2000     |
| <b>VIS</b> | 1.03 | 41      | 65.31       | 34.69 | 0              | 1.04             | 2000     |
| <b>VNM</b> | 1.15 | 32      | 57.85       | 21.48 | 20.66          | 0.98             | 2000     |

| SVR – Gid search |      |         |             |       |                |
|------------------|------|---------|-------------|-------|----------------|
|                  | NMSE | SIGN(%) | DM4Price(%) |       |                |
|                  |      |         | Right       | Wrong | Cannot measure |
| <b>BT6</b>       | 1.28 | 41      | 72.62       | 18.55 | 8.83           |
| <b>DHG</b>       | 1.27 | 40      | 56.5        | 7.7   | 35.71          |



|            |      |    |       |       |       |
|------------|------|----|-------|-------|-------|
| <b>FPT</b> | 1.02 | 45 | 75.87 | 20.09 | 4.04  |
| <b>VIS</b> | 1.07 | 55 | 71.94 | 28.06 | 0     |
| <b>VNM</b> | 1.01 | 33 | 69.13 | 10.2  | 20.66 |

| <b>SVR – Pattern search</b> |      |         |             |       |                |
|-----------------------------|------|---------|-------------|-------|----------------|
|                             | NMSE | SIGN(%) | DM4Price(%) |       |                |
|                             |      |         | Right       | Wrong | Cannot measure |
| <b>BT6</b>                  | 1.24 | 44      | 72.22       | 18.95 | 8.83           |
| <b>DHG</b>                  | 1.13 | 40      | 54.5        | 9.7   | 35.71          |
| <b>FPT</b>                  | 1.34 | 44      | 75.87       | 20.09 | 4.04           |
| <b>VIS</b>                  | 1.02 | 52      | 70.9        | 29.1  | 0              |
| <b>VNM</b>                  | 1.12 | 38      | 70.2        | 9.12  | 20.66          |

- Period = 5:

| <b>ANN</b> |      |         |             |       |                |                  |          |
|------------|------|---------|-------------|-------|----------------|------------------|----------|
|            | NMSE | SIGN(%) | DM4Price(%) |       |                | Độ lỗi khi train |          |
|            |      |         | Right       | Wrong | Cannot measure | NMSE             | Max Loop |
| <b>BT6</b> | 1.4  | 44      | 58.69       | 32.39 | 8.92           | 1.38             | 2000     |
| <b>DHG</b> | 1.17 | 25      | 41.67       | 27.08 | 31.25          | 1.01             | 2000     |
| <b>FPT</b> | 1.21 | 38      | 56.7        | 42.27 | 1.03           | 1.78             | 2000     |
| <b>VIS</b> | 1.03 | 70      | 62.11       | 37.89 | 0              | 1                | 2000     |
| <b>VNM</b> | 1.78 | 50      | 48.74       | 30.25 | 21             | 1.04             | 2000     |

| <b>SVR – Grid search</b> |      |         |             |       |                |
|--------------------------|------|---------|-------------|-------|----------------|
|                          | NMSE | SIGN(%) | DM4Price(%) |       |                |
|                          |      |         | Right       | Wrong | Cannot measure |
| <b>BT6</b>               | 1.29 | 45      | 58.92       | 32.16 | 8.92           |
| <b>DHG</b>               | 1.14 | 31      | 55.1        | 13.65 | 31.25          |
| <b>FPT</b>               | 1.04 | 42      | 52.97       | 45.9  | 1.03           |
| <b>VIS</b>               | 1.8  | 51      | 54.97       | 45.02 | 0              |
| <b>VNM</b>               | 1.21 | 45      | 55.71       | 23.27 | 21             |

| <b>SVR– Pattern search</b> |      |         |             |       |                |
|----------------------------|------|---------|-------------|-------|----------------|
|                            | NMSE | SIGN(%) | DM4Price(%) |       |                |
|                            |      |         | Right       | Wrong | Cannot measure |
| <b>BT6</b>                 | 1.15 | 44      | 58.1        | 32.98 | 8.92           |

|            |      |    |       |       |       |
|------------|------|----|-------|-------|-------|
| <b>DHG</b> | 1.21 | 32 | 55.35 | 13.4  | 31.25 |
| <b>FPT</b> | 1.24 | 41 | 52.23 | 46.64 | 1.03  |
| <b>VIS</b> | 1.14 | 50 | 55    | 45.   | 0     |
| <b>VNM</b> | 1.53 | 45 | 54    | 25    | 21    |

- Period = 10:

| <b>ANN</b> |      |         |             |       |                |                  |          |
|------------|------|---------|-------------|-------|----------------|------------------|----------|
|            | NMSE | SIGN(%) | DM4Price(%) |       |                | Độ lỗi khi train |          |
|            |      |         | Right       | Wrong | Cannot measure | NMSE             | Max Loop |
| <b>BT6</b> | 2.32 | 39      | 51.42       | 42.86 | 5.71           | 1.03             | 2000     |
| <b>DHG</b> | 1.09 | 25      | 38.7        | 29.03 | 32.25          | 0.98             | 2000     |
| <b>FPT</b> | 1.41 | 61      | 58.51       | 36.17 | 5.31           | 2.18             | 2000     |
| <b>VIS</b> | 0.93 | 73      | 50          | 50    | 0              | 1.07             | 2000     |
| <b>VNM</b> | 1.35 | 25      | 38.79       | 46.55 | 14.66          | 1.12             | 2000     |

| <b>SVR – Grid search</b> |      |         |             |       |                |
|--------------------------|------|---------|-------------|-------|----------------|
|                          | NMSE | SIGN(%) | DM4Price(%) |       |                |
|                          |      |         | Right       | Wrong | Cannot measure |
| <b>BT6</b>               | 1.67 | 44      | 56.66       | 37.62 | 5.71           |
| <b>DHG</b>               | 1.4  | 31      | 48.45       | 19.29 | 32.25          |
| <b>FPT</b>               | 1.18 | 35      | 52.04       | 42.64 | 5.31           |
| <b>VIS</b>               | 1.66 | 46      | 52.43       | 47.56 | 0              |
| <b>VNM</b>               | 0.99 | 52      | 48.33       | 37    | 14.66          |

| <b>SVR– Pattern search</b> |      |         |             |       |                |
|----------------------------|------|---------|-------------|-------|----------------|
|                            | NMSE | SIGN(%) | DM4Price(%) |       |                |
|                            |      |         | Right       | Wrong | Cannot measure |
| <b>BT6</b>                 | 1.26 | 42      | 55.12       | 39.16 | 5.71           |
| <b>DHG</b>                 | 1.56 | 28      | 46          | 21.74 | 32.25          |
| <b>FPT</b>                 | 1.32 | 40      | 52.2        | 42.48 | 5.31           |
| <b>VIS</b>                 | 1.59 | 57      | 52.13       | 47.86 | 0              |
| <b>VNM</b>                 | 1.03 | 51      | 45.3        | 40.03 | 14.66          |

