

Xây dựng và đánh giá hiệu quả các mô hình học máy vào bài toán dự báo giá cổ phiếu của 3 ngân hàng hàng đầu Việt Nam

Building and evaluating machine learning models in predicting stock prices of the top 3 banks in Vietnam

1st Trịnh Gia Huy
IS403.N21

2nd Phạm Lê Diệu Ái
IS403.N21

3rd Nguyễn Thị Thảo Hồng
IS403.N21

4th Lương Nguyễn Thành Nhân
IS403.N21

Đại học Công nghệ thông tin
Hồ Chí Minh, Việt Nam
20520556@gm.uit.edu.vn

Đại học Công nghệ thông tin
Hồ Chí Minh, Việt Nam
20520368@gm.uit.edu.vn

Đại học Công nghệ thông tin
Hồ Chí Minh, Việt Nam
20520192@gm.uit.edu.vn

Đại học Công nghệ thông tin
Hồ Chí Minh, Việt Nam
20520667@gm.uit.edu.vn

5th Lâm Võ Khánh My
IS403.N21

Đại học Công nghệ thông tin
Hồ Chí Minh, Việt Nam
20520912@gm.uit.edu.vn

Tóm tắt nội dung—

Index Terms—học máy, dự báo, chuỗi thời gian, Việt Nam, ngân hàng, cổ phiếu, chứng khoán, ARIMA, ARIMAX, SARIMAX, Linear Regression, Random Forest, RNN, LSTM, BNN, GRU, CNN-LSTM, XGBoost, BID, STB, VCB.

I. GIỚI THIỆU

Trong khoảng thời gian gần đây, ngành ngân hàng Việt Nam đã trải qua một sự phát triển mạnh mẽ, đóng góp quan trọng vào sự phát triển kinh tế của đất nước. Dữ liệu cổ phiếu ngân hàng cung cấp thông tin quan trọng về hiệu suất tài chính và biến động của thị trường, giúp hiểu rõ hơn về sự phát triển và tiềm năng của ngành ngân hàng Việt Nam. Năm 2020, thị trường chứng khoán Việt Nam đã gặp nhiều thách thức khi mà đại dịch COVID-19 xuất hiện và tác động lên thị trường Việt Nam khi mà chỉ số thị trường chứng khoán tại Việt Nam là VN-Index ghi nhận sự sụt giảm từ 991 điểm xuống 643 điểm trong Q1 2020[1] kéo theo sự phát triển của các ngân hàng. Vì thế với mục tiêu khám phá, phân tích xu hướng, biến động dữ liệu cổ phiếu và dự báo sự tăng giảm giá cả trong tương lai có sự biến động như thế nào, nhóm nghiên cứu đã chọn dự báo giá cổ phiếu của ba ngân hàng Việt Nam là Vietcombank, Sacombank, BIDV bằng cách sử dụng các mô hình học máy và học sâu để phân tích bài toán dự báo giá cổ phiếu.

Trong bài nghiên cứu này, nhóm sẽ trình bày đánh giá thực nghiệm về 11 mô hình dự báo chuỗi thời gian - time series phổ biến để dự báo giá cổ phiếu. Cụ thể, 11 mô hình dự báo là: ARIMA, ARIMAX, SARIMAX, Linear Regression, Random Forest, RNN, LSTM, BNN, GRU, CNN-LSTM, XGBoost. Việc đánh giá hiệu quả các thuật toán sẽ dựa trên 3 độ đo là RMSE, MDA, MAPE để tìm ra 2 thuật toán tốt nhất để dự báo giá của 30 ngày tiếp theo của bộ dữ liệu từ đó rút ra những nhận xét về các ưu - nhược điểm của từng thuật toán

đối với bài toán chuỗi thời gian nói chung và bài toán dự báo giá cổ phiếu nói riêng.

II. CÁC NGHIÊN CỨU LIÊN QUAN

ARIMA và SARIMAX là các mô hình dự báo sử dụng phổ biến trong bài toán chuỗi thời gian nói chung và bài toán dự báo giá cổ phiếu nói riêng, như trong việc dự báo giá cổ phiếu tại Nigeria và New York [2]. Từ kết quả thực nghiệm, ARIMA được đánh giá có thể dự báo giá cổ phiếu trong trung và ngắn hạn của chỉ số S&P BSE IT và S&P BSE Sensex trên Sàn Giao dịch Chứng khoán Bomba tại Ấn Độ[3]. Trong bài nghiên cứu chiến tranh thương mại giữa Hoa Kỳ và Trung Quốc có ảnh hưởng đến giá dầu Brent hay không. Tác giả Ilma Amira Rahmayanti và các đồng sự đã dựa vào ARIMAX và có thể kết luận rằng giá dầu Brent có bị ảnh hưởng. [4] Đối với bài nghiên cứu của Nari Sivanandam Arunraj và các cộng sự thì SARIMAX được đề xuất cho ra kết quả tốt hơn khi mà R^2 cải thiện từ 0.386 lên 0.613 so với SARIMA đối với bài toán dự báo doanh thu hàng ngày của nhà bán lẻ thực phẩm [5]. Nghiên cứu của tác giả Sreelekshmy Selvin cùng đồng sự đề xuất các mô hình deep learning và kết quả cho thấy RNN có phần trăm lỗi - error percentage lần lượt trên các công ty Infosys, TCS và Cipla là 3.90%, 7.65%, 3.83%, thấp hơn nhiều so với mô hình ARIMA là 31.91%, 21.16%, 36.53%[6].

