MÔ HÌNH RU-SVR TRONG CHIẾN LƯỢC DỰ BÁO CHUỖI THỜI GIAN NHIỀU BƯỚC

RECURSIVE UPDATE SVR MODEL FOR MULTI-STEP AHEAD TIME SERIES FORECASTING STRATEGY

Hà Bình Minh[1], Nguyễn Hoàng Ân[1], Phan Đình Phùng[1], Chu Bình Minh[2], Nguyễn Mai Quyên[3]

[1]Khoa Hệ thống thông tin quản lý, Trường Đại học Ngân hàng Thành phố Hồ Chí Minh
[2]Khoa Khoa học ứng dụng, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp
[3]Khoa Toán kinh tế, Trường Đại học Kinh tế quốc dân

Đến Tòa soạn ngày 05/06/2024, chấp nhận đăng ngày 22/06/2024

Tóm tắt: Bài báo này áp dụng một chiến lược dự báo chuỗi thời gian nhiều bước được gọi là RU-SVR. Phương pháp này liên tục cập nhật mô hình SVR để cải thiện độ chính xác và tính ổn định của dự báo trong các biến động dữ liệu. Nghiên cứu đã thực hiện các thực nghiệm trên tập dữ liệu chỉ số VN-Index từ sàn giao dịch chứng khoán Thành phố Hồ Chí Minh, so sánh hiệu quả của RU-SVR với các mô hình dự báo hiện đại như LSTM, MLP và ConvLSTM. Kết quả thực nghiệm cho thấy RU-SVR không chỉ đạt độ chính xác cao hơn mà còn ổn định hơn trong dự báo chuỗi thời gian nhiều bước.

Từ khóa: RU-SVR, LSTM, ConvLSTM, MLP, Multi-Step Ahead.

Abstract: This paper proposes a novel multi-step time series forecasting strategy called RU-SVR. This method iteratively updates the SVR model to enhance the accuracy and stability of forecasts amidst data fluctuations. The study conducted experiments on the VN-Index dataset from the Ho Chi Minh City Stock Exchange, comparing the performance of RU-SVR against contemporary forecasting models such as LSTM, MLP, and ConvLSTM. Experimental results indicate that RU-SVR not only achieves higher accuracy but also demonstrates greater stability in multi-step time series forecasting.

Keywords: RU-SVR, LSTM, ConvLSTM, MLP, Multi-Step Ahead.

1. GIỚI THIỆU

Bài toán dự báo chuỗi thời gian luôn là một trong các vấn đề trọng tâm trong những nghiên cứu về mặt lý thuyết, cũng như trong những nghiên cứu về mặt ứng dụng thực tiễn. Thông qua việc dự báo chuỗi thời gian, nhiều vấn đề kỹ thuật phức tạp trong các lĩnh vực, ngành nghề khác nhau đã được giải quyết thấu đáo, chẳng hạn như dự báo tốc độ gió tại các trang trại điện gió [1], dự báo tình trạng ô nhiễm bụi mịn PM2.5 trong các thành phố lớn, dự báo giá dầu thô [2], dự báo giá vàng [3], …

Những mô hình được sử dụng trong bài toán dự báo chuỗi thời gian rất đa dạng, từ những mô hình thống kê cho đến những mô hình học máy. Một số mô hình dự báo phổ biến là mô hình ARIMA (Autoregressive Integrated Moving Average [4]), mô hình GARCH (Generalized Autoregressive Conditional Hetoroskedasticity [4]), mô hình ANN (Artificial Neural Network [5, 6]), mô hình SVR (Support Vector Regression [7]), mô hình RNN (Recurrent Neural Network [8]), mô hình LSTM (Long Shortterm Memory Network [9]), mô hình CNN (Convolutional Neural Network [10]), mô hình TCN

(Temporal Convolutional Network [11]), mô hình GKFN (Gaussian Kernel Function Network [12]), mô hình RUGKFN (Recursive Update Gaussian Kernel Function Network [12])…

Ý tưởng về mô hình cập nhật liên tiếp được đưa ra lần đầu trong [12]. Trong bài báo này, nhóm tác giả đã xây dựng một sơ đồ huấn luyện dựa trên hai yếu tố chính: (1) độ dài của vectơ đầu vào, (2) nhân của mô hình dự báo chính là mô hình GKFN. Dựa vào ý tưởng trong [12], chúng tôi xây dựng một sơ đồ huấn luyện tương tự, tuy nhiên khác với [12], nhân của mô hình dự báo của chúng tôi dựa vào mô hình SVR.

