Dự đoán giá cổ phiếu trên thị trường chứng khoán Việt Nam bằng phương pháp lai GA-SVR

A Hybrid GA-SVR Approach for Vietnam Stock Price Prediction

Trần Trung Kiên, Bành Trí Thành, Nguyễn Hoàng Tú Anh

Abstract: *Stock price prediction is an interesting problem that has attracted much attention from both investors and researchers. There are, however, not many researchs in this field with Vietnam stock market because this market is still nascent and high non-stationary. In this paper, we propose a hybrid approach, which integrates Genetic Algorithm (GA) with Support Vector Regression (SVR) to predict Vietnam stock price. In this approach, GA solves two problems simultaneously: finding SVR's optimal parameters and feature selection. Then, SVR's optimal parameters and selected features serve as input for training SVR model. Our experimental results show that the hybrid GA-SVR approach outperforms SVR, Artificial Neural Network (ANN) and can be used in practice to gain profit.*

I. GIỚI THIỆU

Dự đoán giá cổ phiếu là một bài toán thú vị thu hút được sự quan tâm của cả các nhà nghiên cứu lẫn các nhà đầu tư. Tuy nhiên, đây cũng là một bài toán rất khó bởi lẽ giá chứng khoán thường rất phức tạp và nhiễu loạn [8]. Đã có nhiều cố gắng dự đoán thị trường tài chính bằng phương pháp phân tích truyền thống cho đến kỹ thuật trí tuệ nhân tạo như logic mờ và đặc biệt là mạng nơ ron nhân tạo (ANN)[1]. ANN là kỹ thuật được sử dụng nhiều trong lĩnh vực này bởi nó có thể mô tả được mối quan hệ phi tuyến giữa đầu vào với đầu ra. Tuy nhiên, nhược điểm của ANN là dễ bị bẫy bởi cực trị cục bộ. Bên cạnh đó, ANN có số lượng tham số tự do lớn và thường phải chọn bằng phương pháp thử và sai [19].

Gần đây, cộng đồng nghiên cứu có xu hướng tập trung vào một kỹ thuật mới: hồi qui véc tơ hỗ trợ (Support Vector Regression - SVR) [3]. Nguồn gốc của SVR là máy véc tơ hỗ trợ (Support Vector Machine - SVM) [3]. SVM ban đầu được dùng cho bài toán phân lớp, về sau mở rộng cho bài toán hồi qui và gọi là SVR. Nhiều nghiên cứu gần đây cho thấy SVR cho kết quả tốt hơn ANN trong bài toán dự đoán giá cổ phiếu [8]. Đó là do SVR sử dụng nguyên lý tối thiểu hóa rủi ro cấu trúc nên có khả năng tổng quát hóa cao hơn ANN. Ngoài ra, số lượng tham số tự do của SVR cũng ít hơn so với ANN [8].

Khi sử dụng SVR, ta cần giải quyết hai vấn đề: xác định bộ tham số tối ưu cho SVR và chọn lựa các đặc trưng đầu vào. Trong bài toán dự đoán giá cổ phiếu, việc chọn lựa các đặc trưng đầu vào đóng vai trò rất quan trọng. Các đặc trưng đầu vào thường là chỉ số phân tích kỹ thuật. Hiện nay có khá nhiều chỉ số phân tích kỹ thuật (khoảng hơn 100), việc lựa chọn chỉ số phù hợp cho từng mã cổ phiếu là không đơn giản do chỉ số này có thể tốt cho cổ phiếu A nhưng chưa chắc đã tốt cho cổ phiếu B [13]. Rõ ràng, ta cần xây dựng một chiến lược lựa chọn các chỉ số quan trọng tương ứng với một mã cổ phiếu cụ thể.

Để chọn đặc trưng đầu vào trong bài toán dự đoán giá cổ phiếu, Ince và Trafalis [13] sử dụng kỹ thuật phân tích thành phần chính (PCA). Huang và Wu [11] sử dụng GA. Huang và Tsai [9] dùng hệ số quyết định

r^2. Chee [14] đề xuất phương pháp lai giữa F-Score và F_SSFS. Ý tưởng dùng GA để chọn lựa đặc trưng đầu vào cho SVM cũng đã được đề xuất trong một số bài toán áp dụng trên các loại dữ liệu khác [2], [12].

Việc xác định bộ tham số tối ưu cho SVR cũng quan trọng không kém bởi bộ tham số này sẽ ảnh hưởng đến độ chính xác dự đoán của mô hình SVR. Người ta thường sử dụng thuật toán Grid Search [7] để xác định bộ tham số tối ưu cho SVR. Tuy nhiên, thuật toán này tốn thời gian và hiệu quả không cao [10]. Nhằm nâng cao hiệu quả, Chen và Ho [5], Zhu và Wang [19] sử dụng GA để xác định bộ tham số SVR.

Nhìn chung, các nghiên cứu trên chỉ tập trung vào giải quyết một trong hai vấn đề đã nêu của SVR. Chẳng hạn, các tác giả [12] đề xuất mô hình kết hợp giữa GA và SVM, trong đó GA được dùng để chọn lựa các đặc trưng đầu vào, còn các tham số SVM được chọn cố định. Còn [5] kết hợp GA và SVR, trong đó GA được dùng để xác định bộ tham số tối ưu của SVR, các đặc trưng đầu vào được chọn bằng phương pháp thử và sai.

Ngoài ra, các thị trường chứng khoán được thử nghiệm nhiều nhất là Mỹ và Trung Quốc. Với thị trường chứng khoán Việt Nam, hiện tại có khá ít các nghiên cứu áp dụng kỹ thuật máy học để dự đoán bởi vì thị trường này vẫn còn non trẻ và kém ổn định.

Trong bài báo này, chúng tôi đề xuất phương pháp lai GA-SVR để dự đoán giá cổ phiếu ở thị trường chứng khoán Việt Nam. Trong phương pháp lai này, GA thực hiện đồng thời hai nhiệm vụ: xác địnhbộ tham số tối ưu của SVR và lựa chọn các chỉ số kỹ thuật quan trọng nhất để thiết lập đầu vào. Sau đó, bộ tham số tối ưu và các chỉ số kỹ thuật được chọn sẽ được huấn luyện với SVR để cho ra mô hình dự đoán.

Các phần tiếp theo được trình bày như sau: phần II trình bày các lý thuyết nền tảng, phần III trình bày phương pháp đề xuất, phần IV trình bày kết quả thử nghiệm và cuối cùng là kết luận.