- Period = 30:

---

ANN

---

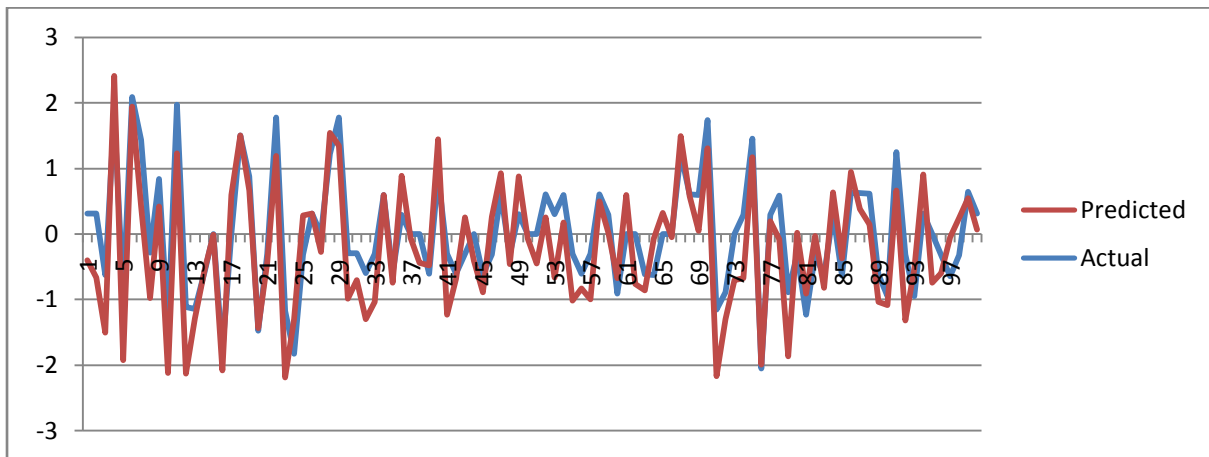
|            | NMSE | SIGN(%) | DM4Price(%) |       |                | Độ lỗi khi train |          |
|------------|------|---------|-------------|-------|----------------|------------------|----------|
|            |      |         | Right       | Wrong | Cannot measure | NMSE             | Max Loop |
| <b>BT6</b> | 0.91 | 46      | 51.01       | 42.93 | 6.06           | 0.98             | 2000     |
| <b>DHG</b> | 2.57 | 68      | 39.5        | 29.63 | 30.86          | 1.15             | 2000     |
| <b>FPT</b> | 1.89 | 55      | 57.31       | 40.24 | 2.43           | 1.04             | 2000     |
| <b>VIS</b> | 1.18 | 79      | 48.75       | 51.25 | 0              | 1.04             | 2000     |
| <b>VNM</b> | 1.41 | 52      | 44.23       | 35.58 | 20.19          | 1.17             | 2000     |

| <b>SVR – Grid search</b> |      |         |             |       |                |  |  |
|--------------------------|------|---------|-------------|-------|----------------|--|--|
|                          | NMSE | SIGN(%) | DM4Price(%) |       |                |  |  |
|                          |      |         | Right       | Wrong | Cannot measure |  |  |
| <b>BT6</b>               | 1.84 | 47      | 50.75       | 43.18 | 6.06           |  |  |
| <b>DHG</b>               | 1.89 | 48      | 45.16       | 23.98 | 30.86          |  |  |
| <b>FPT</b>               | 3.47 | 56      | 54.78       | 42.78 | 2.43           |  |  |
| <b>VIS</b>               | 1.6  | 41      | 49.03       | 50.93 | 0              |  |  |
| <b>VNM</b>               | 3.6  | 52      | 45.26       | 34.54 | 20.19          |  |  |

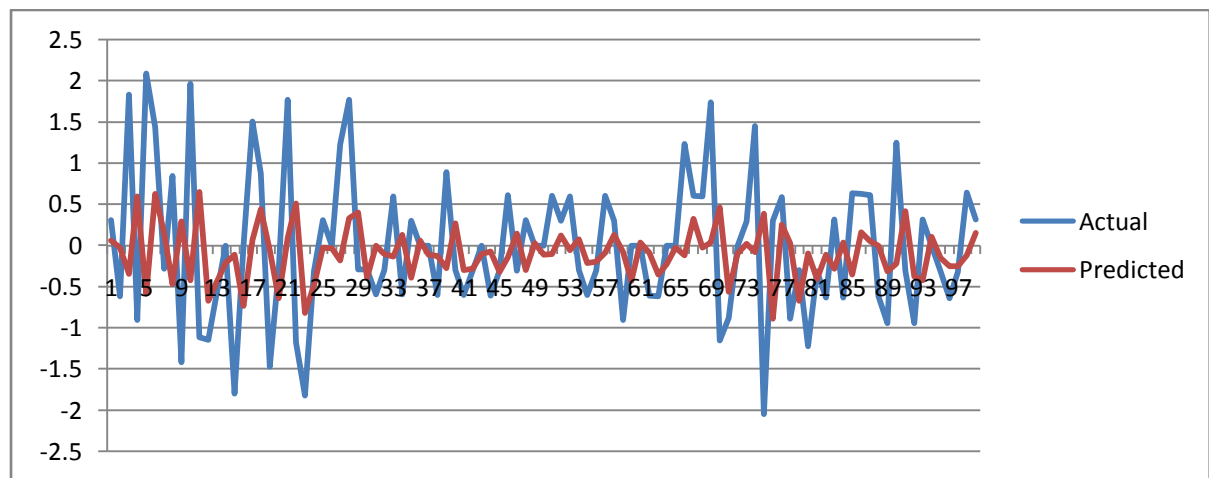
| <b>SVR– Pattern search</b> |      |         |             |       |                |  |  |
|----------------------------|------|---------|-------------|-------|----------------|--|--|
|                            | NMSE | SIGN(%) | DM4Price(%) |       |                |  |  |
|                            |      |         | Right       | Wrong | Cannot measure |  |  |
| <b>BT6</b>                 | 1.62 | 48      | 50          | 43.94 | 6.06           |  |  |
| <b>DHG</b>                 | 1.54 | 46      | 44.9        | 24.25 | 30.86          |  |  |
| <b>FPT</b>                 | 2.12 | 55      | 54.59       | 42.98 | 2.43           |  |  |
| <b>VIS</b>                 | 1.9  | 40      | 46.2        | 53.78 | 0              |  |  |
| <b>VNM</b>                 | 3.76 | 52      | 48.2        | 31.61 | 20.19          |  |  |