Trong phần tiếp theo của bài báo, các mục được phân chia theo cấu trúc sau: Mục 2 giới thiệu về các kiến thức chuẩn bị, mục 3 trình bày về mô hình RU-SVR, là mô hình đề xuất được đưa ra trong bài báo, mục 4 dành cho kết quả tính toán chạy trên dữ liệu thực tế là chuỗi chỉ số VN-Index từ Sở giao dịch chứng khoán Thành phố Hồ Chí Minh. Cuối cùng, phần kết luận của bài báo được tóm lược trong Mục 5.

2. CÁC KIẾN THỨC CHUẨN BỊ

2.1. Bài toán dự báo chuỗi thời gian một bước

Bài toán dự báo chuỗi thời gian một bước được hiểu là việc đi tìm giá trị kế tiếp của chuỗi thời gian dựa trên các quan sát quá khứ trong một khoảng thời gian có độ dài cố định nào đó. Nếu ta gọi L là độ dài của khoảng thời gian của các quan sát trong quá khứ thì vectơ $\mathbf{x}_{t,L}$ chứa L quan sát trong quá khứ từ thời điểm t, được định nghĩa như sau:

$$\mathbf{x}_{t,L} = (x_t, x_{t-1}, \dots, x_{t-L+1})$$

với $x_t, t = 1,2,\dots$ là các quan sát của chuỗi thời gian.

Về mặt toán học, bài toán dự báo chuỗi thời gian một bước được phát biểu như sau: Cho trước chuỗi thời gian $x_t, t = 1,2,..$, ta phải tìm mô hình dự báo f sao cho giá trị dự báo một bước được tính theo công thức:

$$\hat{x}_{t+1} = f(\mathbf{x}_{t,L})$$

có giá trị gần với giá trị thực tế trong chuỗi thời gian, tức là sai số $||\hat{x}_{t+1} - x_{t+1}||$ phải đủ nhỏ. Để đánh giá mô hình dự báo có phù hợp với dữ liệu thực tế hay không, ta sử dụng các độ đo sai số sau đây: MAE (Mean Absolute Error), RMSE (Root Mean Square Error) và MAPE (Mean Absolute Percentage Error). Các độ đo này được tính toán như trong bảng 1.

Bảng 1. Công thức tính toán sai số

MAE	RMSE	MAPE
$\frac{1}{N}\sum_{t=1}^{N}\lvert x_t - \hat{x}_t\rvert$	$\sqrt{\frac{\sum_{t=1}^{N}(x_t - \hat{x}_t)^2}{N}}$	$\frac{1}{N}\sum_{t=1}^{N}\frac{\lvert x_t - \hat{x}_t\rvert}{x_t}$

2.2. Bài toán dự báo chuỗi thời gian nhiều bước

Thay vì chỉ dự đoán giá trị $\hat{x}_{t+1}$ như trong bài toán dự báo chuỗi thời gian một bước, ta sẽ dự đoán một loạt các giá trị tương lai

$$\hat{x}_{t+1,}\hat{x}_{t+2}, \dots, \hat{x}_{t+h}$$

cùng một lúc trong bài toán dự báo chuỗi thời gian nhiều bước. Một số chiến lược trong bài toán dự báo chuỗi thời gian nhiều bước được quan tâm như: chiến lược trực tiếp (direct stratery) [3], chiến lược lặp nhiều bước (recursive multi-step strategy) [12], chiến lược nhiều đầu ra (multiple-output strategy) [1], ...

Trong mục kế tiếp chúng tôi sẽ trình bày mô hình SVR, là mô hình dự báo cốt lõi mà chiến lược dự báo của chúng tôi dựa vào.

2.3. Mô hình dự báo SVR

Thuật toán SVM (Support Vector Machine) là một trong những thuật toán học có giám sát phổ biến trong machine learning [14]. SVM được sử dụng chủ yếu cho các bài toán phân loại và hồi quy. Thuật toán này dựa trên việc tìm kiếm một đường ranh giới tốt nhất để chia các điểm dữ liệu thành các lớp khác nhau.