II. LÝ THUYẾT NỀN TẢNG

1. SVR và các tham số của SVR[3]

Ý tưởng cơ bản của SVR là ánh xạ phi tuyến tập dữ liệu $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subset R^n \times R$ sang một không gian đặc trưng nhiều chiều mà ở đó có thể sử dụng phương pháp hồi qui tuyến tính. Đặc điểm của SVR là khi xây dựng hàm hồi qui ta không cần sử dụng hết tất cả các điểm dữ liệu trong tập huấn luyện. Những điểm dữ liệu có đóng góp vào việc xây dựng hàm hồi qui được gọi là những vectơ hỗ trợ.

Hàm hồi qui của SVR như sau:

$$f(x) = w^T \phi(x) + b \quad (1)$$

Trong đó, $w \in R^m$ là véc tơ trọng số, $b \in R$ là hằng số, $x \in R^n$ là véc tơ đầu vào, $\phi(x) \in R^m$ là véc tơ đặc trưng.

Để tìm w và b, SVR giải quyết bài toán tối ưu hóa sau:

Cực tiểu hóa hàm:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \quad (2)$$

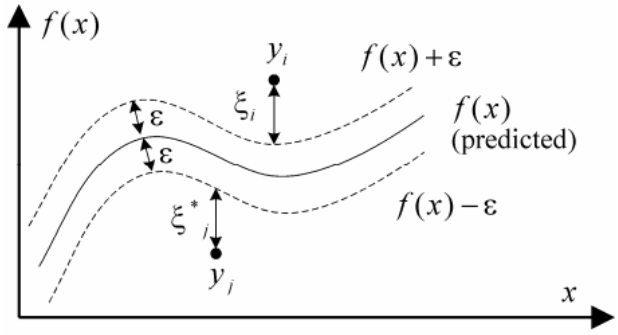
Với các ràng buộc:

$$\begin{cases} y_i - f(x_i) \le \varepsilon + \xi_i \\ f(x_i) - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

Với i = 1, 2, …, N

Trong đó, C là hằng số chuẩn hóa đóng vai trò cân bằng giữa độ lỗi huấn luyện và độ phức tạp mô hình.

Hình 1 minh họa SVR với hàm lỗi ε-insensitive. Đường nét liền ở giữa ứng với đường dự đoán. Giá trị ε xác định độ rộng của ống bao quanh đường dự đoán. Nếu giá trị đích y_i nằm trong ống này thì coi như độ lỗi bằng 0. Nếu giá trị đích y_i nằm ngoài ống này thì độ lỗi bằng ξ_i (nếu y_i nằm ngoài phía trên ống) hoặc ξ_i^* (nếu y_i nằm ngoài phía dưới ống)



Hình 1. Minh họa hàm lỗi của thuật toán SVR [16]

Từ (2) dùng hàm Lagrange và điều kiện Karush-Kuhn-Tucker, ta có bài toán tối ưu hóa tương đương:

Cực đại hóa:

$$-\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i) - \varepsilon\sum_{i=1}^{N}(\alpha_i^* + \alpha_i) \quad (3)$$

Với các ràng buộc:

$$\begin{cases} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ 0 \le \alpha_i, \alpha_i^* \le C, i = 1, \dots, N \end{cases}$$

Trong đó, các nhân tử Lagrange α_i, α_i^* phải thỏa $\alpha_i\alpha_i^* = 0$. Véc tơ trọng tối ưu sẽ có dạng: $w = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\phi(x_i)$. Từ đây, ta có hàm hồi qui của SVR:

$$f(x) = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)K(x, x_i) + b \quad (4)$$

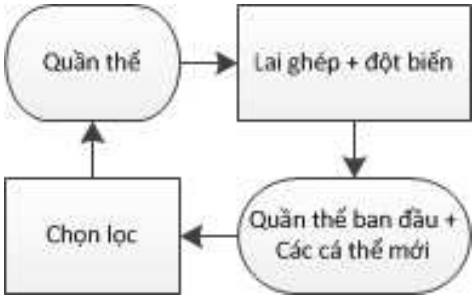
Trong đó, $K(x_i, x_j)$ được gọi là hàm nhân và có giá trị bằng tích vô hướng của hai véc tơ đặc trưng $\phi(x_i), \phi(x_j)$. Bất kỳ một hàm nào thỏa điều kiện Mercer thì đều có thể được dùng làm hàm nhân. Hàm nhân được sử dụng phổ biến nhất là hàm Gaussian:

$$K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2) \quad (5)$$

Như vậy, với SVR sử dụng hàm lỗi ε-insensitive và hàm nhân Gaussian ta có ba tham số cần tìm: hệ số chuẩn hóa C, tham số γ của hàm nhân Gaussian và độ rộng của ống ε. Cả ba tham số này đều ảnh hưởng đến độ chính xác dự đoán của mô hình và cần phải chọn lựa kỹ càng. Nếu C quá lớn thì sẽ ưu tiên vào phần độ lỗi huấn luyện, dẫn đến mô hình phức tạp, dễ bị quá khớp. Còn nếu C quá nhỏ thì lại ưu tiên vào phần độ phức tạp mô hình, dẫn đến mô hình quá đơn giản, giảm độ chính xác dự đoán. Ý nghĩa của ε cũng tương tự C. Nếu ε quá lớn thì có ít vectơ hỗ trợ, làm cho mô hình quá đơn giản. Ngược lại, nếu ε quá nhỏ thì có nhiều vectơ hỗ trợ, dẫn đến mô hình phức tạp, dễ bị quá khớp. Tham số γ phản ánh mối tương quan giữa các vectơ hỗ trợ nên cũng ảnh hưởng đến độ chính xác dự đoán của mô hình.

2. Thuật giải di truyền (GA) [6]

Thuật giải di truyền là một thuật toán tìm kiếm giải pháp tối ưu dựa trên nguyên lý chọn lọc tự nhiên của Darwin và cơ chế di truyền trong sinh học. GA làm việc với một tập các giải pháp, được gọi là quần thể; mỗi giải pháp được gọi là cá thể và được diễn bằng một nhiễm sắc thể (chuỗi bit). Tương tự như quá trình tiến hóa trong tự nhiên, ở mỗi vòng lặp ta có ba hoạt động: lai ghép (crossover), đột biến (mutation) và chọn lọc (selection). Trong đó, lai ghép là quá trình hai nhiễm sắc thể cha mẹ tạo ra hai nhiễm sắc thể con bằng cách trao đổi một đoạn gene ngẫu nhiên cho nhau.Bằng cách này, ta tạo ra được những cá thể mới và do đó, mở rộng vùng không gian tìm kiếm. Đột biến đơn giản là sự thay đổi một bit nào đó trong chuỗi bit nhiễm sắc thể từ 0 thành 1 hoặc từ 1 thành 0. Điều này giúp thuật toán có thể nhảy ra khỏi vùng tối ưu cục bộ. Cuối cùng, chọn lọc giúp giữ lại những cá thể tốt nhất. Mỗi cá thể cần có một giá trị đi kèm gọi là độ thích nghi. Độ thích nghi này được định nghĩa tùy theo từng bài toán cụ thể.