Trong nghiên cứu của Abdullah Bin Omar và đồng sự, Random Forest cho thấy hiệu quả khá ổn định với các bộ dữ liệu quan sát nhỏ và thời gian dự báo ngắn hạn khi được so sánh với thuật toán Deep Neural Network tốt hơn khi dự báo dài hạn và tìm ra xu hướng trong thời gian dài[7]. Trong bài nghiên cứu của Yuqiao Guo, tác giả đã dự báo giá cổ phiếu của Tesla, Amazon, Microsoft bằng LSTM và kết quả dự báo khá tốt [8] Bên cạnh các phương pháp học máy đơn nhất, việc

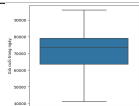
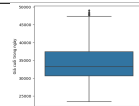
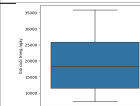
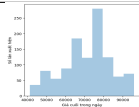
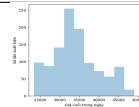
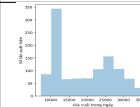
kết hợp các mô hình học máy đã được Wu và đồng sự chứng minh khi kết hợp CNN và LSTM, mô hình SACLSTM được đề xuất đã đem lại hiệu quả cao hơn so với phương pháp sử dụng riêng biệt CNN hoặc LSTM truyền thống [9]. Livieris và đồng sự cũng đề xuất 2 mô hình kết hợp CNN-LSTM cho bài toán dự báo giá vàng vào [10], mô hình đầu tiên được xây dựng với 2 lớp tích chập - convolutional layer, 1 lớp gộp - pooling và 1 lớp LSTM ít hơn mô hình thứ 2 một lớp fully-connected. Cả 2 mô hình được Livieris giới thiệu đều có hiệu quả cao hơn khi sử dụng các thuật toán khác như SVR, FFNN, LSTM đơn lẻ.

Trong nghiên cứu của

III. DỮ LIỆU

A. Tổng quan về dữ liệu

Bộ dữ liệu được sử dụng trong bài nghiên cứu dựa trên giá cổ phiếu của 3 ngân hàng hàng đầu Việt Nam là BIDV (BID)¹, Vietcombank (VCB)², Sacombank (STB)³ được thu thập từ trang Investing. Dữ liệu được lấy từ ngày 01/01/2019 đến hết ngày 01/06/2023. Dữ liệu sẽ được chia thành 3 tập train /test/ validate theo tỉ lệ 6-3-1, 7-2-1, 8-1-1 để đánh giá mô hình.

	VCB	BID	STB
Count	1101	1101	1101
Min	41161.0	23419.5	7300.0
Max	96000.0	49000.0	35850.0
Boxplot			
Histogram			
Nhận xét	-VCB là cổ phiếu có giá trị min và max cao nhất trong 3 tập dữ liệu. -Biểu đồ Boxplot cho thấy dữ liệu không có sự xuất hiện của giá trị ngoại lai và dữ liệu phân tán ít. -Biểu đồ Histogram cho thấy tần suất xuất hiện của giá cổ phiếu từ 75,000 đến 80,000 xuất hiện nhiều nhất trong tập dữ liệu.	-BID là cổ phiếu có giá trị min và max bình thường so với 2 tập dữ liệu còn lại. -Biểu đồ Boxplot cho thấy dữ liệu của BID xuất hiện nhiều giá trị ngoại lai ngoài max và dữ liệu phân tán ít. -Biểu đồ Histogram cho thấy tần suất xuất hiện của giá cổ phiếu từ 30,000 đến 35,000 xuất hiện nhiều nhất.	-STB là cổ phiếu có giá trị min và max thấp nhất trong 3 tập dữ liệu. -Biểu đồ Boxplot cho thấy dữ liệu của STB có sự phân tán lớn từ Q ₁ đến Q ₃ . -Biểu đồ Histogram cho thấy tần suất xuất hiện của giá cổ phiếu từ 10,000 đến 13,000 xuất hiện nhiều nhất.

Bảng 1

THỐNG KÊ MÔ TẢ DỮ LIỆU

B. Công cụ và thư viện

Các công cụ mà nhóm sử dụng trong bài nghiên cứu bao gồm: Visual Studio Code, Google Colab, Microsoft Excel.

Các thư viện Python mà nhóm đã sử dụng trong bài nghiên cứu bao gồm: pandas, numpy, matplotlib, seaborn, Scikit-learn, tensorflow, pmdarima, statsmodels.

C. Khai phá dữ liệu

- Mùa vụ:
- Xu hướng:
- Chu kỳ:
- Nhiều:

D. Chuẩn hóa dữ liệu

Chuẩn hóa là quá trình...

Bài nghiên cứu sử dụng kỹ thuật Min Max Scaler để chuẩn hóa dữ liệu, với: hồng chén công thức t ghi trong group vô, trích dẫn

IV. PHƯƠNG PHÁP LUẬN

A. Linear Regression

Linear Regression hay hồi quy tuyến tính là một kỹ thuật phân tích dữ liệu dùng để dự báo giá trị của một biến dựa trên giá trị của một biến khác. Cho phép thiết lập các yếu tố nào là quan trọng nhất, yếu tố nào có thể bỏ qua và cách các yếu tố đó tương tác với nhau.

Công thức cho một Linear Regression như sau:

$$y = \beta_0 + \beta_1 X + \varepsilon$$

Trong đó:

y là giá trị dự báo của biến phụ thuộc với mọi giá trị cho trước của biến độc lập

β_0 là hệ số chặn, giá trị dự báo của y khi x là 0

β_1 là hệ số hồi quy, giá trị thay đổi khi x tăng

X là biến độc lập (biến ảnh hưởng đến giá trị dự báo y)

ε là sai số của ước lượng, hay độ biến thiên có trong ước tính về hệ số hồi quy.

B. Autoregressive integrated moving average - ARIMA

ARIMA là một mô hình chuỗi thời gian được sử dụng để dự báo giá trị của một biến dựa trên các giá trị quá khứ của biến đó. Được giới thiệu bởi Mô hình này bao gồm ba thành phần chính: autoregressive (AR), integrated (I), và moving average (MA). Thành phần autoregressive mô tả mối quan hệ giữa giá trị hiện tại và các giá trị quá khứ của biến, thành phần moving average mô tả mối quan hệ giữa giá trị hiện tại và các sai số của mô hình, và thành phần integrated mô tả sự khác biệt giữa các giá trị hiện tại và các giá trị quá khứ.