Ban đầu, thuật toán SVM được sử dụng trong bài toán phân loại nhị phân [15]. Với các điểm dữ liệu có thể tách tuyến tính thì luôn tồn tại siêu phẳng phân chia mà mỗi lớp các điểm dữ liệu sẽ nằm một phía của siêu phẳng phân chia này. Có một số thuật toán tìm siêu phẳng phân chia nhưng thuật toán SVM cho phép tìm siêu phẳng phân chia một cách tốt nhất, nghĩa là tìm được siêu phẳng mà có khoảng cách từ các điểm dữ liệu gần nhất của mỗi lớp đến siêu phẳng là lớn nhất. Khoảng cách này gọi là lề (margin).

Giả sử, dữ liệu của tập huấn luyện dạng $\{(\mathbf{x}_i, y_i)\}^N{}_{i=1}$, với vectơ $\mathbf{x}_i \in \mathbb{R}^d$ là đầu vào của một điểm dữ liệu và $y_i \in \{-1,1\}$ là nhãn tương ứng của điểm dữ liệu đó. Các giá trị $-1,1$ của y_i thể hiện việc điểm dữ liệu $\mathbf{x}_i$ tương ứng thuộc class 1, 2. Nếu siêu phẳng phân chia cần tìm là

$$\mathbf{w}^T\mathbf{x} + b = 0 \quad (1)$$

thì khoảng cách từ điểm dữ liệu $(\mathbf{x}_n, y_n)$ gần nhất đến siêu phẳng này là

$$\text{margin} = \min_n \frac{y_n(\mathbf{w}^T\mathbf{x} + b)}{\|\mathbf{w}\|_2}.$$

Ta cần tìm các tham số $\mathbf{w}, b$ sao cho lề đạt giá trị lớn nhất, tức là

$$(\mathbf{w}, b) = \arg\max_{\mathbf{w},b}\left[\min_n \frac{y_n(\mathbf{w}^T\mathbf{x} + b)}{\|\mathbf{w}\|_2}\right]. \#(2)$$

Bài toán (2) có thể đưa về dạng

$$(\mathbf{w}, b) = \arg\min_{(\mathbf{w},b)} \frac{1}{2}\|\mathbf{w}\|_2^2,$$

thỏa mãn điều kiện

$$1 - y_n(\mathbf{w}^T\mathbf{x}_\text{n} + b) \le 0, \forall n = 1,2,\dots,N. \#(3)$$

Nếu tìm được siêu phẳng phân chia $\mathbf{w}^T\mathbf{x} + b = 0$ thì class của điểm dữ liệu $\mathbf{x}$ bất kỳ sẽ được xác định bởi

$$\text{class}(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b),$$

với $sign(.)$ là hàm dấu.

Khi siêu phẳng phân tách hai class với biên quá nhỏ hoặc trường hợp dữ liệu gần phân tách tuyến tính được, ta sử dụng một cải tiến của thuật toán SVM gọi là thuật toán Soft-margin SVM [16]. Ý tưởng của thuật toán Soft-margin SVM là chấp nhận cho một số điểm rơi vào vùng không an toàn để tìm được lề lớn. Như vậy, mục tiêu của thuật toán Soft-margin SVM là tối đa lề và tối thiểu số điểm rơi và vùng không an toàn.

Trong trường hợp các điểm dữ liệu hoàn thoàn không phân tách tuyến tính được, thuật toán Kernel SVM sẽ tìm một phép biến đổi để biến đổi dữ liệu $\mathbf{x}$ từ không gian các đặc tính ban đầu tới một không gian có số chiều cao hơn [17]. Một số hàm kernel thường được sử dụng là Linear Kernel, Polynomial Kernel, Gaussian Kernel, Sigmoid Kernel.

Thuật toán SVM không chỉ được sử dụng cho bài toán phân loại mà còn có thể áp dụng cho bài toán hồi quy. Trên thực tế, SVM trong hồi quy thường được gọi là SVR (Support Vector Regression) [18]. Giả sử, dữ liệu của tập huấn luyện dạng $\{(\mathbf{x}_i, y_i)\}^N{}_{i=1}$, với vectơ $\mathbf{x}_i \in \mathbb{R}^d$ là đầu vào của một điểm dữ liệu và $y_i \in \mathbb{R}$ là biến cần dự báo tương ứng của điểm dữ liệu. Mục tiêu của SVR là tìm một hàm $f(x)$ có dạng