Hình 2. Vòng lặp GA

Hình 2 minh họa cho vòng lặp tiến hóa của GA. Thuật toán sẽ dừng sau một số vòng lặp xác định trước hoặc khi thỏa điều kiện dừng nào đó.

3. Tìm các tham số SVR với Grid Search [7]

Như đã trình bày ở trên, với SVR sử dụng hàm lỗi ε-insensitive và hàm nhân Gaussian ta có 3 tham số cần tìm: hệ số chuẩn hóa C, tham số γ của hàm nhân Gaussian và độ rộng của ống ε. Cách phổ biến để tìm 3 tham số này là dùng Grid Search kết hợp với đánh giá chéo (k-fold crossvalidation). Grid Search đơn giản là phương pháp thử các bộ (C, γ, ε) khác nhau và chọn ra bộ cho độ lỗi đánh giá chéo nhỏ nhất. Người ta thường dùng phương pháp tăng dần theo số mũ. Chẳng hạn C = 2^{-6}, 2^{-5}, …, 2^{8}; γ = 2^{-8}, 2^{-7}, …, 2^{6}; ε = 2^{-11}, 2^{-10}, …, 2^{-1}. Như vậy, C có 15 giá trị, γ có 15 giá trị, ε có 11 giá trị. Tổng cộng ta phải thử 15×15×11 = 2475 lần với đánh giá chéo.

Do tiến hành Grid Search như vậy sẽ rất tốn thời gian nên thông thường Grid Search được chia làm 2 bước: bước một tìm kiếm với một lưới thưa (chẳng hạn C = 2^{-6}, 2^{-4}, …, 2^{8}; γ = 2^{-8}, 2^{-6}, …, 2^{6}; ε = 2^{-11}, 2^{-9}, …, 2^{-1}. Như vậy số lần thử chỉ còn 8×8×6 = 384). Sau khi đã tìm được một bộ tham số tốt nhất, bước hai tìm kiếm với một lưới dày hơn ở vùng lận cận của bộ tham số tốt nhất này.

Ở đây, việc đánh giá chéo được thực hiện trên tập huấn luyện. Sau khi đã tìm ra được bộ tham số tốt nhất bằng Grid Search, bộ tham số này được dùng để huấn huyện SVR với toàn bộ tập huấn luyện và cho ra mô hình dự đoán cuối cùng.

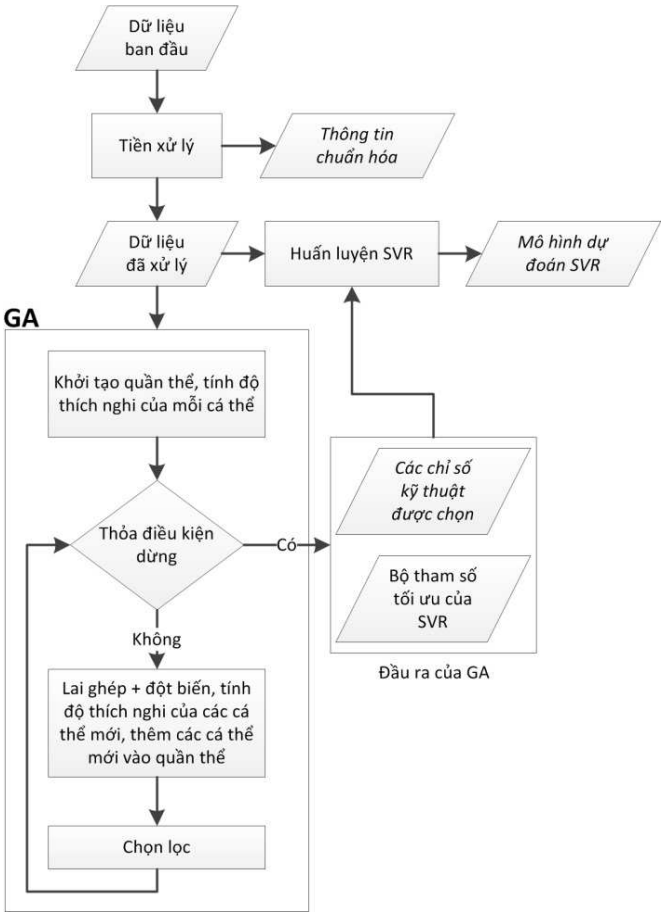
III. DỰ ĐOÁN GIÁ CỔ PHIẾU VỚI PHƯƠNG PHÁP LAI GA-SVR

Trong phần này, chúng tôi trình bày phương pháp lai GA-SVR đề xuất áp dụng cho bài toán dự đoán giá cổ phiếu. Trong phương pháp này, đầu tiên GA được dùng để tìm bộ tham số tối ưu cho SVR và chọn lựa các đặc trưng đầu vào (các chỉ số kỹ thuật). Sau đó, bộ tham số tối ưu của SVR và các đặc trưng đầu vào tìm được sẽ dùng để huấn luyện SVR và cho ra mô hình dự đoán.

Hệ thống của chúng tôi gồm có hai phần chính: module huấn luyện và module dự đoán.

1. Module huấn luyện

Hình 3 mô tả module huấn luyện. Một cách tổng quan nhất, đầu vào của module này là dữ liệu ban đầu, kết quả đầu ra gồm có 3 thành phần: thông tin chuẩn hóa, các chỉ số kỹ thuật được chọn và mô hình dự đoán SVR.



Hình 3. Module huấn luyện

Đầu tiên, từ dữ liệu ban đầu gồm giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa và khối lượng giao dịch, hệ thống tiến hành tiền xử lý dữ liệu. Bước

tiền xử này bao gồm tính toán các chỉ số kỹ thuật, thiết lập đầu vào, đầu ra và chuẩn hóa dữ liệu. Kết quả của quá trình tiền xử lý là dữ liệu đã xử lý và thông tin chuẩn hóa. Thông tin chuẩn hóa này sẽ được dùng trong module dự đoán.