Là mô hình đơn giản nhất trong

Trong phân tích hồi quy với dữ liệu chuỗi thời gian, một giả định rất quan trọng là chuỗi thời gian đang xem xét là chuỗi dừng (stationary). Một chuỗi thời gian dừng nếu trung bình (mean) và phương sai (variance) của nó không đổi qua thời gian và giá trị hiệp phương sai (covariance) giữa hai giai đoạn chỉ phụ thuộc vào khoảng cách giữa hai giai đoạn ấy chứ không phụ thuộc vào thời gian thực sự tại đó hiệp phương sai được tính, cos nhiax

¹ <https://www.investing.com/equities/commercial-bank-investment-develop>

² <https://www.investing.com/equities/joint-stock-commercial-bank>

³ <https://www.investing.com/equities/sai-gon-thuong-tin-commercial>

Mô hình ARIMA(p,d,q) tổng quát có dạng:

$$Y_t = \phi_0 + \phi_1 Y_{(t-1)} + \phi_2 Y_{(t-2)} + \dots + \phi_p Y_{(t-p)} + u_t + \theta_1 u_{(t-1)} + \dots + \theta_q u_{(t-q)} + e_t$$

Trong đó:

Y_i là chuỗi dừng bậc d của chuỗi ban đầu (chuỗi khảo sát).

ϕ là tham số tự hồi quy

θ là tham số trung bình di động

u_t là nhiễu trắng

e_t là sai số dự báo

p là bậc tự hồi quy

q là bậc trung bình trượt

C. Autoregressive Integrated Moving Average with Exogenous Variables - ARIMAX

Mô hình ARIMAX là mô hình mở rộng của ARIMA. Mô hình bao gồm các biến độc lập khác, được thêm vào cuối và được gọi là biến exogenous (biến ngoại sinh). Điều này liên quan đến việc thêm một biến bên ngoài riêng biệt khác để giúp đo lường biến endogenous (biến phụ thuộc) của mô hình. [11] ARIMAX có công thức như sau:

$$(1 - \phi_1 B - \dots - \phi_p B^p) y_t = c + \beta X_t + (1 - \theta_1 B - \dots - \theta_p B^p) \varepsilon_t$$

Trong đó:

y_t là biến phụ thuộc của chuỗi thời gian tại thời điểm t.

ε_t là sai số tại thời điểm t.

ϕ là hệ số của thành phần tự hồi quy.

θ là hệ số trung bình trượt.

B là toán tử sai phân.

D. Seasonal Autoregressive Integrated Moving Average with Exogenous Variables - SARIMAX

Mô hình SARIMAX là một dạng mở rộng của mô hình ARIMA. Mô hình ARIMA truyền thống có nhược điểm đó là không giải thích được dữ liệu có tính mùa vụ - là một chuỗi thời gian có sự lặp lại trong dữ liệu. Vì thế, mô hình SARIMA được sử dụng để giải thích các dữ liệu có tính mùa vụ nhờ bổ sung các thành phần mùa vụ vào mô hình ARIMA thông qua các tham số mới P, D, Q, s.

Trong SARIMA, s thể hiện độ dài của chu kỳ diễn ra. Ví dụ, trong dữ liệu theo quý thì $s = 4$, dữ liệu theo tháng thì $s = 12$ [12]. P, D, Q lần lượt là các thành phần autoregressive (AR), integrated (I), và moving average (MA) trong chu kỳ.

SARIMAX là sự mở rộng từ mô hình SARIMA, được tăng cường với khả năng tích hợp biến ngoại sinh để cải thiện việc dự báo của mô hình[13]. Từ bài báo của Mohamed M. Fathi và các đồng sự thì mô hình tổng quát SARIMAX(p, d, q)(P, D, Q, s) có công thức[14]:

$$\phi_p(B)\Phi_p(B^s)(1-B)^d(1-B^s)^D Y_t = c + X_t\beta + \theta_q(B)\Theta_q(B^s)\varepsilon_t \quad (1)$$

Trong đó:

Y_t là biến phụ thuộc của chuỗi thời gian tại thời điểm t.

ε_t là sai số tại thời điểm t.

c là hằng số trong mô hình.

B là toán tử sai phân ($BY_t = Y_t - 1$).

p, q, d lần lượt là các thành phần tự hồi quy, trung bình trượt và sai phân không mùa vụ.

P, D, Q lần lượt là các thành phần tự hồi quy, trung bình trượt và sai phân trong mùa vụ.

ϕ và Φ lần lượt là các hệ số của thành phần tự hồi quy và trung bình trượt không mùa vụ.

θ và Θ lần lượt là các hệ số của thành phần tự hồi quy và trung bình trượt trong mùa vụ.

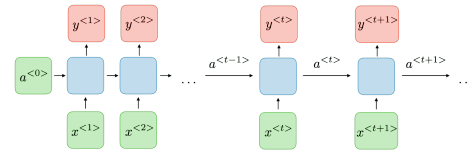
X_t là biến ngoại sinh giải thích tại thời điểm t.

β là hệ số biến ngoại sinh của biến ngoại sinh X_t .

s là độ dài chu kỳ trong chuỗi thời gian.

E. Recurrent Neural Network - RNN

Cấu trúc mạng lặp lại sử dụng đầu ra - output như là một bộ nhớ linh động, là tiền thân của mô hình RNN ngày nay, được Jordan đề xuất vào năm 1986[15]. Mô hình RNN dựa trên ý tưởng giống như mô hình mạng nơ-ron truyền thẳng. Sự khác biệt là các đầu ra của mạng nơ-ron truyền thẳng tại bất cứ thời điểm t nào, đều là hàm từ đầu vào - input tại thời điểm hiện tại cùng với trọng số - weight. Trong khi đó các đầu ra của RNN tại thời điểm t thì không chỉ dựa vào đầu vào hiện tại và weight mà còn dựa vào đầu vào trước đó, điều này khắc phục nhược điểm của mô hình mạng nơ-ron truyền thẳng là bị giới hạn bởi chúng không giải thích được sự phụ thuộc thời gian - temporal dependencies. Xem minh họa cấu trúc RNN tại hình 1.



Hình 1. Mô hình hóa thuật toán RNN

Mô hình RNN được Shuai Li và các đồng sự được mô tả theo công thức sau [16]:

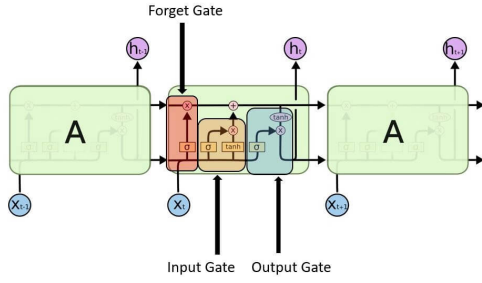
$$\mathbf{h}_t = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}) \quad (2)$$

Trong đó $\mathbf{x}_t \in \mathbb{R}^M$ và $\mathbf{h}_t \in \mathbb{R}^N$ lần lượt là các giá trị đầu vào và trạng thái ẩn tại time step. $\mathbf{W} \in \mathbb{R}^{N \times M}$, $\mathbf{U} \in \mathbb{R}^{N \times N}$ và $\mathbf{b} \in \mathbb{R}^N$ lần lượt là các weight của giá trị đầu vào và giá trị lặp lại và các bias của mạng nơ-ron. σ là hàm kích hoạt được sử dụng trong mạng, và N là số lượng nơ-ron được sử dụng trong tầng RNN hiện tại.

F. Long Short-Term Memory - LSTM

So với mạng nơ-ron truyền thống, mạng nơ-ron tái phát (RNN) có bộ nhớ trong, chúng ghi nhớ tất cả thông tin đã được lưu trữ trong quá khứ và sử dụng nó để ra quyết định trong các bước tiếp theo. Mặc dù RNN hoạt động tốt khi xử lý dữ liệu chuỗi ngắn, nhưng chúng gặp hai vấn đề chính là gradient vanishing và gradient bùng nổ gradient exploding.

Để giải quyết các vấn đề của RNN, có thể sử dụng Long Short-Term Memory (LSTM). LSTM hoạt động như một RNN cụ thể với cải tiến trong việc xác định sự phụ thuộc dài hạn trong dữ liệu chuỗi. Khác với RNN, LSTM thực hiện một cách tính toán phức tạp hơn trong việc tính toán trạng thái ẩn bằng cách thay thế các nơ-ron trong lớp ẩn truyền thống bằng các

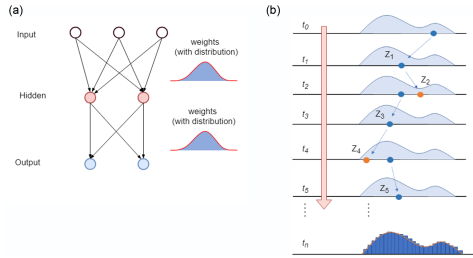


Hình 2. Kiến trúc thuật toán LSTM

ô nhớ. Chúng có khả năng điều chỉnh thông tin thông qua ba cấu trúc cổng (Input Gate, Output Gate, Forget Gate) và các cổng này cập nhật thông tin theo cách lựa chọn bằng cách học những loại thông tin trong chuỗi quan trọng để giữ lại hoặc xác định thông tin không hữu ích và loại bỏ chúng. [8]

G. Bayesian Neural Networks - BNNs

Mạng nơ-ron Bayesian cung cấp một phương pháp thực thi xác suất cho một mạng nơ-ron tiêu chuẩn với khác biệt chính là trọng số và độ lệch được biểu diễn thông qua phân phối xác suất hậu nghiệm như được thể hiện trong Hình 3. Tương tự như các mạng nơ-ron tiêu chuẩn, mạng nơ-ron Bayesian cũng có khả năng xấp xỉ hàm liên tục vô hạn. Quá trình suy luận

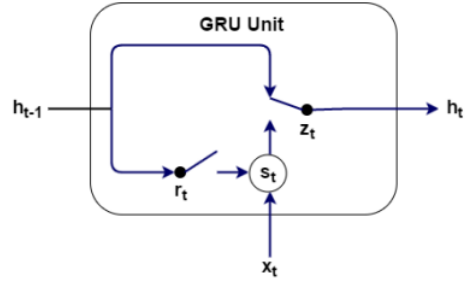


Hình 3. Kiến trúc thuật toán BNNs

bắt đầu bằng cách thiết lập phân phối tiên nghiệm (prior) trên trọng số và độ lệch. Sau đó, hệ thống lấy mẫu (chẳng hạn như MCMC) sử dụng một hàm hợp lý tính toán khẩu độ huấn luyện và chấp nhận hoặc từ chối một mẫu được đề xuất. Do các hàm kích hoạt phi tuyến tính trong mạng nơ-ron Bayesian, tính kết hợp của prior và posterior bị mất đi. Hơn nữa, do số lượng thông số lớn cho các ứng dụng khác nhau, việc thu được thông tin tiên nghiệm rất khó, do đó việc lấy mẫu phân phối xác suất hậu nghiệm cũng gặp khó khăn.

H. Gated Recurrent Unit - GRU

Grated Recurrent Unit (GRU) được đề xuất vào năm 2014 nhằm giải quyết vấn đề gradient biến mất bằng cách thay đổi cách tính toán các đơn vị ẩn. Một đơn vị ẩn h_t của GRUs được tính toán dựa trên x_t và h_{t-1} , nhưng với cơ chế cổng khác biệt. Hình 4 thể hiện cấu trúc đơn giản của một đơn vị GRU.



Hình 4. Cấu trúc của GRU

Các phương trình sau được sử dụng để tính toán một đơn vị ẩn GRU:[17]

$$z_t = \sigma(x_t U^z + h_{t-1} U^z + b^z) \quad (3)$$

$$r_t = \sigma(x_t U^r + h_{t-1} U^r + b^r) \quad (4)$$

$$s_t = \tanh(x_t U^s + (h_{t-1} \odot r_t) W^s + b^s) \quad (5)$$

$$h_t = (1 - z_t) \odot s_t + z_t \odot h_{t-1} \quad (6)$$

Hàm σ được gọi là hard sigmoid và ký hiệu \odot mang ý nghĩa phép nhân từng phần.

Các cổng GRU r_t và z_t được gọi là cổng đặt lại và cập nhật tương ứng. Cổng đặt lại quyết định cách kết hợp đầu vào mới x_t với bộ nhớ trước đó h_{t-1} để tính toán s_t . s_t có thể xem như một trạng thái ẩn "ứng cử viên". Việc tính toán của h_t sử dụng cổng cập nhật z_t để xác định bao nhiêu bộ nhớ trước đó cần được giữ lại. Một mô-đun GRU với 2 lớp ẩn để nắm bắt các tương tác đặc trưng cấp cao giữa các bước thời gian khác nhau. Các đơn vị trong lớp ẩn thứ hai được tính toán tương tự như trong lớp ẩn đầu tiên.

I. XGBoost

XGBoost hoạt động bằng phương pháp Gradient Boosting - là sự kết hợp việc sử dụng một tập hợp các mô hình yếu (weak learner) để tạo ra một mô hình mới dự báo mạnh hơn. Ý tưởng cơ bản của thuật toán gradient boosting là xây dựng một loạt các mô hình dự báo tuần tự, trong đó mỗi mô hình mới cố gắng cải thiện sai số dự báo của mô hình trước đó 5. Để làm điều này, thuật toán tối ưu hóa gradient boosting sử dụng phương pháp gradient descent.[18]

Bằng cách này, mô hình hóa thuật toán dự báo được cải thiện để xấp xỉ giá trị thực tế. Xem minh họa cấu trúc XGBoost ở Hình 5.

Về cơ bản, XGBoost được tính toán tổng quát như sau:

$l(x_1, x_2)$ - cost function

$f_i(x)$: thứ tự tại cây đang thực thi

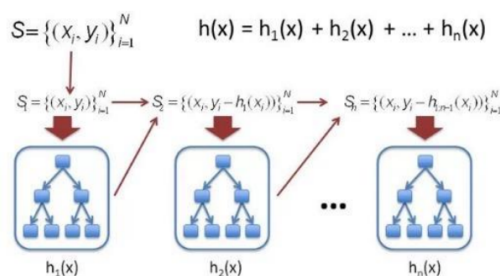
$\hat{y}_i^0 = 0$: khởi tạo giá trị dự báo ban đầu.

Giá trị đóng góp của cây thứ nhất trong việc dự báo giá trị trong việc dự báo giá trị trong mỗi mẫu dữ liệu (x_i)

$$\hat{y}_i^1 = \hat{y}_i^0 + f_1(x_i)$$

$$\hat{y}_i^2 = f_1(x_i) + f_2(x_i) = \hat{y}_i^1 + f_2(x_i)$$

Quá trình lặp đi lặp lại cho đến khi đạt được giá trị dự báo cuối cùng.

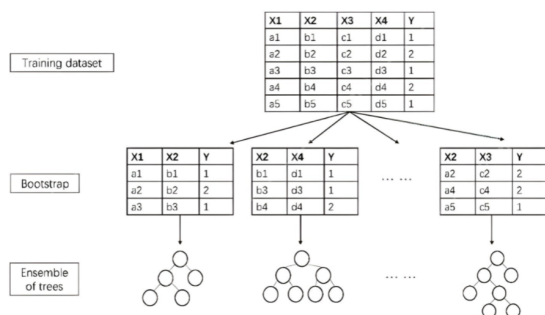


Hình 5. Mô hình hóa thuật toán hóa thuật toán XGBoost

$$\begin{aligned}\hat{y}_i^t &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \\ obj^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \\ &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant\end{aligned}$$

J. Rừng ngẫu nhiên - Random Forest

Nhằm khắc phục tình trạng overfitting của thuật toán cây quyết định, thuật toán Random forest được giới thiệu lần đầu bởi Breiman (2001)[19], tác giả đã đề xuất tạo một tập hợp các cây quyết định được xây dựng dựa trên một tập con ngẫu nhiên của các mẫu dữ liệu với các thuộc tính quyết định cũng được lựa chọn ngẫu nhiên. Dữ liệu của mỗi cây được lấy ngẫu nhiên và có thể trùng lặp(Bootstrapping) để đem lại các kết quả dự báo từ bộ dữ liệu huấn luyện độc lập nhau. Từ đó, kết quả dự báo là giá trị trung bình của tập hợp cây quyết định. Xem minh họa cấu trúc rừng ngẫu nhiên ở Hình 6.



Hình 6. Mô hình hóa thuật toán Random Forest

Trong bài nghiên cứu, nhóm sử dụng hàm Random Forest thư viện Scikit-learn với các cây được tạo nên từ thuật toán cây quyết định Classification and Regression Trees - CART[20].

Về cơ bản, rừng ngẫu nhiên về tổng quát được xây dựng như sau:

Đầu vào: Dataset (D) với N thuộc tính và n cây.

Đầu ra: Rừng ngẫu nhiên.

Lặp i=1 tới n:

Bước 1: Lấy bộ dữ liệu bootstrap(ngẫu nhiên) từ D.

Bước 2: Xây dựng rừng cây ngẫu nhiên từ bộ dữ liệu và lặp lại đến khi đạt được số node tối thiểu.

1) Chọn 1 tập con của \sqrt{N} thuộc tính.

2) Lặp từ j = 1 tới \sqrt{N} , chọn thuộc tính làm thuộc tính quyết định để chia cây tiếp tục thành 2 nhánh.

Giá trị dự báo của thuật toán là trung bình của các giá trị dự báo từ tất cả các cây quyết định.

K. CNN-LSTM

Mạng nơron tích chập (Convolutional Neural Network) là một loại mạng nơron đặc biệt để xử lý dữ liệu có cấu trúc dạng lưới-ma trận. Chẳng hạn, dữ liệu dạng chuỗi thời gian (time-series) có thể được xem là ma trận 1 chiều chứa các mẫu được lấy tại các khoảng thời gian nhất định, hay dữ liệu hình ảnh là một dạng ma trận 2 chiều.

Việc kết hợp CNN-LSTM

Trong CNN-LSTM, CNN được sử dụng như một bộ lọc nhằm rút trích ra các đặc trưng từ dữ liệu đầu vào, tiếp tục đưa vào các lớp của mô hình LSTM huấn luyện mô hình.

Mô hình CNN-LSTM được nhóm thực nghiệm bao gồm 2 phần CNN và LSTM, phần CNN gồm 6 lớp() và LSTM gồm 6 lớp(), được mô hình hóa như Hình ??.

V. THỰC NGHIỆM

A. Độ đo

a) *Root Mean Square Error - RMSE*: độ lớn trung bình sai số tuyệt đối giữa giá trị thực tế và giá trị được dự báo.

$$RMSE = \sqrt{\frac{\sum (Y_A - f)^2}{n}}$$

b) *Mean Absolute Percentage Error - MAPE*: tỉ lệ phần trăm trung bình giữa sai số tuyệt đối và giá trị thực tế.

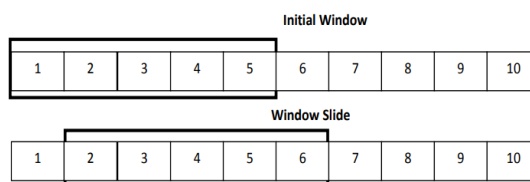
$$MAPE = \frac{1}{n} \sum \left| \frac{Y_A - f}{Y_A} \right|$$

c) *Mean Directional Accuracy - MDA*: tỉ lệ phần trăm các dự đoán xu hướng đúng so với giá trị thực tế.

$$MDA = \frac{1}{n} \sum (1_{(y_A - y_{pred_i})(y_A - f_i) > 0})$$

B. Cửa sổ trượt - Sliding window

Cửa sổ trượt là kỹ thuật xử lý dữ liệu nhằm biến đổi dữ liệu dạng chuỗi thời gian sang dạng ma trận để phù hợp với giá trị đầu vào của các mô hình học máy, học sâu. Với cửa sổ trượt kích thước n = 5, dữ liệu từ ngày 1 đến ngày 5 được sử dụng để dự báo giá trị ngày 6, tương tự với ngày 2 đến 6 dự báo ngày thứ 7 như hình 7[21]. Trong bài nghiên cứu, trừ các thuật toán Arima/Sarimax có cơ chế tự hồi quy, dữ liệu chuỗi thời gian ban đầu đều cần được áp dụng cửa sổ trượt để phù hợp với đầu vào của các mô hình học máy trong bài.



Hình 7. Tiến trình cửa sổ trượt

C. ARIMA
D. SARIMA
E. SARIMAX
F. Linear Regression
G. Random Forest
H. XGBoost
I. RNN
J. GRU
K. BNN
L. LSTM
M. CNN-LSTM

Kết quả dự báo trên cổ phiếu BID với tỉ lệ dữ liệu 8-1-1

VI. TỔNG KẾT VÀ NHẬN ĐỊNH

Từ kết quả so sánh của Bảng II, trên tập dữ liệu test, có thể thấy

Trong khi đó, trên tập dữ liệu validation như Bảng III, có sự thay đổi

Dù gặp một chút khó khăn trong việc....tuy nhiên...

VII. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

Trong tương lai,

VIII. LỜI CẢM ƠN

Cuối cùng, nhóm tác giả xin gửi lời cảm ơn đến thầy Nguyễn Đình Thuận và anh Nguyễn Minh Nhựt đã luôn hỗ trợ và góp ý để nhóm có thể hoàn thành bài nghiên cứu một cách tốt nhất.

TÀI LIỆU

- [1] N. H. Tien, R. J. S. Jose, S. E. Ullah, and H. V. Thang, "The impact of world market on ho chi minh city stock exchange in context of covid-19 pandemic," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 14, pp. 4252–4264, 2021.
- [2] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the arima model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014, pp. 106–112. DOI: 10.1109/UKSim.2014.67.
- [3] M. L. Challa, V. Malepati, and S. N. R. Kolusu, "S& BSE sensenx and s& BSE IT return forecasting using ARIMA," *Financial Innovation*, vol. 6, no. 1, Nov. 2020. DOI: 10.1186/s40854-020-00201-5. [Online]. Available: <https://doi.org/10.1186/s40854-020-00201-5>.
- [4] I. A. Rahmayanti, C. Andreas, and S. M. Ulyah, "Does US-china trade war affect the brent crude oil price? an ARIMAX forecasting approach," in *INTERNATIONAL CONFERENCE ON MATHEMATICS, COMPUTATIONAL SCIENCES AND STATISTICS 2020*, AIP Publishing, 2021. DOI: 10.1063/5.0042359. [Online]. Available: <https://doi.org/10.1063/5.0042359>.
- [5] N. S. Arunraj, D. Ahrens, and M. Fernandes, "Application of sarimax model to forecast daily sales in food retail industry," *International Journal of Operations Research and Information Systems (IJORIS)*, vol. 7, no. 2, pp. 1–21, 2016.
- [6] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 international conference on advances in computing, communications and informatics (icacci)*, IEEE, 2017, pp. 1643–1647.
- [7] A. B. Omar, S. Huang, A. A. Salameh, H. Khurram, and M. Fareed, "Stock market forecasting using the random forest and deep neural network models before and during the covid-19 period," *Frontiers in Environmental Science*, vol. 10, 2022, ISSN: 2296-665X. DOI: 10.3389/fenvs.2022.917047. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fenvs.2022.917047>.
- [8] Y. Guo, "Stock price prediction based on LSTM neural network: The effectiveness of news sentiment analysis," in *2020 2nd International Conference on Economic Management and Model Engineering (ICEMME)*, IEEE, Nov. 2020. DOI: 10.1109/icemme51517.2020.00206. [Online]. Available: <https://doi.org/10.1109/icemme51517.2020.00206>.
- [9] J. M.-T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C.-W. Lin, "A graph-based CNN-LSTM stock price prediction algorithm with leading indicators," *Multimedia Systems*, Feb. 2021. DOI: 10.1007/s00530-021-00758-w. [Online]. Available: <https://doi.org/10.1007/s00530-021-00758-w>.
- [10] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN-LSTM model for gold price time-series forecasting," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17351–17360, Apr. 2020. DOI: 10.1007/s00521-020-04867-x. [Online]. Available: <https://doi.org/10.1007/s00521-020-04867-x>.
- [11] W. K. Adu, P. Appiahene, and S. Afrifa, "VAR, ARIMA and ARIMA models for nowcasting unemployment rate in ghana using google trends," *Journal of Electrical Systems and Information Technology*, vol. 10, no. 1, Feb. 2023. DOI: 10.1186/s43067-023-00078-1. [Online]. Available: <https://doi.org/10.1186/s43067-023-00078-1>.
- [12] A. E. Permanasari, I. Hidayah, and I. A. Bustoni, "SARIMA (seasonal ARIMA) implementation on time series to forecast the number of malaria incidence," in *2013 International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, Oct. 2013. DOI: 10.1109/icitee.2013.6676239. [Online]. Available: <https://doi.org/10.1109/icitee.2013.6676239>.
- [13] S. I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, and A. G. Bakirtzis, "Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting,"

Thuật toán		VCB			STB			BID		
		6-3-1	7-2-1	8-1-1	6-3-1	7-2-1	8-1-1	6-3-1	7-2-1	8-1-1
ARIMAX	RMSE	9731.21	7049.51	8128.18	2792.92	6438.76	1611.67	8406.57	10710.78	3780.19
	MAPE	10.40	6.69	8.26	10.41	28.05	6.85	18.62	27.65	8.27
	MDA	7989.39	5284.90	6330.51	2370.64	6027.20	1286.31	6817.42	10023.63	3016.87
SARIMAX	RMSE	9604.00	7051.56	8342.83	2815.49	6462.76	1573.07	8377.70	4834.11	4181.76
	MAPE	10.27	6.70	8.51	10.51	28.16	6.67	18.57	10.77	8.69
	MDA	7987.58	5299.25	6518.43	2391.18	6054.93	1260.15	6798.17	4010.16	3184.59
ARIMA	RMSE	5785.12	13804.94	6383.53	8629.39	16389.1	3875.37	7111.1	11006.56	3089.55
	MAPE	5.27	16.82	6.73	31.75	71.07	17.23	13.94	28.7	7.39
	MDA	4254.99	12718.21	5246.55	6633.54	14686.62	3150.91	5536	10138.41	2581.55
RNN	RMSE	1546.25	1507.29	1750.61	1377.98	756.23	835.60	1101.11	852.41	835.60
	MAPE	1.54	1.49	1.87	4.14	2.83	3.72	2.28	2.16	3.72
	MDA	1198.30	1136.59	1377.98	989.91	596.75	684.77	852.41	761.19	684.77
XGBOOST	RMSE	2229.33	2126.43	2542.09	3334.50	2260.55	1408.88	3042.82	1332.55	1292.60
	MAPE	2.11	2.08	2.58	11.47	8.90	6.57	4.75	2.89	3.11
	MDA	1673.50	1605.75	1891.98	2774.59	1870.13	1157.45	1940.19	1041.42	1066.74
LINEAR	RMSE	12298.83	11783.98	12758.43	7057.09	9788.89	11188.35	5023.54	4866.87	4411.51
	MAPE	11.52	11.30	13.21	22.92	27.55	33.00	11.17	10.54	8.75
	MDA	10356.49	10013.77	11571.10	6218.10	8674.59	10408.60	4291.35	3992.55	3449.62
RF	RMSE	1797.58	1663.69	1994.44	2521.36	1873.76	874.95	2176	1174.55	1139.23
	MAPE	1.62	1.66	2.03	8.2	6.8	3.84	3.45	2.59	2.57
	MDA	1279.9	1262.71	1479.32	1985.11	1462.86	700.3	1394.22	931.02	887.43
CNN_LSTM	RMSE	3558.47	3728.23	4624.61	3589.5	1938.39	2703.02	3413.74	2993.07	2130.51
	MAPE	3.58	3.91	4.89	11.59	7.32	12.84	6.07	5.77	4.62
	MDA	2815.34	2996.24	3589.5	2784.52	1554.98	2315.35	2361.52	2130.51	1624.02
BNN	RMSE	11728.97	4951.20	7732.88	7887.56	6923.25	2277.85	15030.44	6159.61	5236.54
	MAPE	13.22	5.15	8.76	23.85	24.09	11.35	32.89	14.34	13.15
	MDA	10601.93	3801.29	6465.86	6619.41	5814.57	2248.36	12891.92	5274.22	4535.26
GRU	RMSE	1093.08	1524.86	1753.87	762.63	815.52	803.92	1041.25	1112.63	1102.92
	MAPE	2.38	1.51	1.80	2.45	3.06	3.66	2.12	2.51	2.42
	MDA	821.34	1146.68	1323.16	585.35	641.02	670.74	789.45	894.30	835.89
LSTM	RMSE	2489.00	2427.51	2789.12	1889.55	1761.77	1671.86	1641.69	1422.98	1300.86
	MAPE	2.45	2.42	2.94	5.98	6.41	7.71	3.42	3.13	2.93
	MDA	1924.14	1842.35	2166.21	1481.62	1329.87	1412.37	1304.09	1123.94	1002.65

Bảng II

BẢNG SO SÁNH HIỆU QUẢ THUẬT TOÁN TRÊN TẬP DỮ LIỆU TEST

- in 2016 *IEEE International Energy Conference (ENERGYCON)*, IEEE, Apr. 2016. DOI: 10.1109/energycon.2016.7514029. [Online]. Available: <https://doi.org/10.1109/energycon.2016.7514029>.
- [14] M. M. Fathi, A. G. Awadallah, A. M. Abdelbaki, and M. Haggag, “A new budyko framework extension using time series SARIMAX model,” *Journal of Hydrology*, vol. 570, pp. 827–838, Mar. 2019. DOI: 10.1016/j.jhydrol.2019.01.037. [Online]. Available: <https://doi.org/10.1016/j.jhydrol.2019.01.037>.
- [15] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, Mar. 1990. DOI: 10.1207/s15516709cog1402_1. [Online]. Available: https://doi.org/10.1207/s15516709cog1402_1.
- [16] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, “Independently recurrent neural network (indrnn): Building a longer and deeper rnn,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.
- [17] W. Chen, Y. Zhang, C. K. Yeo, C. T. Lau, and B. S. Lee, “Stock market prediction using neural network through news on online social networks,” in *2017 International Smart Cities Conference (ISC2)*, 2017, pp. 1–6. DOI: 10.1109/ISC2.2017.8090834.
- [18] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 785–794, ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. [Online]. Available: <http://doi.acm.org/10.1145/2939672.2939785>.
- [19] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001. DOI: 10.1023/A:1010950718922.
- [20] D. Steinberg, “Cart: Classification and regression trees,” 2009.
- [21] H. S. Hota, R. Handa, and A. K. Shrivastava, “Time series data prediction using sliding window based rbf neural network,” 2017.

Tên Model		VCB			STB			BID		
		6-3-1	7-2-1	8-1-1	6-3-1	7-2-1	8-1-1	6-3-1	7-2-1	8-1-1
ARIMAX	RMSE	31367.64	23525.82	24202.86	3761.91	1143.24	5694.94	23144.69	6837.24	11195.97
	MAPE	34.62	25.76	26.51	14.04	3.56	21.93	52.24	14.71	25.00
	MDA	31151.57	23229.47	23905.93	3605.25	879.30	5596.77	23062.78	6464.09	11069.62
SARIMAX	RMSE	31118.16	23255.97	24398.65	3683.52	1149.16	5803.20	23092.16	12515.07	11863.82
	MAPE	34.34	25.46	26.71	13.70	3.59	22.37	52.12	27.90	26.43
	MDA	30899.81	22957.19	24089.86	3519.84	886.17	5707.88	23010.21	12362.68	11711.01
ARIMA	RMSE	15009.61	4515.62	18288.14	3118.21	11821.22	3074.98	13161.95	8026.1	2309.18
	MAPE	15.1	3.73	19.61	11.89	46.76	10.71	29.27	17.49	4.5
	MDA	14368.84	3194.17	17766.06	2957.75	11782.26	2777.93	12984.64	7894	2041.57
RNN	RMSE	1214.92	1208.63	1298.02	1007.47	507.88	528.35	1098.06	629.23	528.35
	MAPE	1.02	1.01	1.11	3.52	1.51	1.54	2.05	1.00	1.54
	MDA	929.66	916.21	1007.00	899.27	386.42	393.35	928.32	454.89	393.35
XGBOOST	RMSE	4758.31	2232.49	6253.56	3144.38	983.63	919.83	7428.10	2399.64	1217.92
	MAPE	4.73	2.04	6.35	11.75	33.25	3.03	16.23	4.96	2.05
	MDA	4350.82	1848.63	5835.26	2991.85	821.41	771.03	7347.67	2219.92	941.52
LINEAR	RMSE	9359.8	6065.14	4288.56	6701.19	10261.58	8980.09	3076.04	4317.25	3782.51
	MAPE	8.76	5.16	3.40	20.76	28.75	26.07	6.57	9.82	8.41
	MDA	8594.13	4851.82	3095.81	6628.97	10216.47	8927.93	2739.76	3956.20	3438.04
RF	RMSE	4684.16	2460.58	2526.03	2185	679.1	636.46	4903.53	1497.16	1050.75
	MAPE	4.61	2.26	2.45	7.93	2.11	1.91	10.54	2.79	1.77
	MDA	4246.39	2081.47	2250.64	2016.16	536.14	490.67	4782.30	1244.43	812.64
CNN_LSTM	RMSE	5527.28	8833.62	2309.18	2543.45	1083.59	1181.71	8521.74	4119.65	2130.51
	MAPE	5.67	9.44	2.27	9.5	3.66	3.79	18.59	5.77	4.62
	MDA	5207.59	8634.7	2079.3	2425.05	936.77	960.14	8388.82	2130.51	1624.02
BNN	RMSE	11728.97	4951.20	7732.88	7887.56	6923.25	2277.85	15030.44	6159.61	5236.54
	MAPE	13.22	5.15	8.76	23.85	24.09	11.35	32.89	14.34	13.15
	MDA	10601.93	3801.29	6465.86	6619.41	5814.57	2248.36	12891.92	5274.22	4535.26
GRU	RMSE	672.90	1232.45	1315.16	526.36	545.06	533.38	988.25	1080.21	758.39
	MAPE	1.05	1.02	1.11	1.55	1.61	1.54	1.81	2.03	1.25
	MDA	478.38	926.59	1000.34	397.56	412.52	395.80	821.69	919.20	570.51
LSTM	RMSE	1856.45	1830.95	2066.50	903.49	741.72	690.96	2790.35	2737.37	738.69
	MAPE	1.59	1.58	1.88	2.73	2.46	2.20	5.93	5.70	1.23
	MDA	1448.64	1434.02	1724.20	715.81	629.38	569.64	2679.46	2573.51	556.59

Bảng III
BẢNG SO SÁNH HIỆU QUẢ THUẬT TOÁN TRÊN TẬP DỮ LIỆU VALIDATION