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b, \# \quad (4)$$

với $\mathbf{w}$ là vectơ trọng số và b là bias được xác định bởi

$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}(\xi_n + \xi_n^*), \#(5)$$

thỏa mãn các ràng buộc

$$\begin{aligned} &y_n - (\mathbf{w}^T\mathbf{x}_\mathrm{n} + b) \le \varepsilon + \xi_n, \\ &(\mathbf{w}^T\mathbf{x}_\mathrm{n} + b) - y_n \le \varepsilon + \xi_n^*, \# \\ &\xi_n, \xi_n^* \ge 0, \forall n = 1,2,\dots,N, \end{aligned} \quad (6)$$

với C là tham số điều chỉnh mức độ quan trọng giữa việc giảm lỗi và việc giữ cho mô hình đơn giản; ε là phạm vi sai số cho phép; ξ_n, ξ_n^* là biến giả cho phép điểm dữ liệu nằm ngoài khoảng ε.

Để giải bài toán tối ưu này, chúng ta sử dụng phương pháp nhân tử Lagrange. Hàm Lagrangian cho SVR được xác định như sau:

$$\begin{aligned} \mathcal{L} \quad &= \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}(\xi_n + \xi_n^*) \\ &+ \sum_{n=1}^{N}\alpha_n[y_n - (\mathbf{w}^T\mathbf{x}_\mathrm{n} + b) - \varepsilon - \xi_n] \\ &+ \sum_{n=1}^{N}\alpha_n^*[(\mathbf{w}^T\mathbf{x}_\mathrm{n} + b) - y_n - \varepsilon - \xi_n] \\ &+ \sum_{n=1}^{N}\eta_n\xi_n + \sum_{n=1}^{N}\eta_n^*\xi_n^*, \end{aligned} \quad \#(7)$$

trong đó $\alpha_n, \alpha_n^*, \eta_n, \eta_n^*$ là các nhân tử Lagrange.

Giải bài toán đối ngẫu của hàm Lagrangian này giúp tìm ra các giá trị $\mathbf{w}$ và b tối ưu.

Trong trong trường hợp các điểm dữ liệu hoàn thoàn không phân tách tuyến tính được, ta có thể sử dụng hàm kernel để ánh xạ dữ liệu từ không gian đầu vào ban đầu sang một không gian đặc trưng cao hơn, nơi mà dữ liệu có thể trở nên tuyến tính hơn hoặc dễ phân tách hơn. Khi đó, hàm $f(\mathbf{x})$ có dạng

$$f(\mathbf{x}) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)k(\mathbf{x}) + b, \# \quad (8)$$

với $k(\mathbf{x})$ là hàm kernel.

Thuật toán SVR được thực hiện như sau:

Bảng 2. Thuật toán SVR

Bước 1.	Chuẩn bị dữ liệu: Tiền xử lý dữ liệu bằng cách chuẩn hóa và làm sạch dữ liệu nếu cần thiết.
Bước 2.	Xác định tham số: Xác định C, ε.
Bước 3.	Xác định hàm kernel: Lựa chọn hàm kernel phù hợp để ánh xạ dữ liệu vào không gian phi tuyến tính.
Bước 4.	Xây dựng mô hình: - Thiết lập hàm mục tiêu (5) và các điều kiện ràng buộc (6); - Tạo hàm Lagrangian (7); **+ Giải bài toán đối ngẫu** của (7) để tìm các giá trị $\mathbf{w}, b, \alpha_n, \alpha_n^*$.
Bước 5.	Huấn luyện mô hình: Sử dụng dữ liệu huấn luyện để xây dựng mô hình SVR. Giải bài toán tối ưu hóa để tìm trọng số $\mathbf{w}$ và b.
Bước 6.	Đánh giá và tinh chỉnh mô hình: -Sử dụng các độ đo như MAE, RMSE và MAPE để đánh giá hiệu suất của mô hình. - Tối ưu các tham số C và ε để cải thiện hiệu suất của mô hình.
Bước 7.	Dự đoán giá trị mới: Sử dụng mô hình SVR đã được huấn luyện để dự đoán giá trị đầu ra cho các dữ liệu mới.

Qua các bước trên, thuật toán SVR có thể áp dụng vào bài toán hồi quy và tạo ra một mô hình dự đoán chính xác cho giá trị liên tục. Việc lựa chọn hàm kernel và tinh chỉnh tham số quan trọng để đạt được hiệu suất tốt cho bài toán hồi quy cụ thể.