Sau đó, dữ liệu đã xử lý sẽ đưa vào GA. Kết quả đầu ra của GA gồm có các chỉ số kỹ thuật được chọn và bộ tham số tối ưu của SVR. Cuối cùng, chúng sẽ dùng để huấn luyện SVR và cho ra mô hình dự đoán.

Phần dưới đây sẽ trình bày chi tiết về bước tiền xử lý, cách biểu diễn nhiễm sắc thể và qui trình tính độ thích nghi của nhiễm sắc thể.

a. Tiền xử lý

Bước tiền xử lý gồm có hai phần: thiết lập đầu vào, đầu ra và chuẩn hóa dữ liệu.

Thiết lập đầu vào, đầu ra:

Đầu vào của hệ thống bao gồm các chỉ số phân tích kỹ thuật sau: Giá đóng cửa**,** Bollinger Bands (20, 2) với Middle Band, Upper Band và Lower Band, EMA(5), MACD(12, 26, 9) với giá trị của MACD và Signal Line, RSI(7), ROC-1, ROC-2, ROC-3, ROC-4, ROC-5. Tất cả tạo thành véc tơ đầu vào 13 chiều. Đây là các chỉ số thường được sử dụng trong phân tích kỹ thuật. Chi tiết về các chỉ số này được trình bày ở phần Phụ lục.

Về đầu ra, ta có thể chọn đầu ra là giá đóng cửa của ngày kế tiếp. Tuy nhiên, theo [15] việc chọn đầu ra là ROC+1 (Rate Of Change) sẽ cho kết quả dự đoán tốt hơn so với việc chọn đầu ra là giá đóng cửa. Giá trị ROC+1 cho ta biết giá đóng cửa ngày mai tăng hay giảm bao nhiêu % so với giá đóng cửa ngày hôm nay. Hệ thống sử dụng ROC+1 là kết quả đầu ra. Công thức tính của ROC+1 như sau:

$$\text{ROC+1} = \frac{C_{t+1} - C_t}{C_t} \times 100 \qquad (6)$$

Trong đó, C_t là giá đóng cửa của ngày thứ t và C_{t+1} là giá đóng cửa của ngày thứ t+1.

Chuẩn hóa dữ liệu:

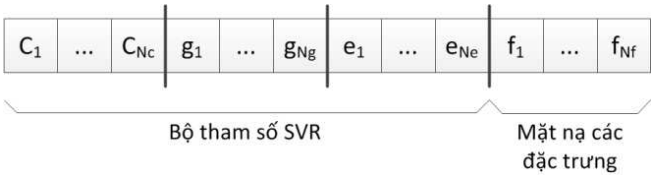
Tất cả các đặc trưng đầu vào được chuẩn hóa về [0, 1] theo công thức:

$$x'_a = \frac{x_a - min_a}{max_a - min_a} \qquad (7)$$

Trong đó, x_a là giá trị ban đầu của đặc trưng a, min_a là nhỏ trị nhỏ nhất của đặc trưng a, max_a là giá trị lớn nhất của đặc trưng a và x'_a là giá trị sau khi chuẩn hóa của đặc trưng a. Hai lợi ích chính của việc chuẩn hóa này là các đặc trưng có miền giá trị lớn không lấn át các đặc trưng có miền giá trị nhỏ và tránh gặp phải các khó khăn trong quá trình tính toán [7]. Thông tin chuẩn hóa (min_a, max_a) sẽ được lưu để dùng khi tiến hành dự đoán với đầu vào mới.

b. Biểu diễn nhiễm sắc thể

Trong phương pháp lai GA-SVR đề xuất, GA làm đồng thời hai việc: tìm các tham số tối ưu của SVR và chọn các đặc trưng đầu vào. Với SVR sử dụng hàm lỗi ε-insensitive và hàm nhân Gaussian ta có 3 tham số cần tìm: hệ số chuẩn hóa C, tham số γ của hàm nhân Gaussian và độ rộng của ống ε. Như vậy, một nhiễm sắc thể bao gồm 4 thành phần: C, γ, ε và mặt nạ các đặc trưng. Mỗi nhiễm sắc thể sẽ được biểu diễn bằng một chuỗi bit. Hình 4 minh họa cấu trúc nhiễm sắc thể, trong đó 3 phần đầu ứng với bộ tham số SVR và phần cuối ứng là mặt nạ các đặc trưng.



Hình 4. Cấu trúc nhiễm sắc thể

Phần bộ tham số SVR:

Trong Hình 4, đoạn bit từ C_1 đến C_{Nc} biểu diễn giá trị của C, từ g_1 đến g_{Ng} biểu diễn giá trị của γ, từ e_1 đến e_{Ne} biểu diễn giá trị của ε. Nc, Ng, Ne lần lượt là số bit cần dùng để biểu diễn C, γ, ε.

Từ chuỗi bit ứng với C, giá trị của C được tính theo công thức:

$$C = min_C + \frac{max_C - min_C}{2^{Nc}-1} \times d_C \quad (8)$$

Trong đó d_C là giá trị thập phân của chuỗi bit ứng với C. Cách tính γ , ε hoàn toàn tương tự.

Phần mặt nạ các đặc trưng:

Số bit của phần này luôn bằng với số đặc trưng đầu vào, trong đó ta qui ước: bit 1 ứng với đặc trưng được chọn, bit 0 ứng với đặc trưng không được chọn.

c. Qui trình tính độ thích nghi

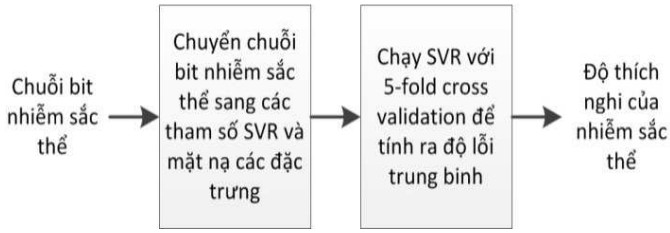
Qui trình tính độ thích nghi dùng để đánh giá một nhiễm sắc thể là tốt hay xấu. Đầu vào của qui trình này là chuỗi bit nhiễm sắc thể và kết quả đầu ra là độ thích nghi của nhiễm sắc thể đó. Nhiễm sắc thể có độ thích nghi càng lớn thì càng tốt, càng có nhiều cơ hội được giữ lại thông qua quá trình chọn lọc.