Dưới đây là các kết quả biểu diễn bằng đồ thị của FPT

- Period = 1:
- ANN

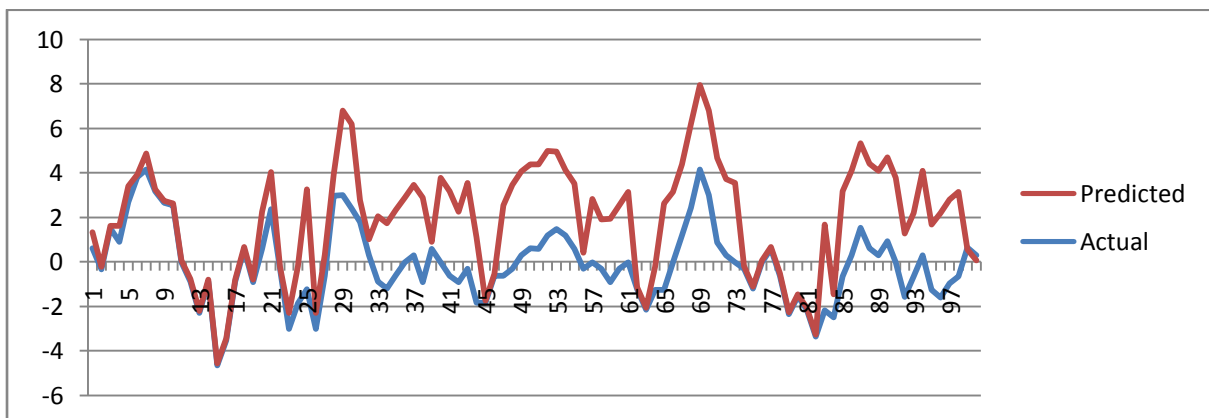


- SVR

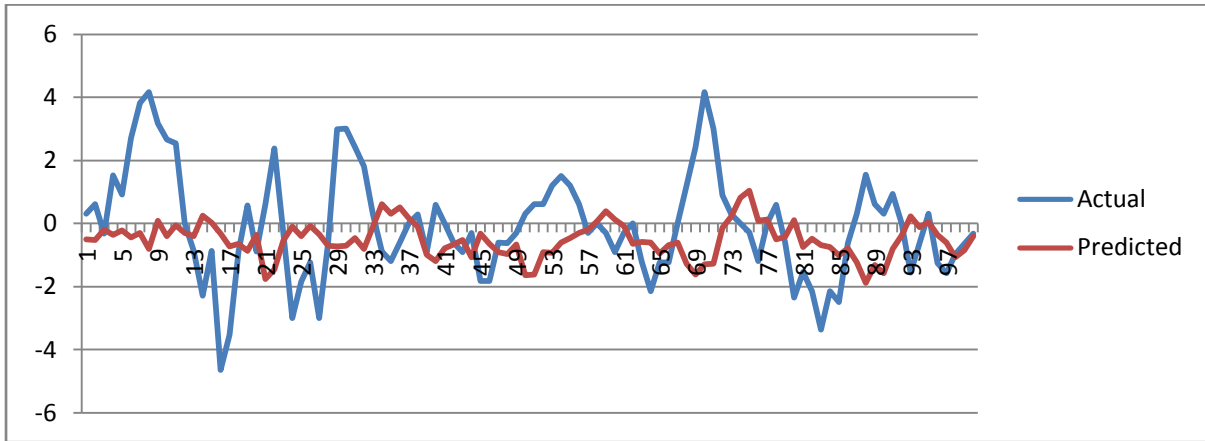


• Period = 5:

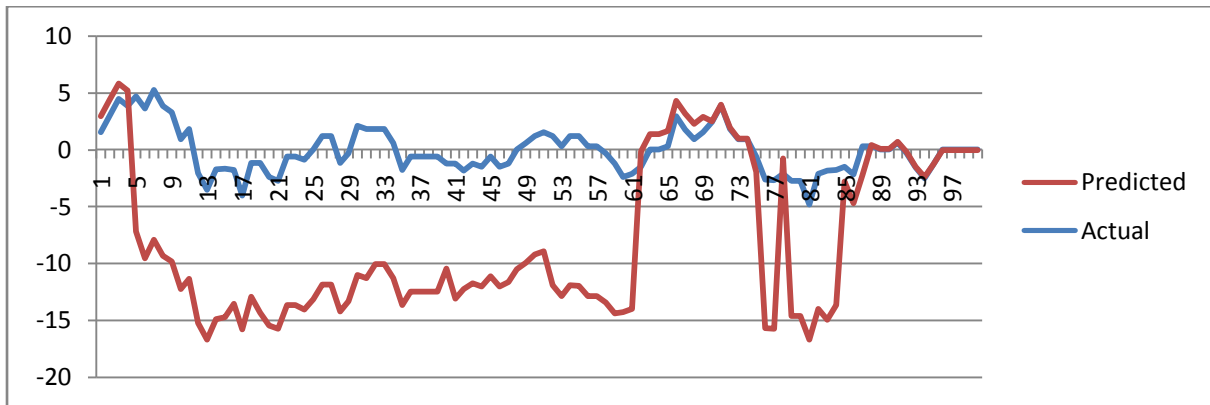
- ANN



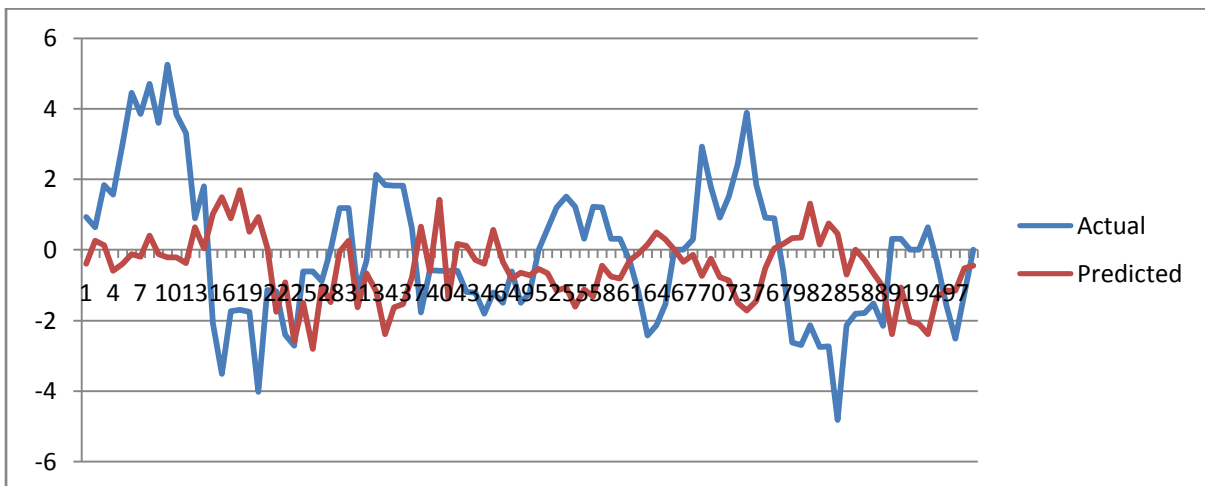
- SVR



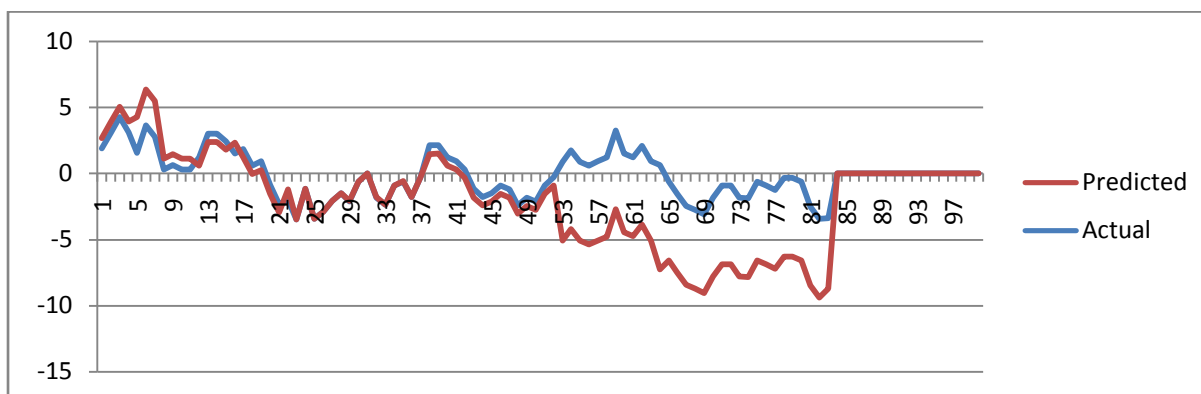
- Period = 10:
- ANN



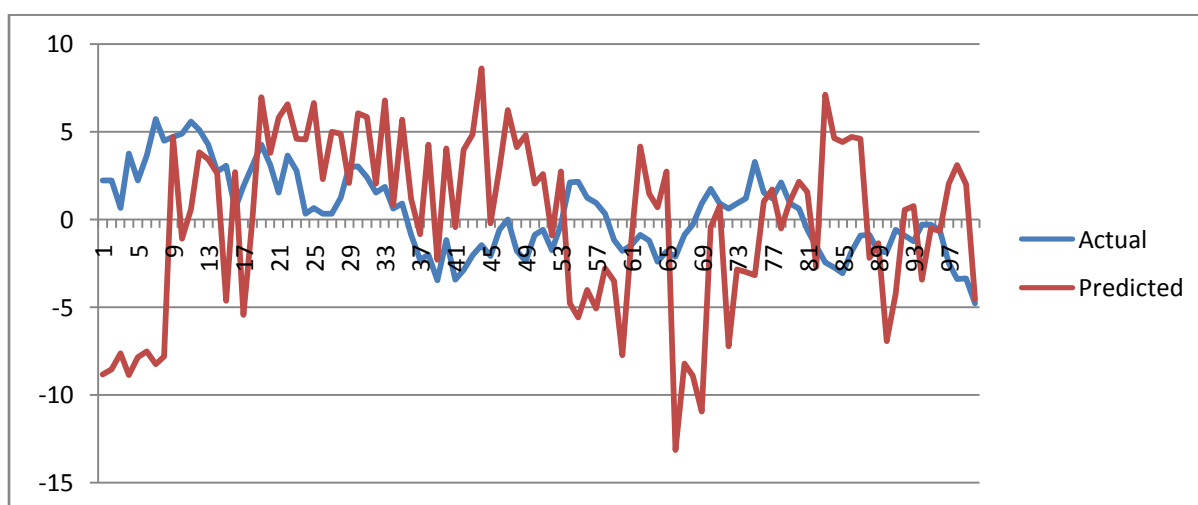
- SVR



- Period 30:
- ANN



- SVR



### 6.3.1 Nhận xét

- Đối với việc so sánh với mô hình [4], SVR và ANN của nhóm kém hơn. Tuy nhiên, điều này cũng dễ hiểu vì mô hình [4] sử dụng SVR có cải tiến. Bên cạnh đó, ta thấy SVR cho kết quả tốt hơn khá nhiều so với ANN.
- Đối với dữ liệu thị trường trong nước, SVR cũng cho kết quả tốt hơn ANN. Tuy nhiên ở period 30, ANN lại có được hiệu quả cao hơn SVR.
- Trong mô hình SVR cho xu hướng, việc sử dụng phương pháp grid search tỏ ra tối ưu hơn pattern search.
- Như đã đề cập, việc giá giữ nguyên sau vài ngày (return = 0) làm cho kết quả dự đoán đánh giá bằng Sign không cao (không quá 60%). Tuy vậy, với cách đánh giá bằng DM4, mô hình cho kết quả khả quan.

## CHƯƠNG 7 KẾT LUẬN VÀ CÔNG VIỆC TƯƠNG LAI

### 7.1 KẾT LUẬN

#### 7.1.1 Dự đoán giá

Về phần SVR, ta có một vài nhận định như sau:

- Phương pháp Cross-Validation cải tiến đã giúp cho thời gian chạy Grid Search giảm đáng kể.
- Ta đạt được kết quả tương đối tốt về giá trị. Về xu hướng, phương pháp cải tiến để tăng độ chính xác về xu hướng đã đạt được hiệu quả đối với 2 mã nước ngoài và 1 mã Việt Nam (FPT.) Tuy nhiên, sự đúng về xu hướng vẫn còn dưới 60%. Trong tương lai, ta cần đưa thêm các thông tin về xu hướng vào quá trình huấn luyện để tăng độ chính xác về xu hướng hơn nữa.

Về phần ANN, có một số kết luận như sau:

- Phương pháp Step-Training cho độ lỗi MSE thấp hơn và thể hiện xu hướng tốt hơn so với cách train truyền thống.

Xét về yếu tố xu hướng, mô hình ANN trong cả 4 mã chứng khoán đều cho kết quả tốt hơn so với SVR. Tuy nhiên, về độ lỗi MSE thì SVR cho kết quả tốt hơn ở 2 mã chứng khoán nước ngoài, còn ANN cho kết quả tốt hơn ở 2 mã chứng khoán Việt Nam.

#### 7.1.2 Dự đoán xu hướng

Về phần SVR cho xu hướng, với sự so sánh tương quan kết quả trong [4] và kiểm thử với nhiều period khác nhau, ta rút ra được một vài kết luận sau đây:

- So sánh với [4], kết quả kiểm thử phản ánh việc cài đặt đúng hướng. Tuy nhiên, có thể do đặc thù của mô hình mà kết quả cao nhất đạt được có bộ tham số khác với [4].

- Grid search phản ánh sự hiệu quả của nó, khi tìm được bộ tham số giúp nâng tỷ lệ chính xác lên 10 điểm phần trăm. Tuy nhiên, với cách cài đặt grid search như trên, có thể mô hình đã bỏ qua điểm tối ưu hơn do việc chọn bước nhảy delta (2 cho search thưa và 0.25 cho search kỹ). Pattern search trong nhiều trường hợp không cho kết quả tốt bằng grid search.
- Đối với period bằng 1, cách chọn dữ liệu là 5 node phản ánh return của 5 bộ 1 ngày trước không phản ánh được xu hướng, hiệu quả, dẫn đến kết quả (đồ thị) dự đoán không bám sát được với giá trị thực. Điều này ta cũng có thể dễ dàng nhận ra qua sự biến thiên quá đột ngột của return từng ngày.
- Đối với period từ 5 trở lên, dữ liệu đã có xu hướng hơn, nhưng do đầu vào chưa tạo được sự phản ánh phù hợp mà kết quả dự đoán chưa bám được vào đường xu hướng thật hay nói cách khác là chưa khớp (underfit).
- Cần có một công thức return phù hợp hơn với thị trường Việt Nam.

Về phần ANN, sự quá khớp làm cho kết quả dự đoán không ổn định:

- Đối với period nhỏ, mô hình tỏ ra không hiệu quả bằng SVR.
- Khi period = 30, hiệu quả đạt được lại cao hơn. Tuy nhiên, dễ nhận thấy đường đồ thị không có tính tổng quát cao, cách khá xa return thực.
- ANN mà cụ thể là Back propagation (BP) ANN – mạng lan truyền ngược bị hạn chế vì tín hiệu nhiễu, biến động dữ dội của dữ liệu chứng khoán. Khi training mạng lan truyền ngược trước những tín hiệu nhiễu của dữ liệu thường cho ra những kết quả dự đoán thường hay xuất hiện, giải pháp mà BP sử dụng thường tập trung vào sự tối ưu cục bộ cũng như kiến trúc của mạng (số lớp ẩn, số node từng lớp) thường phải qua thực nghiệm để quyết định, khó đưa ra tối ưu toàn cục, chịu sự tác động tăng thời gian xử lý khi dữ liệu lớn dần, trở nên phức tạp khi gặp vấn đề số chiều thay đổi.

### **7.1.3 Toàn cảnh**

- Thị trường Việt Nam chưa cho thấy sự ổn định trong tâm lý nhà đầu tư, dẫn đến sự không ổn định thể hiện qua giá chứng khoán.



- Độ chênh lệch giá giữa 2 ngày liên tiếp nhiều trường hợp bằng 0, dẫn đến khó đánh giá mô hình ở chu kỳ ngắn.
- Các phương pháp máy học phần nào thể hiện được sự hiệu quả đối với việc giải quyết bài toán kinh tế cụ thể. Tuy nhiên, cần có nhiều nghiên cứu hơn về cách xử lý dữ liệu cũng như việc chọn tham số để mô hình đạt hiệu quả cao.

## 7.2 Công việc tương lai

Từ kết luận trên, ta rút ra được một vài điều cần làm trong tương lai như sau:

- Định dạng dữ liệu đầu vào
  - Công thức return phù hợp cho thị trường Việt Nam.
  - Cân nhắc việc đưa vào một vài chỉ số kỹ thuật.
  - Việc bố trí số node đầu vào, định dữ liệu trong từng node đó
- Phương pháp huấn luyện
  - Cách huấn luyện cải tiến: để dự đoán giá của một ngày, ta luôn phải tiến hành train. Kích thước bộ train này tương đối nhỏ. Cách làm này rất hứa hẹn bởi những lý do sau:
    - Bên ANN đã có bài báo cho kết quả rất tốt.
    - Với kích thước bộ train nhỏ, ta có thể cải thiện đáng kể tốc độ thực thi của cross-validation và do đó, cải thiện tốc độ thực thi của các thuật toán tìm bộ tham số (ta có thể áp dụng Grid Search.)
    - Cách train này cũng giải quyết luôn vụ online vì rằng để dự đoán giá của một ngày mới, ta luôn phải tiến hành train lại từ đầu.
  - Tìm cách đặt trọng vào dữ liệu quan trọng hơn.
- Cân nhắc phương pháp chọn bộ tham số tối ưu
  - Tập trung cải tiến grid search cho xu hướng và pattern search cho dự đoán giá.
  - Xem xét đưa GA vào chọn bộ tham số.

➤ Xem xét về kiến trúc

- Nghiên cứu việc đưa ra mối tương quan giữa các node đầu vào, đặt trọng vào node quan trọng hơn. (Hướng tới việc đưa fuzzy vào mô hình)
- Giải quyết tình trạng overfit của ANN và underfit của SVR.

➤ Online

- Song song với việc cải tiến mô hình hiện tại, nhóm sẽ nghiên cứu một mô hình có khả năng thực hiện dự đoán online, không hẳn là chỉ thay đổi cách huấn luyện.

## PHỤ LỤC. HƯỚNG DẪN SỬ DỤNG

Giao diện của chương trình gồm 2 phần Thực Nghiệm (EXPERIMENT) và Ứng Dụng (APPLICATION) tương ứng thể hiện ở 2 tab:

The screenshot displays the 'STOCK PREDICTION' application window with the 'EXPERIMENT' tab selected. The interface is divided into several sections:

- SETTING:** Contains fields for 'Input file (\*.csv)' with a 'Browse...' button, 'Num input node' (set to 5), 'Preprocess' (set to 'ScaleByMinMax'), 'Num days predicted', 'Training Measure' (set to 'MSE'), and a list of 'Performance Measures' (MSE, NMSE, DM4Price, DS, RMSE). It also includes 'ANN SETTING' with fields for 'Num hidden node' (4), 'Learning rate' (0.3), 'Max loops' (2000), 'Bias' (0), 'Momentum' (0.01), and 'Accuracy (%)'. The 'SVR SETTING' section includes 'Model selection' (set to 'Grid search') and 'Num fold (1: leave one out)' (set to 10).
- PREDICTION TYPE:** Features radio buttons for 'Price Prediction' (selected) and 'Trend Prediction'.
- MODEL TYPE:** Features radio buttons for 'ANN' and 'SVR' (selected).
- TRADITIONAL TRAINING AND TEST:** A section with three sub-sections:
  - 1. DATA PREPARATION:** Includes 'Training set ratio (%)' (80) and an 'OK' button.
  - 2. TRAINING:** Includes 'Training set file' with a 'Browse...' button and a 'Train' button.
  - 3. TEST:** Includes 'Test set file' and 'Model file', both with 'Browse...' buttons, and a 'Test' button.
- STEP TRAINING AND TEST:** Includes 'Training size (days)' (30), 'From (d/M/yyyy)' (WholeData), and 'To' (WholeData), with a 'Step train and test' button.

### 7.3 PHẦN THỰC NGHIỆM (EXPERIMENT)

Giao diện chương trình gồm các thành phần chính: Lựa chọn bài toán dự đoán và mô hình dự đoán chứng khoán, thiết lập các thông số cho mô hình đã chọn (SETTING), cách thức tiến hành huấn luyện và kiểm thử theo kiểu truyền thống (TRADITIONAL TRAINING AND TEST) và Kiểu huấn luyện và kiểm thử theo phương pháp cải tiến từng bước (STEP TRAINING AND TEST).

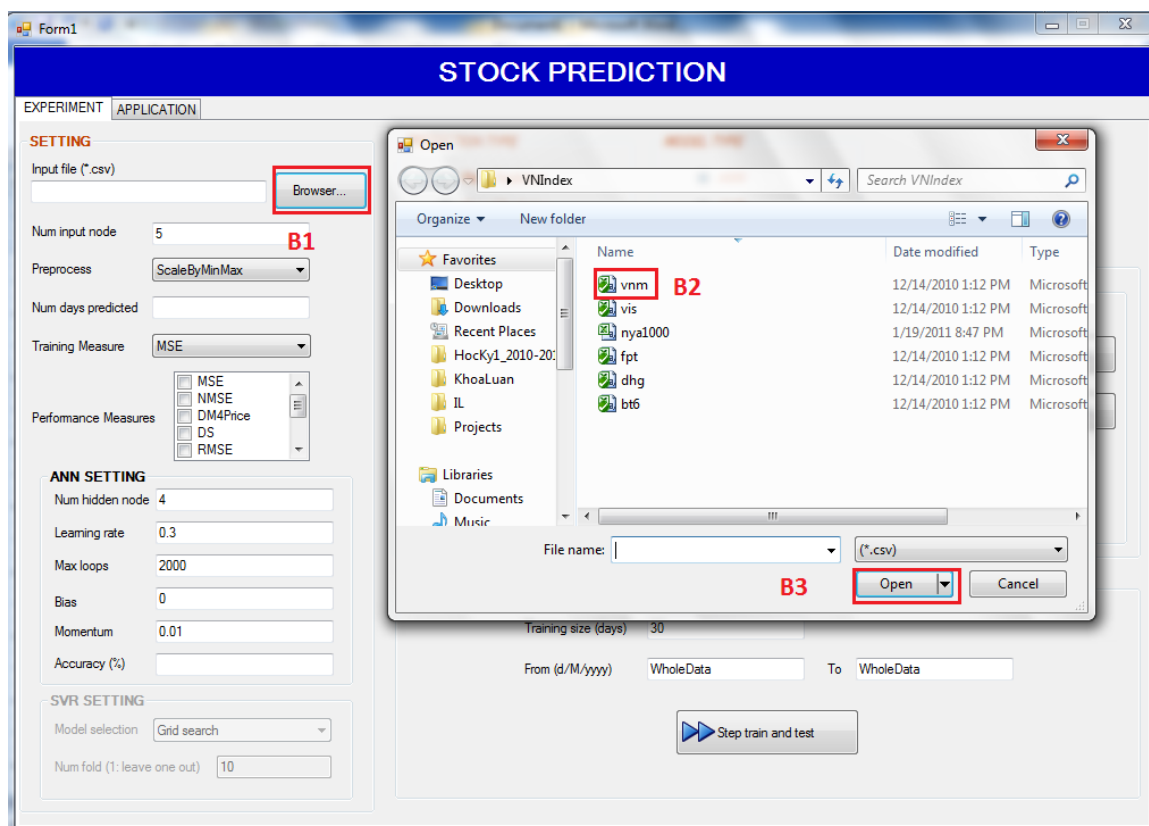
Đầu tiên bạn sẽ chọn lựa bài toán dự đoán là dự đoán xu hướng (Trend Prediction) hay giá (Price Prediction) và chọn lựa mô hình tương ứng

| PREDICTION TYPE                                   | MODEL TYPE                           |
|---|--------------------------------------|
| <input type="radio"/> Price Prediction            | <input checked="" type="radio"/> ANN |
| <input checked="" type="radio"/> Trend Prediction | <input type="radio"/> SVR            |

## Hình minh hoạ chọn bài toán dự đoán xu hướng với mô hình ANN

Cài đặt các thông số cho mô hình huấn luyện đã chọn (SETTING)

Bấm vào nút Browser... để mở file dữ liệu của mã chứng khoán cần cho huấn luyện và kiểm thử

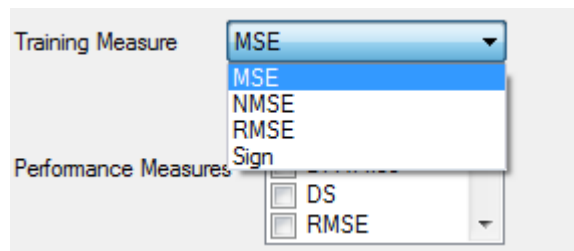


Mặc định của chương trình kiểm huấn luyện với số điểm dữ liệu đưa vào là 5 cho cả 2 mô hình ANN và SVR tuy nhiên bạn có thể thay đổi số điểm dữ liệu đưa vào bằng việc nhập số khác ở ô Num Input Node

Sau khi chọn xong dữ liệu và số node dữ liệu đầu vào, ta tiến hành thiết lập các thông số đầu vào cho các mô hình.

Chuẩn hoá dữ liệu về MinMax hoặc Return[-1,1], khi lựa chọn chuẩn hoá về dạng MinMax thì phần Num day predicted sẽ bị ẩn đi vì, còn nếu chọn Return[-1,1] thì sáng lên, mục đích của chọn như vậy là ứng với từng bài toán dự đoán là xu hướng hay giá.

Chọn lựa hàm tính độ lỗi khi huấn luyện cho mô hình đã chọn ở đây có các hàm tính độ lỗi: MSE, NMSE, RMSE, SIGN



Sau đó ta chọn hàm đánh giá độ chính xác khi dự đoán của mô hình ứng với từng bài toán dự đoán xu hướng hay giá khi thực hiện kiểm tra mô hình ở bước kiểm thử sau quá trình huấn luyện, các hàm đánh giá gồm: MSE, NMSE, DM4Price, DS, RMSE, APE, MAPE, WDS, SIGN

Thiết lập các thông số mô hình ANN hoặc SVR:

Đối với mô hình ANN, thì bạn cần thiết lập các thông số sau:

| ANN SETTING     |      |
|-----------------|------|
| Num hidden node | 4    |
| Learning rate   | 0.3  |
| Max loops       | 2000 |
| Bias            | 0    |
| Momentum        | 0.01 |
| Accuracy (%)    |      |

Trong đó:

Num hidden node: là số node ở tầng ẩn (Hidden Layer), số node tầng nhập (Input Layer) phụ thuộc vào bạn chọn Num Input Node ở trên đã đề cập. và mặc định tầng xuất (Output Layer) là 1.

Learning Rate: là hệ số học của mạng

Max loops: số vòng lặp tối đa khi mạng không đạt được ngưỡng đặt ra, dùng để dừng quá trình huấn luyện ở số vòng lặp xác định

Bias và Momentum: là hệ số thêm vào mạng (có thể để mặc định)

Accuracy: Độ chính xác mong muốn khi thực hiện huấn luyện mạng

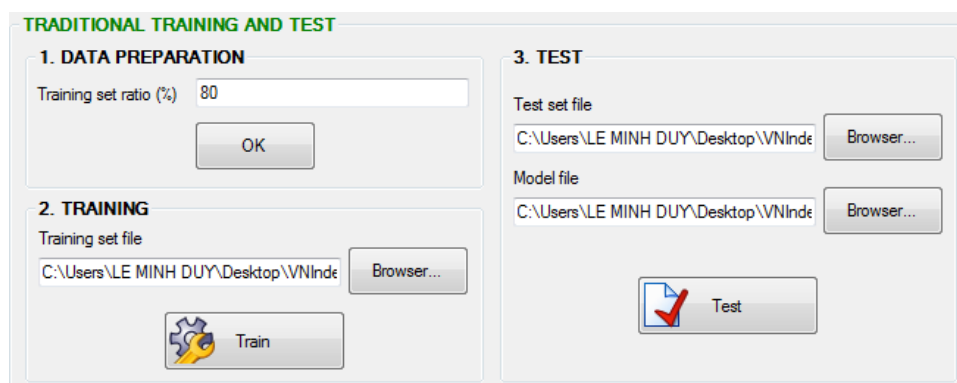
Đối với mô hình SVR, thì bạn cần thiết lập các thông số sau:



The image shows a dialog box titled "SVR SETTING". It contains two main settings: "Model selection" which is a dropdown menu currently showing "Grid search", and "Num fold (1: leave one out)" which is a text input field containing the number "10".

Với thông số Model selection để xác định chọn cách thức tìm các tham số thích hợp của SVR (C, Gama, epxilon) theo thuật toán nào. Ở đây có 3 lựa chọn grid search, pattern search hay sử dụng giá trị mặc định Use Default Value và chọn số fold.

Huấn luyện và kiểm thử mô hình theo cách truyền thống (TRADITIONAL TRAINING AND TEST)



The image shows a dialog box titled "TRADITIONAL TRAINING AND TEST". It is divided into three sections: 1. DATA PREPARATION with a "Training set ratio (%)" input field set to "80" and an "OK" button; 2. TRAINING with a "Training set file" input field showing a file path and a "Browse..." button, and a "Train" button with a gear icon; 3. TEST with "Test set file" and "Model file" input fields, each with a "Browse..." button, and a "Test" button with a checkmark icon.

## Chuẩn bị dữ liệu (DATA PREPARATION)

Chọn cách phân chia tỉ lệ dữ liệu huấn luyện tỉ lệ (%) có ý nghĩa là mình sẽ dành bao nhiêu phần dữ liệu từ cho Huấn luyện và còn lại cho test. Và bấm nút Ok để thực hiện phân chia dữ liệu và tiền xử lý luôn( lưu ý cách thức tiền xử lý đã thiết lập ở SETTING). Khi thực hiện bước này sẽ cho ra 2 tập tin tương ứng là \*\_train.txt và \*\_test.txt.

## Thực hiện Huấn luyện (TRAINING)

Chọn tập tin cần huấn luyện sau khi đã được tiền xử lý và phân chia dữ liệu để huấn luyện, sau đó bấm nút Train để thực hiện huấn luyện mạng. Thông thường chương trình tự điền vào tên file khi có file csv ban đầu. Tuy nhiên, cần chọn lại vì đối với xu hướng, tên file có thể thay đổi. Sau bước này chúng ta sẽ có được mô hình học và được xuất ra các tập tin \*\_Model.txt.

## Thực hiện kiểm thử (TEST)

Bấm nút browser... ở phần Test set file để chọn tập tin kiểm thử mô hình

Bấm nút browser... ở phần Model set file để chọn tập tin mô hình đã có ở bước Huấn luyện

Sau đó bấm nút Test để thực hiện kiểm thử. Quá trình kiểm thử này với các hàm đánh giá mô hình huấn luyện đã được chọn ở bước SETTING nếu chưa chọn hàm đánh giá thì phải chọn để biết kết quả. Kết quả đánh giá sẽ xuất ra tập tin tên *PerformanceMeasure.txt* và giá trị dự đoán ở tập tin \*\_Predict.txt.

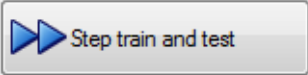
**Lưu ý:** Không nhất thiết phải thực hiện từng bước ngay từ Data preperation. Ta có thể train ngay một file nếu đã được tiền xử lý trước đó. Hoặc có thể test với file test và model đã có miễn là điền đủ các thông tin Setting phù hợp.

## Huấn luyện và kiểm thử theo phương pháp cải tiến từng bước (STEP TRAINING AND TEST)

**STEP TRAINING AND TEST**

Training size (days)

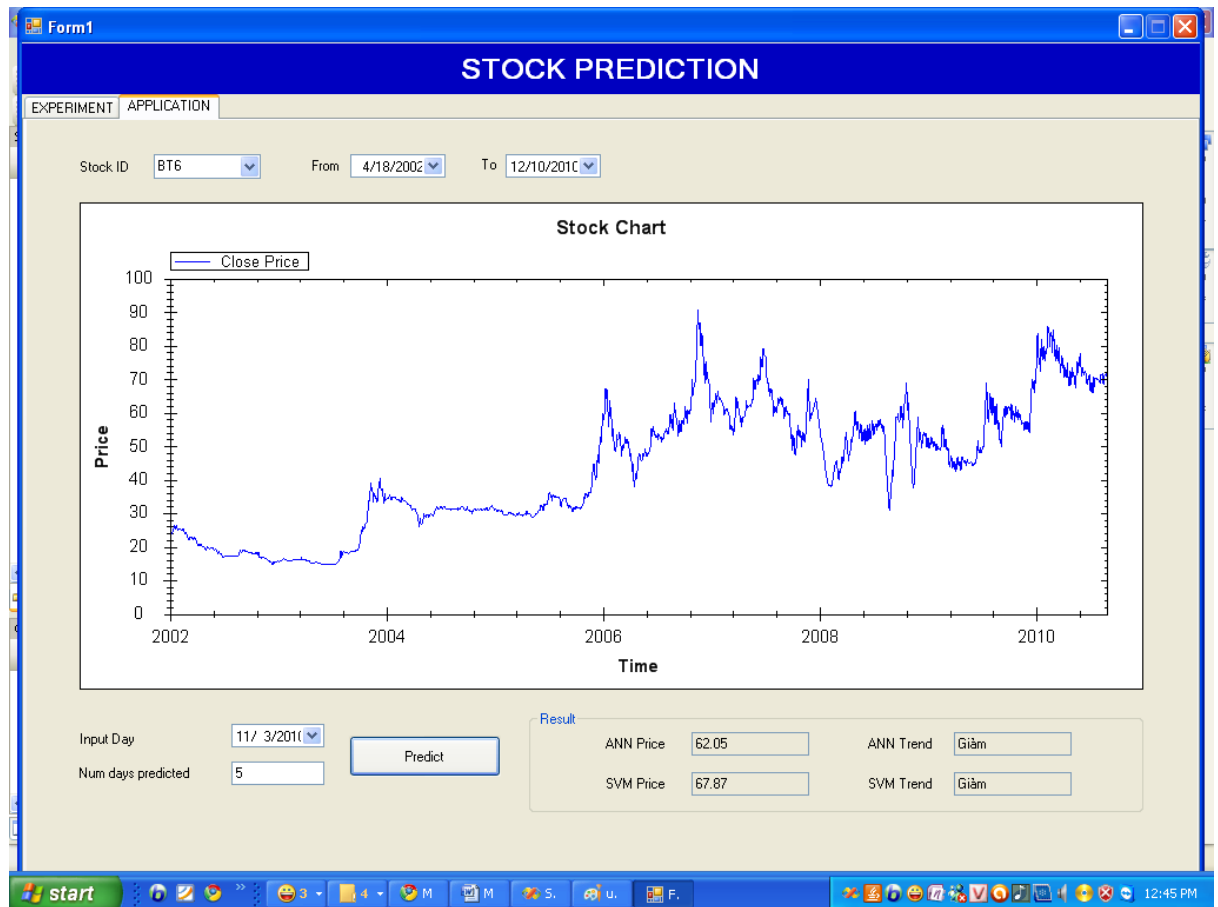
From (d/M/yyyy)  To

 Step train and test

Để thực hiện huấn luyện và kiểm thử theo phương pháp từng bước này thì cần thiết lập thông số về số ngày để train trong mỗi bước. Xác định khoảng thời gian muốn thực hiện huấn luyện và kiểm thử trong khung nhập From và To (thời gian bắt đầu lấy dữ liệu để huấn luyện – kiểm thử và thời gian kết thúc). Nếu để mặc định, chương trình sẽ chạy hết bộ dữ liệu. Sau đó bấm nút Step train and test để thực hiện.



## 7.4 PHẦN ỨNG DỤNG (APPLICATION)



Chọn ngày “hiện tại” ở Input day, ngày này mang ý nghĩa là dữ liệu sẽ được lấy trước đó rồi dự đoán cho ngày kế tiếp.

Đối với dự đoán giá, kết quả luôn là giá ngày kế tiếp. Còn với xu hướng, ta cần xác định số ngày muốn dự đoán.

## TÀI LIỆU THAM KHẢO

- [1] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer (2007)
- [2] Shukuan Lin; Shaomin Zhang; Jianzhong Qiao; Hualei Liu; Ge Yu, *A Parameter Choosing Method of SVR for Time Series Prediction*, IEEE, 2008
- [3] Prapaphan Pan-0, *A stock price prediction model by the neural network approach*, Master thesis, 2003
- [4] Shiyi Chen, Kiho Jeong, Wolfgang K. Hardle, *Recurrent Support Vector Regression for a Nonlinear ARMA Model with Applications to Forecasting Financial Returns*, SFB 649 Discussion Paper, 2008
- [5] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a Library for Support Vector Machines*