3. MÔ HÌNH ĐỀ XUẤT

3.1. Mô hình dự báo RU-SVR

Mô hình dự báo được đề xuất trong bài báo

này là mô hình Recursive Update SVR (RU-SVR). Đây là một phương pháp học máy dựa trên thuật toán SVR nhưng có thêm tính năng đặc biệt để cập nhật mô hình khi có dữ liệu mới được thêm vào. Điều này cho phép mô hình thích ứng với dữ liệu mới mà không cần huấn luyện lại toàn bộ dữ liệu cũ. Do vậy, mô hình RU-SVR có thể sử dụng để dự báo các giá trị trong chuỗi thời gian trên nhiều bước tiếp theo. Mục tiêu của mô hình là học mối quan hệ giữa các giá trị trong chuỗi thời gian và dự đoán các giá trị tương lai trong chuỗi trên nhiều bước tiếp theo.

Mô hình đề xuất trong bài báo này được minh họa trong bảng 3.

Bảng 3. Mô hình dự báo RU-SVR đề xuất

Bước 1.	Xác định khoảng quan sát tối ưu L từ dữ liệu chuỗi thời gian
Bước 2.	Chia dữ liệu chuỗi thời gian thành 2 phần: bộ dữ liệu huấn luyện (80%) và bộ dữ liệu kiểm tra (20% còn lại)
Bước 3.	Huấn luyện mô hình cập nhật liên tiếp RU-SVR trên bộ dữ liệu huấn luyện
Bước 4.	So sánh, đánh giá, kiểm nghiệm các mô hình trên bộ dữ liệu kiểm tra (20% còn lại)

Trong các mục tiếp theo, chúng tôi sẽ trình bày chi tiết các vấn đề kỹ thuật trong Bước 1 và Bước 3, là những bước quan trọng trong chiến lược dự báo đề xuất. Các kết quả so sánh, đánh giá, kiểm nghiệm trong Bước 4 sẽ được trình bày chi tiết trong mục 4 của bài báo.

3.2. Bước 1 trong mô hình dự báo RU-SVR đề xuất

Để xác định khoảng quan sát tối ưu từ dữ liệu đầu vào, trước tiên ta cần đo độ trơn của hàm dự báo f. Độ trơn của hàm dự báo được xác định trong công thức sau đây [12, 13]:

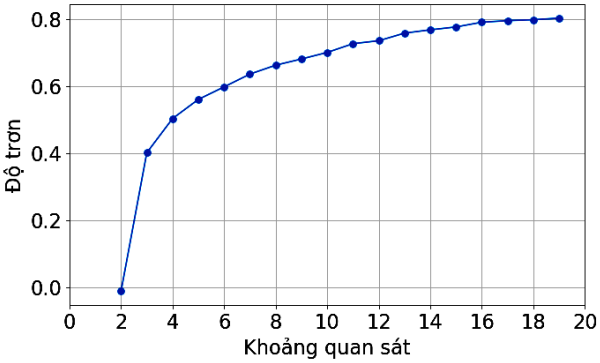
$$S(L,f) = 1 - \frac{1}{n-(L-1)} \sum_{t=L-1}^{n} \Delta f(\mathbf{x}_{t,L})$$

trong đó $\Delta f(\mathbf{x}_{t,L})$ là gradient của hàm f được tính như sau

$$\Delta f(\vec{x}_{t,L}) = \frac{|f(\mathbf{x}_{t,L}) - f(\mathbf{x}^1_{t,L})|}{||\ \mathbf{x}_{t,L} - \mathbf{x}^1_{t,L}\ ||},$$

với $\mathbf{x}^1_{t,L}$ là vectơ lân cận của vectơ $\mathbf{x}_{t,L}$ trong không gian Euclid nhiều chiều cùng với chuẩn tương ứng.

Tiếp theo, ta sẽ vẽ đồ thị của độ trơn $S(L,f)$ theo tham số L như trong hình 1.