Hình 5 mô tả qui trình tính độ thích nghi. Đầu tiên, chuỗi bit nhiễm sắc thể sẽ chuyển sang các tham số SVR và mặt nạ đặc trưng. Dựa vào mặt nạ đặc trưng, ta thiết lập tập huấn luyện với đầu vào bao gồm các đặc trưng được chọn. Kế đến, tập huấn luyện này và các tham số SVR sẽ dùng để chạy SVR với 5-fold cross validation. Hình 6 mô tả quá trình chạy SVR với 5-fold cross validation. Tập huấn luyện được chia làm 5 phần bằng nhau. Sau đó, cứ lần lượt 4 phần dùng để huấn luyện, 1 phần còn lại dùng để thử nghiệm. Khi đó, ta có hàm tính độ thích nghi như sau:

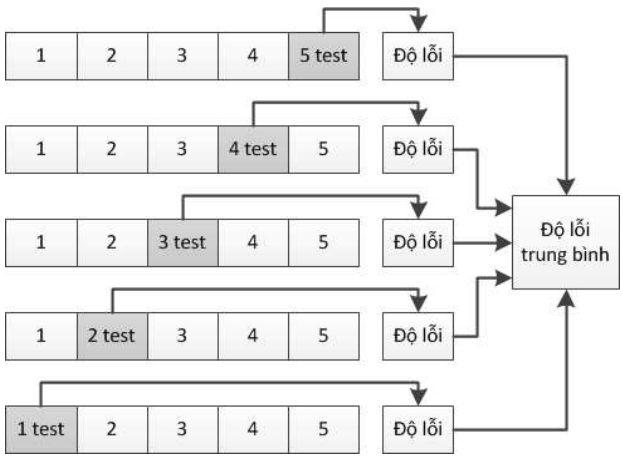
$$f(x) = -\frac{1}{N}\sum_{n=1}^{N}[p_n(x) - a_n]^2 \quad (9)$$

Trong đó: x là nhiễm sắc thể, N là số mẫu của tập huấn luyện, a_n là giá trị thật, $p_n(x)$ là giá trị dự đoán có được thông qua quá trình chạy SVR với 5-fold cross validation (ứng với bộ tham số SVR và các đặc trưng được chọn có được từ nhiễm sắc thể x)

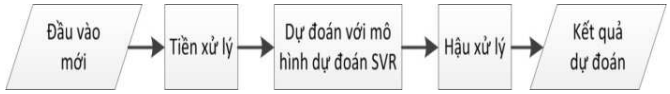
Vì mỗi lần tính độ thích nghi phải chạy 5-fold cross validation nên quá trình chạy GA sẽ tốn nhiều thời gian. Để tăng tốc, chúng tôi lưu lại danh sách các nhiễm sắc thể đã từng tính độ thích nghi. Khi đưa một nhiễm sắc thể vào tính độ thích nghi, trước hết hệ thống kiểm tra nhiễm sắc thể đó có nằm trong danh sách này hay không, nếu có thì dùng lại độ thích nghi đã tính mà không cần chạy cross validation nữa.



Hình 5. Qui trình tính độ thích nghi



Hình 6. Qui trình chạy SVR với 5-fold cross validation



Hình 7. Module dự đoán

2. Module dự đoán

Sau quá trình huấn luyện, thu được thông tin chuẩn hóa, các chỉ số kỹ thuật được chọn và mô hình dự đoán SVR. Hình 7 mô tả module dự đoán.

Trước tiên, giá trị đầu vào mới sẽ qua bước tiền xử lý gồm hai công việc:

- Thiết lập lại đầu vào dựa vào các chỉ số kỹ thuật được chọn.

- Chuẩn hóa đầu vào mới dựa vào thông tin chuẩn hóa.

Đầu vào sau khi tiền xử lý được đưa vào mô hình dự đoán SVR. Kết quả dự đoán của mô hình SVR là giá trị ROC+1 được chuyển sang giá đóng cửa ở bước hậu xử lý và cho ra kết quả dự đoán cuối cùng.

IV. KẾT QUẢ THỬ NGHIỆM

1.Mô tả dữ liệu

Bảng 1. Mô tả dữ liệu

Mã	Công ty phát hành	Nhóm ngành	Số ngày giao dịch
ITA	Công ty cổ phần đầu tư công nghiệp Tân Tạo	Bất động sản	996
SAM	Công ty cổ phần đầu tư và phát triển Sacom	Công nghệ và thiết bị viễn thông	994
VIP	Công ty cổ phần vận tải xăng dầu Vipco	Vận tải	994

Chúng tôi tiến hành thử nghiệm trên 3 mã cổ phiếu của sàn giao dịch TP Hồ Chí Minh[1]. Ba mã cổ phiếu này đại diện cho 3 nhóm ngành khác nhau. Cả ba mã đều được lấy từ ngày 2/1/2007 đến ngày 31/12/2010, bao gồm khoảng gần 1000 ngày giao dịch. Chi tiết về dữ liệu được trình bày ở Bảng 1.

Sau khi tiền xử lý, bộ dữ liệu được chia thành 2 tập là tập huấn luyện và tập thử nghiệm, trong đó tập thử nghiệm bao gồm 100 ngày giao dịch gần đây nhất.

2. Các độ đo chất lượng dự đoán

Chúng tôi sử dụng hai độ đo là MAPE (Mean Absolute Percentage Error) và Hit Rate [18]. Trong đó, MAPE đo độ lỗi về mặt giá trị, MAPE càng nhỏ thì càng tốt. Hit Rate đo độ chính xác về mặt xu hướng, Hit Rate càng lớn thì càng tốt.

Công thức tính của hai độ đo này như sau:

$$MAPE = \frac{1}{N}\sum_{n=1}^{N}\frac{|p_n - a_n| \times 100}{a_n} \quad (10)$$

$$Hit\ Rate = \frac{|\{n | r_n \times r'_n > 0, n = 1,\dots,N\}|}{|\{n | r_n \times r'_n \neq 0, n = 1,\dots,N\}|} * 100 \quad (11)$$

Trong đó:

$$\begin{cases} r_n = a_n - c_n \\ r'_n = p_n - c_n \end{cases}$$

Với p_n và a_n lần lượt là giá đóng cửa dự đoán và giá đóng cửa thực sự, c_n là giá đóng cửa (thực sự) của ngày hiện tại, N là số mẫu của tập thử nghiệm.

3. Kịch bản thử nghiệm và các tham số cài đặt

Để đánh giá chất lượng của phương pháp lai GA-SVR, chúng tôi so sánh kết quả dự đoán của phương pháp lai này với SVR sử dụng Grid Search để tìm bộ tham số tối ưu (Grid-SVR) và ANN. Do tính ngẫu nhiên của thuật giải di truyền, GA-SVR được thực thi 5 lần rồi lấy giá trị trung bình.

SVR trong cả hai phương pháp GA-SVR và Grid-SVR đều giống nhau với hàm lỗi ε-insensitive và hàm nhân Gaussian. Chúng tôi sử dụng thư viện LIBSVM [4] để thực thi SVR, thư viện AForge.NET[2] để thực thi GA và thư viện Neural Dot Net[3] để thực thi ANN.

Bảng 2 mô tả các tham số cài đặt của GA-SVR. Trong đó, kích thước quần thể và số vòng lặp tối đa được chọn thông qua thực nghiệm. Xác suất lai ghép và xác suất đột biến là các giá trị mặc định của thư viện thực thi GA. Miền giá trị của bộ tham số SVR được chọn dựa vào [17] và thực nghiệm. Số bit dùng để biểu diễn mỗi tham số SVR được chọn dựa vào miền giá trị của các tham số này. Với 20 bit dùng để biểu diễn mỗi tham số SVR, ta có chiều dài của một nhiễm sắc thể: 20 × 3 + 13 = 73 bit (với 3 là số lượng

[1]www.cophieu68.com

[2]http://www.aforgenet.com/framework

[3]http://neurondotnet.freehostia.com

các tham số SVR, 13 là số lượng các đặc trưng đầu vào).

Các tham số cài đặt của Grid-SVR được mô tả ở Bảng 3. Miền giá trị của bộ tham số SVR được chọn giống như GA-SVR. Bước tăng số mũ lưới thưa và lưới dày của Grid Search được chọn theo [7].

Bảng 4 mô tả các tham số cài đặt của ANN. Chúng tôi sử dụng mạng truyền thẳng 3 lớp, trong đó số node tầng ẩn được chọn thông qua thực nghiệm. Hệ số học là giá trị mặc định của thư viện thực thi ANN. Số vòng lặp tối đa được chọn thông qua thực nghiệm.

Bảng 2. Các tham số cài đặt GA-SVR

GA	
Kích thước quần thể	200
Số vòng lặp tối đa	500
Điều kiện dừng	Đạt số vòng lặp tối đa
Xác suất lai ghép	0.75
Xác suất đột biến	0.10
Miền giá trị của C	$[2^{-6}, 2^{8}]$
Miền giá trị của γ	$[2^{-8}, 2^{6}]$
Miền giá trị của ε	$[2^{-11}, 2^{-1}]$
Số bit biểu diễn mỗi tham số SVR	20

Bảng 3. Các tham số cài đặt Grid-SVR

Grid Search	
Miền giá trị của C	$[2^{-6}, 2^{8}]$
Miền giá trị của γ	$[2^{-8}, 2^{6}]$
Miền giá trị của ε	$[2^{-11}, 2^{-1}]$
Bước tăng số mũ của lưới thưa	2
Bước tăng số mũ của lưới dày	0.25

Bảng 4. Các tham số cài đặt ANN

ANN	
Kiến trúc mạng	3 lớp
Số node tầng ẩn	4
Hàm kích hoạt	Sigmoid
Hệ số học	Giảm dần qua mỗi vòng lặp từ 0.3 đến 0.05
Số vòng lặp tối đa	1000

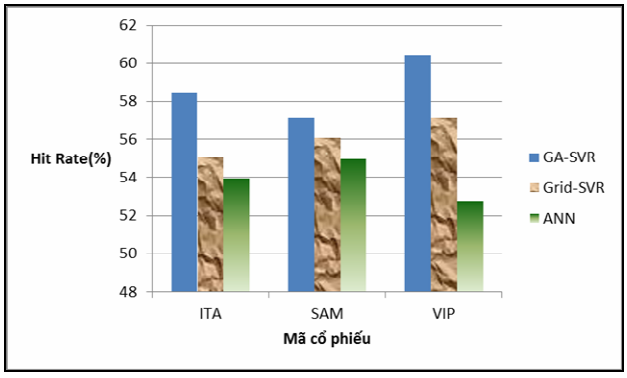
4. Kết quả thử nghiệm

Bảng 4 so sánh kết quả dự đoán giữa GA-SVR, Grid-SVRvà ANN. Ta thấy ở cả 3 mã cổ phiếu, GA-SVR luôn cho MAPE thấp hơn và Hit Rate cao hơn hai phương pháp còn lại. Hơn nữa, SVR luôn cho kết quả tốt dự đoán tốt hơn ANN. Điều này một lần nữa khẳng định tính vượt trội của SVR so với ANN trong bài toán dự đoán giá cổ phiếu, điều đã được nhiều nghiên cứu đề cập đến.

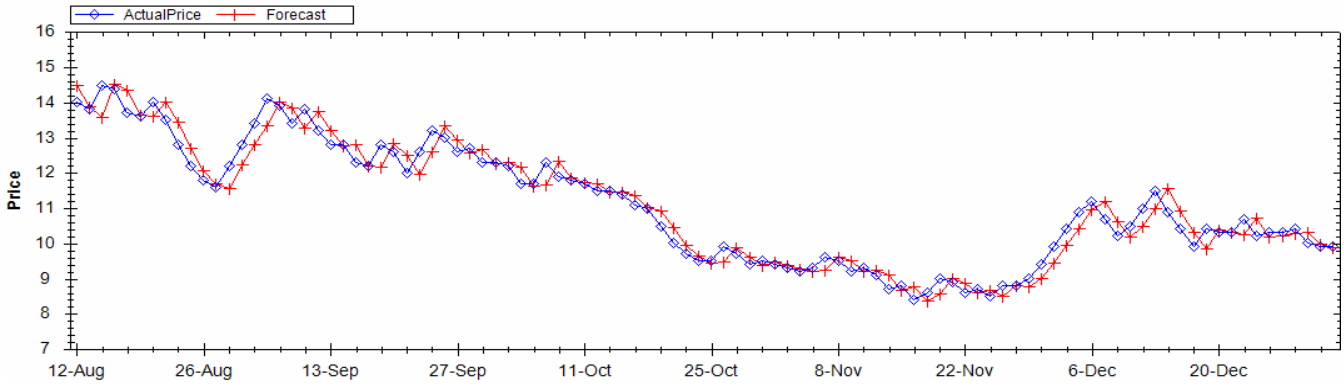
Kết quả dự đoán theo độ đo Hit Rate của ba phương pháp GA-SVR, SVR-Grid và ANN được thể hiện bằng đồ thị ở hình 8. Hit Rate của phương pháp lai GA-SVR ở 3 mã cổ phiếu đạt 58.427%, 57.143% và 60.44%. Đây là tín hiệu khả quan cho thấy khả năng ứng dụng thực tế các kỹ thuật máy học để giải quyết bài toán dự đoán giá cổ phiếu trên thị trường chứng khoán non trẻ Việt Nam.

Bảng 5. Kết quả dự đoán trung bình của GA-SVR, Grid-SVR và ANN

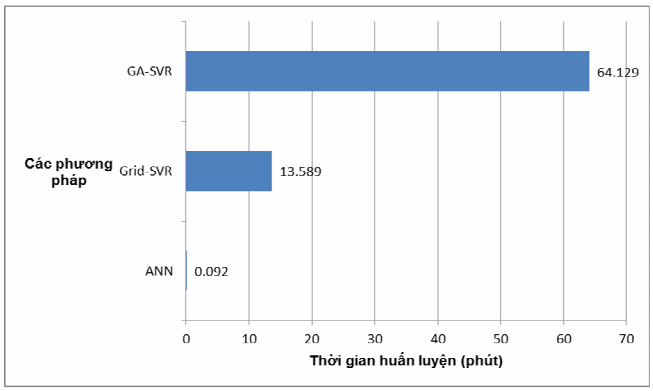
Mã	**Phương pháp**	**MAPE**	**Hit Rate**
ITA	GA-SVR	2.45	58.427
	Grid-SVR	2.474	55.056
	ANN	2.513	53.933
SAM	GA-SVR	2.36	57.143
	Grid-SVR	2.368	56.044
	ANN	2.382	54.945
VIP	GA-SVR	2.712	60.44
	Grid-SVR	2.763	57.143
	ANN	2.839	52.747



Hình 8. So sánh kết quả dự đoán theo độ đo Hit Rate



Hình 9. Minh họa kết quả dự đoán mã VIP với phương pháp GA-SVR



Hình 10. Thời gian huấn luyện của các phương pháp

Hình 9 minh họa kết quả dự đoán mã VIP bằng phương pháp GA-SVR. Trong đó, các điểm được đánh dấu bằng hình thoi thể hiện cho giá đóng cửa thực sự và các điểm được đánh dấu bằng hình dấu cộng thể hiện cho giá đóng cửa dự đoán.

Hình 10 cho thấy thời gian huấn luyện trung bình của các phương pháp. Phương pháp GA-SVR có thời gian huấn luyện trung bình lâu nhất trong 3 phương pháp. Tuy nhiên, đánh đổi lại là độ chính xác dự đoán. Về thời gian dự đoán, nhìn chung các phương pháp đều có thời gian dự đoán rất nhanh (thời gian dự đoán cho mỗi mẫu $\leq 0.15\text{x}10^{-3}$ giây).

Chúng tôi cũng so sánh mô hình đề xuất GA-SVR với kết quả của Chen và Ho [5]. Ở đây, Chen và Ho sử dụng GA để tìm bộ tham số tối ưu của SVR. Đặc trưng đầu vào trong bài báo này là giá đóng cửa và số đặc trưng đầu vào được chọn một cách thủ công bằng phương pháp thử và sai. Dữ liệu được sử dụng là mã TAIEX (Taiwan Stock Exchange Market Weighted Index) được lấy từ ngày 2/1/2001 đến ngày 23/1/2003 với 504 ngày giao dịch. Tập thử nghiệm bao gồm 100 ngày giao dịch gần đây nhất và số ngày dự đoán kế tiếp là 1 ngày. Bảng 6 cho thấy phương pháp đề xuất của chúng tôi cho độ lỗi MAPE thấp hơn phương pháp của Chen và Ho trên bộ dữ liệu của mã TAIEX.

Bảng 6. Kết quả theo độ đo MAPE của GA-SVR và phương pháp của Chen và Ho

Độ đo	**Phương pháp**	
	GA-SVR	Chen và Ho[5]
MAPE	**1.308**	1.316

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo này đề xuất phương pháp lai GA-SVR để dự đoán giá cổ phiếu Việt Nam. Trong phương pháp lai này, GA thực hiện đồng thời hai nhiệm vụ: xác định bộ tham số tối ưu cho SVR và chọn lựa các đặc trưng đầu vào. Kế đến, bộ tham số tối ưu và các đặc trưng đầu vào được chọn này sẽ được dùng để huấn luyện SVR. Kết quả thử nghiệm cho thấy phương pháp đề xuất cho kết quả dự đoán tốt hơn SVR, ANN và có khả năng ứng dụng thực tế trên thị trường chứng khoánViệt Nam, một thị trường còn non trẻ và kém ổn định.

Để chứng minh tính hiệu quả của phương pháp đề xuất, chúng tôi dự định tiếp tục thử nghiệm GA -SVR trên các mã cổ phiếu Việt Nam khác. Mặt khác, chúng tôi sẽ tiến hành thử nghiệm với các chỉ số phân tích kỹ thuật khác cũng như là tăng khoảng thời gian dự đoán từ 1 ngày kế tiếp lên 5-10 ngày kế tiếp.

TÀI LIỆU THAM KHẢO

[1] ABRAHAM A. , BAIKUNTH N., MAHANTI P. K., *Hybrid intelligent systems for stock market analysis*, LNCS, Springer-Verlag, Vol. 2074, 2001, pp. 337–345.

[2] ANG J.H., TEOH E.J., TAN C.H., GOH K.C., TAN K.C., *Dimension reduction using evolutionary Support Vector Machines*, IEEE Congress on Evolutionary Computation, 2008, pp. 3634-3641.

[3] BISHOP C.M., *Pattern Recognition and Machine Learning*, Springer, 2007.

[4] CHANG C-C., LIN C-J., *LIBSVM: A library for Support Vector Machines*.
http://www.csie.ntu.edu.tw/~cjlin/libsvm

[5] CHEN K-Y., HO C-H., *An Improved Support Vector Regression Modeling for Taiwan Stock Exchange Market Weighted Index Forecasting*, ICNN&B'05, 2005, Vol.3.

[6] GOLDBERG D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[7] HSU C-W., CHANG C-C., LIN C-J., *A Practical Guide to Support Vector Classication*.
http://www.csie.ntu.edu.tw/~cjlin

[8] HSU S-H., HSIEH JJ.P-A., CHIH T-C., HSU K-C., *A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression*, Expert Systems with Applications 36, 2009, pp. 7947–7951.

[9] HUANG C-L., TSAI C-Y., *A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting*, Expert Systems with Applications 36, 2009, pp. 1529–1539.

[10] HUANG C-L., WANG C-J., *A GA-based feature selection and parameters optimization for support vector machines*, Expert Systems with Applications 31, 2006, pp. 231–240.

[11] HUANG S-C., WU T-K., *Integrating GA-based time-scale feature extractions with SVMs for stock index forecasting*, Expert Systems with Applications 35, 2008, pp. 2080–2088.

[12] HUERTA E.B., DUVAL B., HAO J-K., *A Hybrid GA/SVM Approach for Gene Selection and Classification of Microarray Data*, EvoWorkshops, 2006, pp. 34-44.

[13] INCE H., TRAFALIS T.B., *Kernel Principal Component Analysis and Support Vector Machines for Stock Price Prediction*, IIE Transactions on Quality and Reliability, 39(6), 2007, pp. 629-637.

[14] LEE M-C., *Using support vector machine with a hybrid feature selection method to the stock trend prediction*, Expert Systems with Applications 36, 2009, pp. 10896–10904.

[15] MAGER J., PAASCHE U., SICK B., *Forecasting Financial Time Series with Support Vector Machines Based on Dynamic Kernels*, IEEE Conference on Soft Computing in Industrial Applications, 2008, pp. 252-257.

[16] MINGDA W., LAIBIN Z., WEI L., YINGCHUN Y., *Research on the optimized support vector regression machines based on the differential evolution algorithm*, ICIECS'2009, 2009, pp. 1-4.

[17] MOMMA M., BENNETT K. P., *A pattern search method for model selection of support vector regression*, SIAM Conference on Data Mining, 2002, pp. 261-274.

[18] NYGREN K., *Stock Prediction – A Neural Network Approach*, Master thesis, 2004.

[19] SAPANKEVYCH N.I., SANKAR R., *Time Series Prediction Using Support Vector Machines: A Survey*, IEEE Computational Intelligence Magazine, Vol. 4, No. 2, 2009, pp. 24-38.

[20] ZHU M., WANG L., *Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms*, IJCNN'2010, 2010, pp. 1-5.

PHỤ LỤC

Công thức tính của các chỉ số phân tích kỹ thuật[4]

1. *BB-Middle(20, 2):* chỉ số Bollinger Band gồm có 3 dải ứng với BB-Middle, BB-Upper và BB-Lower

$$\text{BB-Middle}(20,2)_t = \text{SMA20}_t \quad (1)$$

$$\text{BB-Upper}(20,2)_t = \text{SMA20}_t + 2\times\text{SD20}_t \quad (2)$$

$$\text{BB-Upper}(20,2)_t = \text{SMA20}_t - 2\times\text{SD20}_t \quad (3)$$

Trong đó,$SMA20_t$và $SD20_t$ lần lượt là trung bình và độ lệch chuẩncủa giá đóng cửa của 20 ngày trước ngày t (kể cả ngày t)

2. EMA5 (Exponential Moving Average)

$$\text{EMA5}_t = \left(C_t - \text{EMA5}_{t-1}\right)\times k + \text{EMA5}_{t-1} \quad (4)$$

Trong đó, C_t là giá đóng cửa ngày t, k là hệ số nhân: k = 2/(1+period) với period = 5

3. MACD(12, 26) (Moving Average Convergence/ Divergence)

$$\text{MACD}(12,26)_t = \text{EMA12}_t - \text{EMA26}_t \quad (5)$$

4. MACD Signal

$\text{MACD Signal}_t = \text{EMA9}_t$ của MACD(12,26) (6)

5. RSI7 (Relative Strength Index)

$$\text{RSI7}_t = \frac{\sum_{i=t-6}^{t} \text{Gain}_i}{\sum_{i=t-6}^{t} \text{Gain}_i + \sum_{i=t-6}^{t} \text{Loss}_i} \times 100 \quad (7)$$

Trong đó:

$$\text{Gain}_i = \begin{cases} 0 \text{ nếu } C_i < C_{i-1} \\ C_i - C_{i-1} \text{ nếu } C_i > C_{i-1} \end{cases}$$

$$\text{Loss}_i = \begin{cases} 0 \text{ nếu } C_i > C_{i-1} \\ C_{i-1} - C_i \text{ nếu } C_i < C_{i-1} \end{cases}$$

Với C_k là giá đóng cửa ngày k

6. ROC-p (Rate Of Change)

$$\text{ROC-p}_t = \frac{C_t - C_{t-p}}{C_{t-p}} \times 100 \quad (8)$$

Với C_k là giá đóng cửa ngày k

Nhận bài ngày: 28/3/2011

[4]www.stockcharts.com

SƠ LƯỢC VỀ TÁC GIẢ

TRẦN TRUNG KIÊN

Ngày sinh 07/08/1989

Tốt nghiệp Trường Đại học Khoa Học Tự Nhiên, Đại học Quốc gia Tp. HCM năm 2011.

Hiện là trợ giảng tại Khoa CNTT, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Tp. HCM

Lĩnh vực quan tâm: máy học và ứng dụng.

ĐT: 0976044860, Email: ttkien@fit.hcmus.edu.vn

BÀNH TRÍ THÀNH

Ngày sinh 16/04/1989

Tốt nghiệp Đại học Khoa Học Tự Nhiên, Đại học Quốc gia Tp.HCM năm 2011.

Lĩnh vực quan tâm: máy học, xử lý ảnh.

ĐT: 0908828391, Email: 89btthanh@gmail.com

NGUYỄN HOÀNG TÚ ANH

Ngày sinh 02/03/1969

Tốt nghiệp Đại học Tổng hợp Kishinhốp, Cộng hòa Mônđôva năm 1992. Bảo vệ luận án Thạc sĩ ngành Tin học tại Trường Đại học Khoa Học Tự Nhiên, Đại học Quốc gia Tp. HCM, 2002.

Hiện là giảng viên Khoa CNTT, Trường Đại học Khoa Học Tự Nhiên, Đại học Quốc gia Tp.HCM

Lĩnh vực nghiên cứu: công nghệ tri thức và ứng dụng, khai thác dữ liệu, text mining, web mining.

ĐT : 091 826 1438, Email: nhtanh@fit.hcmus.edu.vn