Hình 1. Đồ thị thể hiện độ trơn $S(L,f)$ theo tham số L

Đồ thị trong hình 1 cho ta thấy rằng $S(L,f)$ sẽ tăng dần tới 1 khi L ngày càng lớn. Nếu ta chọn số L lớn để huấn luyện cho mô hình thì thời gian để tính toán và huấn luyện mô hình sẽ tăng thêm nhiều lần. Vì vậy, vấn đề đặt ra là cần chọn khoảng thời gian quan sát L một cách tối ưu sao cho nó vừa đủ lớn để có độ trơn gần 1, và vừa đủ nhỏ để có thể huấn luyện mô hình một cách nhanh chóng. Trên thực tế, ta sẽ chọn L sao cho $S(L,f)$ là số dương và gia số biến đổi của $S(L,f)$ sẽ là lớn nhất khi L tăng giảm một đơn vị. Với cách chọn như vậy, dựa vào hình 1, ta sẽ chọn $L = 4$. Theo cách chọn này, ta sẽ dễ dàng huấn luyện được mô hình và thu được giá trị dự báo với độ chính xác cao.

3.3. Bước 3 trong mô hình dự báo RU-SVR đề xuất

Phần quan trọng nhất của mô hình được đề xuất trong bảng 3 là nhân của mô hình, mô hình dự báo RU-SVR. Mô hình này được xây dựng bằng cách cập nhật liên tiếp các giá trị dự báo ở ở bước trước vào đầu vào cho mô hình SVR để dự báo các bước sau. Cụ thể, với chuỗi thời gian $x_t, t = 1,2,\ldots$, và L đã biết, mô hình RU-SVR dự báo h bước tương lai

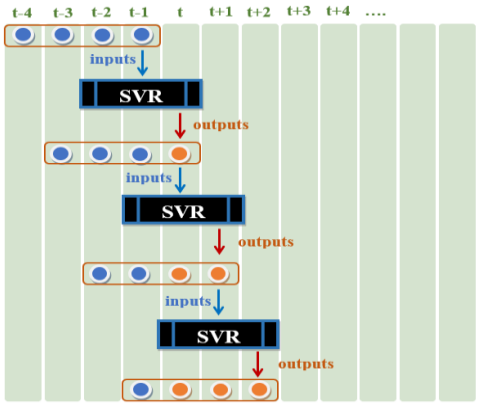
$\hat{x}_{t+1,}\hat{x}_{t+2},\ldots,\hat{x}_{t+h}$

được xác định như thuật toán trong bảng 4.

Bảng 4. Tính toán chi tiết của Bước 3 trong mô hình dự báo RU-SVR đề xuất

Bước 1.	$i := 0$.
Bước 2.	Tính $\hat{x}_{t+i+1} = \hat{f}(\vec{x}_{t,L})$ với $\hat{f}$ là mô hình SVM hồi quy, $\vec{x}_{t+i,L} = (\tilde{x}_{t+i}, \tilde{x}_{t+i-1}, \ldots, \tilde{x}_{t+i-L+1})$ và $\tilde{x}_{t+j} = \begin{cases} \hat{x}_{t+j} & khi\ j > 0 \\ & j = i, i-1, \cdots, i-L+1 \\ x_{t+j} & ngược\ lại \end{cases}$
Bước 3.	Nếu $i = h - 1$: Dừng. Ngược lại: $i := i + 1$, quay lại Bước 2.

Hình 2 minh họa sơ đồ tính toán chi tiết trong bảng 4.



Hình 2. Sơ đồ thể hiện tính toán trong bảng 4

4. KẾT QUẢ THỰC NGHIỆM

Thực nghiệm được thực hiện trên tập dữ liệu chỉ số VN-Index của sàn giao dịch chứng khoán Thành phố Hồ Chí Minh từ ngày 01/01/2012 đến ngày 30/03/2023, với tổng cộng 2803 ngày giao dịch. Tập dữ liệu này được chia thành hai phần: 80% (tương đương 2242 ngày giao dịch) được sử dụng để huấn luyện mô hình và 20% còn lại (516 ngày giao dịch) để đánh giá mô hình. Trong thực nghiệm, thư viện Tensorflow được sử dụng để xây dựng mô hình dự báo từ 1 đến 10 bước theo phương pháp đề xuất. Kết quả thực nghiệm, được trình bày trong Bảng 5, bao gồm so sánh giữa mô hình đề xuất RU-SVR với ba mô hình ConvLSTM (Convolutional-LSTM), LSTM (Long Short Term Memory) và MLP (Multi-layer Perceptron). Các mô hình này được đánh giá dựa trên ba chỉ số: RMSE, MAE, và MAPE. Kết quả cho thấy mô hình dự báo RU-SVR đạt hiệu suất tốt hơn so với các mô hình còn lại ở tất cả các bước dự báo.

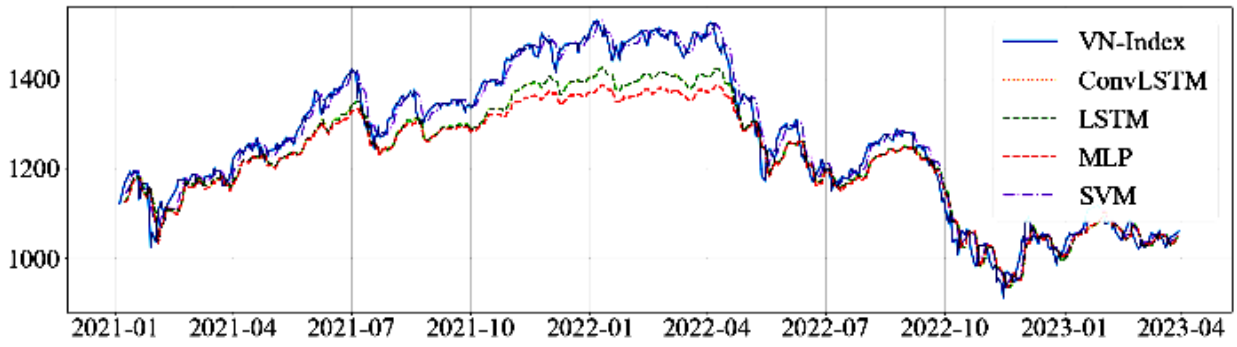
Bảng 5. So sánh kết quả thực nghiệm trên các chỉ số RMSE, MAE và MAPE

Bước	Mô hình	RMSE	MAE	MAPE
1	ConvLSTM	35.17	26.55	1.99
	RU-SVR	**17.50**	**12.40**	**1.02**
	LSTM	61.41	44.52	3.25
	MLP	59.27	45.35	3.37
2	ConvLSTM	42.12	32.38	2.43
	RU-SVR	**19.51**	**13.83**	**1.14**
	LSTM	64.44	46.60	3.40
	MLP	54.45	42.96	3.25
3	ConvLSTM	43.45	33.44	2.51
	RU-SVR	**20.65**	**14.80**	**1.22**
	LSTM	58.35	41.72	3.06

Bước	Mô hình	RMSE	MAE	MAPE
	MLP	58.64	44.84	3.32
4	ConvLSTM	35.37	26.79	2.06
	RU-SVR	**21.60**	**15.46**	**1.28**
	LSTM	66.85	46.26	3.36
	MLP	71.89	58.20	4.38
5	ConvLSTM	49.25	38.72	2.93
	RU-SVR	**23.63**	**17.06**	**1.41**
	LSTM	45.93	32.78	2.42
	MLP	65.22	50.20	3.70
6	ConvLSTM	54.18	43.98	3.32
	RU-SVR	**24.38**	**17.71**	**1.46**
	LSTM	41.09	32.22	2.42
	MLP	66.05	52.57	3.93
7	ConvLSTM	57.02	46.65	3.50
	RU-SVR	**24.31**	**17.60**	**1.45**
	LSTM	51.28	40.23	3.01

Bước	Mô hình	RMSE	MAE	MAPE
	MLP	66.27	52.90	3.96
8	ConvLSTM	62.18	50.32	3.75
	RU-SVR	**23.77**	**17.30**	**1.43**
	LSTM	66.94	54.39	4.05
	MLP	61.84	48.68	3.63
9	ConvLSTM	37.12	29.45	2.27
	RU-SVR	**24.05**	**17.53**	**1.44**
	LSTM	68.46	54.07	4.01
	MLP	61.62	48.27	3.59
10	ConvLSTM	42.70	35.18	2.73
	RU-SVR	**24.10**	**17.80**	**1.46**
	LSTM	72.57	56.06	4.13
	MLP	65.86	51.44	3.80

Trong hình 3 minh họa kết quả dự báo cho 4 mô hình cùng với giá trị thật của VN-Index ở bước dự báo thứ 5.



Hình 3. Đồ thị thể hiện kết quả dự báo ở bước thứ 5

5. KẾT LUẬN

Dựa trên các phân tích và kết quả thu được, phương pháp dự báo chuỗi thời gian nhiều bước theo mô hình đề xuất RU-SVR đã chứng tỏ hiệu quả tốt hơn so với các phương pháp LSTM, MLP và ConvLSTM. Cụ thể, mô hình RU-SVR không chỉ mang lại độ chính xác cao hơn trong việc dự báo mà còn ổn định hơn trong việc xử lý các biến động dữ liệu ở nhiều bước. Điều này có thể được lý giải bởi khả năng của mô hình RU-SVR trong việc xử lý các mẫu dữ liệu phi tuyến và ngoài mẫu hiệu quả hơn. Do đó, mô hình RU-SVR là lựa chọn tốt cho các bài toán dự báo chuỗi thời gian trong bối cảnh nghiên cứu này.

TÀI LIỆU THAM KHẢO

[1] Li, D., Jiang, F., Chen, M., & Qian, T., Multi-step-ahead wind speed forecasting based on a hybrid decomposition method and temporal convolutional networks. Energy, 238, 121981, (2022).

[2] Chen, Y., Zhang, C., He, K., & Zheng, A., Multi-step-ahead crude oil price forecasting using a hybrid grey wave model. Physica A: Statistical Mechanics and its Applications, 501, 98-110 (2018).

[3] Livieris, I. E., & Pintelas, P., A novel multi-step forecasting strategy for enhancing deep learning models' performance. Neural Computing and Applications, 1-18, (2022).

[4] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M., Time series analysis: forecasting and control. John Wiley & Sons (2015).

[5] Guresen, E., Kayakutlu, G., & Daim, T. U., Using artificial neural network models in stock market index prediction. Expert Systems with Applications, 38(8), 10389-10397, (2021).

[6] Bao, T. Q., & My, B. T. T., Forecasting stock index based on hybrid artificial neural network models. Science & Technology Development Journal-Economics-Law and Management, 3(1), 52-57, (2019).

[7] Santamaría-Bonfil, G., Reyes-Ballesteros, A., & Gershenson, C. J. R. E., Wind speed forecasting for wind farms: A method based on support vector regression. Renewable Energy, 85, 790-809, (2016).

[8] Ni, L., Li, Y., Wang, X., Zhang, J., Yu, J., & Qi, C., Forecasting of forex time series data based on deep learning. Procedia computer science, 147, 647-652, (2019).

[9] Yu, S., & Li, Z., Forecasting stock price index volatility with LSTM deep neural network. In Recent developments in data science and business analytics, 265-272, Springer, Cham (2018).

[10] Gasparin, A., Lukovic, S., & Alippi, C., Deep learning for time series forecasting: The electric load case. CAAI Transactions on Intelligence Technology, 7(1), 1-25, (2022).

[11] Gan, Z., Li, C., Zhou, J., & Tang, G., Temporal convolutional networks interval prediction model for wind speed forecasting. Electric Power Systems Research, 191, 106865, (2021).

[12] Yoo, K. M., Kil, R. M., & Youn, H. Y. (2021, January). Time series prediction based on recursive update gaussian kernel function networks. In 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM), IEEE,1-6, (2021).

[13] Kil, R. M., Park, S. H., & Kim, S., Time series analysis based on the smoothness measure of mapping in the phase space of attractors. In IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339), IEEE, vol. 4,2584-2589, (1999).

[14] [Cortes] Cortes, C., Vapnik, V., Support-vector networks. Mach Learn 20, 273–297, https://doi.org/10.1007/BF00994018, (1995).

[15] [Ber] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik., A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning

theory (COLT '92). Association for Computing Machinery, New York, NY, USA, 144–152. https://doi.org/10.1145/130385.130401, (1992).

[16] Boser, B.E., Guyon, I.M., & Vapnik, V.N., A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning, 144-152, (1992).

[17] Burges, C.J, A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121–167, (1998). https://doi.org/10.1023/A:1009715923555

[18] Drucker, H., Burges, C.J., Kaufman, L., Smola, A., & Vapnik, V.N., Support Vector Regression Machines. NIPS, 928-932, (1996).

Thông tin liên hệ: **Chu Bình Minh**
Điện thoại: 0912207854 - Email: cbminh@uneti.edu.vn
Khoa Khoa học ứng dụng, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp.