

Dự báo giá cổ phiếu ngân hàng Việt Nam bằng các mô hình Thống kê, Máy học và Học sâu

1st Trần Ngọc Tố Như
Khoa Hệ thống thông tin,
Trường Đại học Công nghệ thông tin-DHQG
21520385@gm.uit.edu.vn

2nd Lê Thùy Dương
Khoa Hệ thống thông tin,
Trường Đại học Công nghệ thông tin-DHQG
21520203@gm.uit.edu.vn

3rd Trần Thanh Huy
Khoa Hệ thống thông tin,
Trường Đại học Công nghệ thông tin-DHQG
21522170@gm.uit.edu.vn

4th Mai Trần Khương Duy
Khoa Hệ thống thông tin,
Trường Đại học Công nghệ thông tin-DHQG
21521998@gm.uit.edu.vn

5th Vũ Tiến Linh
Khoa Hệ thống thông tin,
Trường Đại học Công nghệ thông tin-DHQG
19521760@gm.uit.edu.vn

Tóm tắt nội dung—Việc tích hợp công nghệ thông tin vào các khía cạnh khác nhau của cuộc sống, bao gồm kinh tế, y tế và thương mại, ngày càng trở nên phổ biến. Nhất là trong các lĩnh vực được quan tâm đặc biệt, nhu cầu ứng dụng mạnh mẽ công nghệ thông tin ngày càng tăng cao, đặt ra thách thức cho những người làm trong ngành công nghệ thông tin. Một trong những bài toán được quan tâm hiện nay là việc dự đoán giá cổ phiếu của các tổ chức ngân hàng. Báo cáo này tập trung vào dự đoán giá cổ phiếu của ba ngân hàng: BIDV, VCB và MBB, sử dụng nhiều thuật toán thuộc nhiều loại khác nhau, bao gồm các thuật toán học sâu, thuật toán máy học và thuật toán thống kê (Linear Regression, Holt-Winter, ARIMA, XGBoost, Linear Regression với CalendarFourier, DeterministicProcess, MICN, CNN-LSTM, RNN, GRU, LSTM) để dự đoán, đồng thời so sánh, đánh giá kết quả khi sử dụng những thuật toán nêu trên.

Từ khóa—Stock, Linear Regression, Holt-Winter, ARIMA, XG-Boost, Linear Regression Calendar Fourier, Deterministic, RNN, GRU, LSTM, CNN-LSTM, MICN

I. GIỚI THIỆU

Trong bất kỳ quốc gia nào, thị trường chứng khoán đóng vai trò quan trọng trong nền kinh tế của mỗi quốc gia, thị trường chứng khoán Việt Nam xem là tương đối trẻ so với các quốc gia trên toàn cầu. Sự bùng nổ về trí tuệ nhân tạo và học máy đã thúc đẩy sự quan tâm của các nhà đầu tư cá nhân, tổ chức vào việc tận dụng các công nghệ này để dự đoán thị trường chứng khoán nói chung và giá cổ phiếu nói riêng tại Việt Nam.

Do đó trong nghiên cứu này, nhóm sẽ đi sâu vào việc áp dụng các thuật toán thống kê, máy học, học sâu khác nhau như Linear Regression, Holt-Winters, ARIMA, Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Extreme Gradient Boosting (XGBoost), Linear Regression áp dụng CalendarFourier, DeterministicProcess, Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM) và Multi-Channel Neural Networks (MICN), đồng thời cũng đánh giá hiệu suất của các thuật toán nêu trên.

Trong nghiên cứu này, nhóm chọn ngành ngân hàng là vì nó có tác động trực tiếp đến nền kinh tế của đất nước. Ngoài ra,

cổ phiếu ngân hàng còn đại diện cho một ngành ổn định với vốn hóa thị trường đáng kể trên sàn giao dịch chứng khoán Việt Nam. Nghiên cứu tập trung vào dự đoán giá cổ phiếu cho ba ngân hàng lớn được niêm yết trên thị trường chứng khoán Việt Nam hiện nay là BIDV, VCB và MBB.

II. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Trong những năm qua, việc xác định dữ liệu chứng khoán rất phức tạp, do đó có rất nhiều phương pháp dự đoán chuỗi thời gian đã được đề xuất. Điển hình, theo [1] nhóm tác giả đã sử dụng CNN-LSTM để đưa ra dự đoán cổ phiếu của một công ty trên thị trường chứng khoán nếu biết được những thông tin trước đó. Độ chính xác của mô hình CNN-LSTM được đánh giá là cao ngay cả khi huấn luyện trên dữ liệu thi trường chứng khoán theo thời gian thực. Bằng cách chuyển đổi dữ liệu chứng khoán thành dạng tensor (dữ liệu nhiều hơn 2 chiều) và sau đó gửi nó tới mạng thần kinh LSTM để tìm ra các mẫu. Từ đó dự đoán cổ phiếu thị trường trong một khoảng thời gian nhất định. Mặt khác, Poongodi M, Vijayakumar V và Naveen Chilamkurti đã thu thập dữ liệu về bitcoin blockchain từ ngày 28/04/2013 đến ngày 31/07/2017 có sẵn công khai trên <https://coinmarketcap.com> và áp dụng mô hình ARIMA để dự đoán giá bitcoin [2].

Các nghiên cứu trước đây đã khám phá các phương pháp khác nhau để giải quyết nhiệm vụ khó khăn của việc dự đoán giá cổ phiếu. [3] Nghiên cứu đã đề xuất một mô hình kết hợp bằng cách sử dụng mạng nơ-ron hồi quy (RNN) với Random Forest, thể hiện sự cải thiện đáng kể trong dự báo giá cổ phiếu.

Trong một bài báo khác, Xiwen Jin và Chaoran Yi đã kết luận rằng LSTM và GRU cho kết quả tương đối tốt hơn và Random Forest là tệ nhất. Điểm R2 cho các mô hình khác nhau mà các tác giả đã phân tích: LSTM 0.84, GRU 0.86, mô hình Hồi quy Random Forest 0.51, mô hình Hồi quy XGBoost 0.69, Hồi quy Tuyến tính 0.73 và mô hình Hồi quy LGBM 0.72. Từ đó có thể thấy rằng XGBoost và Random Forest có hiệu suất không tốt bằng các mô hình LSTM và GRU [4].

MICN mang lại sự cải thiện 17,2% và 21,6% cho phương pháp đa biến. Sử dụng kết hợp CNN và Transformers để hướng tới mục tiêu sử dụng thông tin tổng thể của đầu vào một cách hiệu quả. Trước tiên là trích xuất đặc trưng cục bộ của dữ liệu, sau đó lập mô hình mối tương quan toàn cầu trên cơ sở này [5].

Awajan, Ismail và Alwadi đã phát triển phương pháp EMD-HW bằng cách kết hợp Empirical Mode Decomposition và Holt-Winter để dự đoán thị trường chứng khoán. Dữ liệu chứng khoán được phân rã thành các Intrinsic Mode Functions (IMFs) và các phần dư còn lại. Tất cả các thành phần được dự báo bằng kỹ thuật Holt - Winter. Các giá trị dự báo sẽ được tổng hợp để có giá trị dự đoán cho thị trường chứng khoán. Điểm mạnh của EMD-HW này nằm ở khả năng dự đoán các chuỗi thời gian không ổn định và phi tuyến mà không cần sử dụng bất kỳ phương pháp biến đổi nào [6].

III. TẬP DỮ LIỆU

A. MÔ TẢ TỔNG QUAN

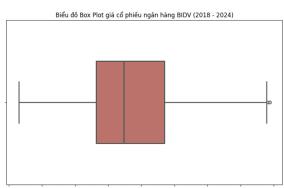
Lịch sử giá cổ phiếu của 3 ngân hàng: Ngân hàng Thương mại cổ phần Đầu tư và Phát triển Việt Nam (BIDV), ngân hàng thương mại cổ phần Ngoại thương Việt Nam (VCB), ngân hàng Thương mại cổ phần Quân đội (MBB). Dữ liệu được lấy từ ngày 1 tháng 1 năm 2018 đến ngày 1 tháng 6 năm 2024. Mỗi bộ dữ liệu chứa khoảng 1555 dòng và bao gồm 7 thuộc tính: Date, Price, Open, High, Low, Vol, Change. Trong đó:

- Date: Ngày giao dịch
- Price: Giá trị cuối cùng của cổ phiếu tại giờ đóng cửa
- Open: Giá mở cửa của cổ phiếu tại ngày giao dịch
- High: Giá cao nhất mà cổ phiếu đạt đến trong ngày
- Low: Giá thấp nhất mà cổ phiếu đạt được trong ngày
- Vol: Khối lượng giao dịch của cổ phiếu trong ngày (đơn vị: triệu cổ phiếu)
- Change: Sự chênh lệch của giá đóng cửa so với giá đóng cửa của ngày trước đó (giá trị)

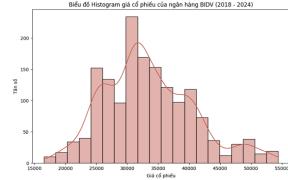
B. THỐNG KÊ MÔ TẢ

Bảng I: BIDV, VCB, MBB's Descriptive Statistics

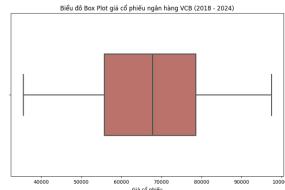
	BIDV	VCB	MBB
Count	1598	1598	1598
Mean	33466.789	67714.813	15575.501
Std	7261.683	14916.299	5360.553
Min	16531.4	35483	7206.6
25%	28166.65	55687.75	10936.525
50%	32375.3	67731	14605.7
75%	38500	78577	18786.2
Max	54400	97400	28666.7



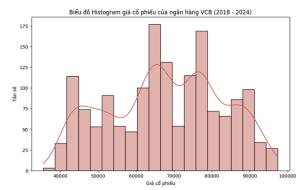
Hình 1: Box Plot of BIDV stock price (2018 - 2024)



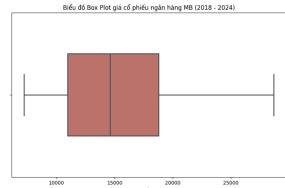
Hình 2: Histogram of BIDV Stock Price (2018 - 2024)



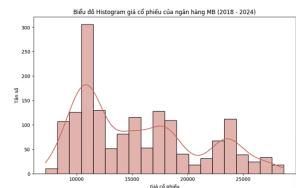
Hình 3: Box Plot of VCB Stock Price (2018 - 2024)



Hình 4: Histogram of VCB Stock Price (2018 - 2024)



Hình 5: Box Plot of MB Bank Stock Price (2018 - 2024)



Hình 6: Histogram of MB Bank Stock Price (2018 - 2024)

IV. PHƯƠNG PHÁP LUẬN

A. ARIMA

Mô hình ARIMA là sự kết hợp của quá trình hồi quy (Auto Regression – AR), quá trình trung bình trượt (Moving Average - MA) và Tính hợp sai phân (Integrated – I). Mô hình ARIMA chỉ hoạt động tốt nếu dữ liệu phụ thuộc nhiều vào thời gian và trong chuỗi dữ liệu dừng. Những dữ liệu dạng ngẫu nhiên thường ít hoạt động tốt đối với mô hình ARIMA. Mô hình ARIMA cần bộ dữ liệu có tính dừng, nếu không có tính dừng, cần tích hợp sai phân để làm bộ dữ liệu có tính dừng.

Chuỗi dừng: Một chuỗi thời gian có tính dừng là một chuỗi các giá trị Mean, Variance, Autocorrelation không thay đổi theo thời gian và nó không bao hàm yếu tố xu hướng.

Mô hình ARIMA không có tính mùa vụ: được biểu diễn với ký hiệu tiêu chuẩn được sử dụng là ARIMA(p,d,q)

- I(d) Integrated so sánh sự khác nhau giữa d quan sát, hiệu giữa giá trị hiện tại và d giá trị trước đó. Quá trình sai phân được thực hiện:

$$\text{Sai phân bậc 1: } I(1) = \Delta y_t = y_t - y_{t-1}$$

$$\text{Sai phân bậc 2: } I(2) = \Delta^2 y_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$$

Sai phân bậc d được ký hiệu là I(d)

- AR(p) Autoregression: là quá trình tìm mối quan hệ giữa dữ liệu hiện tại và p dữ liệu quá khứ trước đó gọi là (Lag) được biểu diễn với công thức.

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \dots + a_p y_{t-p} + \epsilon_t$$

Điều kiện dừng của việc chọn p: $\sum_{i=0}^p a_i < 1$

- MR(q) Moving Average là quá trình tìm mối quan hệ giữa dữ liệu hiện tại và q phần lõi quá khứ trước đó được biểu diễn với công thức

$$y_t = \beta_0 + \beta_1 \epsilon_{t-1} + \beta_2 \epsilon_{t-2} + \dots + \beta_q \epsilon_{t-q} + \mu_t$$

Điều kiện dừng của việc chọn q: $\sum_{i=0}^q \beta_i < 1$

B. Linear Regression

Hồi quy tuyến tính là một phương pháp thống kê để mô hình hóa mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Thuật toán tìm đường thẳng tốt nhất để dự đoán giá trị của biến phụ thuộc dựa trên các biến độc lập, sao cho sai

số giữa kết quả dự đoán và giá trị thực tế của biến phụ thuộc là nhỏ nhất. Mô hình hồi quy tuyến tính có hai dạng chính: hồi quy tuyến tính đơn biến và hồi quy tuyến tính đa biến.

Hồi quy tuyến tính đơn biến chỉ sử dụng một biến độc lập, trong khi hồi quy tuyến tính đa biến sử dụng nhiều hơn một biến độc lập. Hồi quy tuyến tính đơn biến là một trường hợp đặc biệt của hồi quy tuyến tính đa biến. Vậy nên mô hình hồi quy tuyến tính có công thức chung như sau: [7]

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon$$

Trong đó:

- Y là biến phụ thuộc (biến mục tiêu)
- X_1, X_2, \dots, X_k là các biến độc lập (biến giải thích)
- β_0 là hệ số giao nhau (hệ số chặn)
- $\beta_1, \beta_2, \dots, \beta_k$ là các hệ số hồi quy cho các biến độc lập
- ε là sai số (thành phần chưa giải thích bởi mô hình)

Mô hình hồi quy tuyến tính đa biến cho phép đánh giá tác động của từng biến độc lập lên biến phụ thuộc và sử dụng thông tin này để dự đoán giá trị của biến phụ thuộc dựa trên các giá trị của các biến độc lập.

C. Holt Winter

Holt Winters (HW) là một phương pháp mở rộng của phương pháp Holt, được áp dụng khi dữ liệu có xu hướng và có tính mùa vụ. Tùy thuộc vào loại mùa vụ, Holt Winters có thể là “additive” hoặc “multiplicative”, xác định dựa vào sự dao động theo thời gian. Trong cả hai phiên bản, các dự báo sẽ phụ thuộc vào ba thành phần của một chuỗi thời gian có tính mùa vụ: mức độ, xu hướng và hệ số mùa vụ của nó. [8]

Exponential smoothing (Làm mịn hàm mũ) là một kỹ thuật được sử dụng để làm mịn dữ liệu chuỗi thời gian bằng cách gán trọng số giảm dần theo hàm mũ cho các quan sát trong quá khứ, do đó làm giảm ảnh hưởng của các điểm dữ liệu cũ hơn lên kết quả tổng thể được làm mịn.

Có ba biến thể chính của Exponential Smoothing: Single Exponential Smoothing, Double Exponential Smoothing, và Triple Exponential Smoothing.

Công thức cụ thể cho Single Exponential Smoothing [9]:

$$S_t = \alpha \cdot X_t + (1 - \alpha) \cdot S_{t-1}$$

Trong đó:

- X_t là giá trị thực tại thời điểm t .
- S_t là ước tính làm mịn của mức độ vào cuối kỳ t .
- S_{t-1} là ước tính làm mịn của mức độ vào cuối kỳ $t-1$.
- α là tham số làm mịn, thường nằm trong khoảng từ 0 tới 1.

Công thức cụ thể cho Double Exponential Smoothing:

$$S_t = \alpha \cdot y_t + (1 - \alpha) \cdot (S_{t-1} + b_{t-1}) << 1$$

$$b_t = \gamma \cdot (S_t - S_{t-1}) + (1 - \gamma) \cdot b_{t-1} << 1$$

Trong đó:

- b_t là ước tính làm mịn của tốc độ tăng trưởng trung bình vào cuối kỳ t .
- γ là tham số làm mịn cho xu hướng, thường nằm trong khoảng từ 0 tới 1.
- b_{t-1} là ước tính làm mịn trước đó của xu hướng vào cuối kỳ $t-1$.
- S_t là ước tính làm mịn của mức độ vào cuối kỳ t .
- S_{t-1} là ước tính làm mịn của mức độ vào cuối kỳ $t-1$.

Triple Exponential Smoothing mở rộng từ Double Exponential Smoothing bao gồm thành phần mùa vụ để xử lý chuỗi thời gian có tính mùa vụ.[10]

$$S_t = \alpha \frac{y_t}{I_{t-L}} + (1 - \alpha)(S_{t-1} + b_{t-1})$$

Trong đó:

- S_t là quan sát đã được làm mịn tại thời điểm (t) .
- y_t là quan sát thực tế tại thời điểm (t) .
- I_{t-L} là chỉ số mùa vụ cho cùng mùa trong năm trước.
- α là hằng số cần được ước tính.
- b_t là yếu tố xu hướng tại thời điểm (t) .

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1}$$

Trong đó:

- (b_t) thể hiện thành phần xu hướng.
- (γ) là một hằng số khác cần được ước tính.

$$I_t = \beta \frac{y_t}{S_t} + (1 - \beta)I_{t-L}$$

Trong đó:

- (I_t) đại diện cho chỉ số mùa vụ tại thời điểm (t) .
- (β) là một hằng số khác cần được ước tính.

$$F_{t+m} = (S_t + mb_t)I_{t-L+m}$$

Trong đó:

- (F_{t+m}) là dự báo tại (m) kỳ tiếp theo.
- (m) là số kỳ trong tương lai.

D. Linear Regression áp dụng Calendar Fourier, Deterministic

Hồi quy tuyến tính áp dụng CalendarFourier được sử dụng để mô hình hóa tính thời vụ bằng chuỗi Fourier để xử lý các yếu tố có tính chu kỳ trong dữ liệu. Fourier sử dụng các hàm sóng sin và cos để biểu diễn một hàm tuần hoàn. Trong hồi quy tuyến tính sử dụng các hàm sóng này như các biến độc lập. Hữu ích để nắm bắt các mô hình theo mùa phức tạp.

$$Y = \beta_0 + \beta_1 \cos(\omega t) + \beta_2 \sin(\omega t) + \dots + \beta_n \cos(k\omega t) + \beta_{n+1} \sin(k\omega t)$$

Trong đó:

- $\omega = \frac{2\pi}{T}$ là tần số góc, trong đó T biểu thị khoảng thời gian.
- k là số lần lặp lại của chuỗi Fourier.

Hồi quy tuyến tính áp dụng DeterministicProcess là một phương pháp để tạo các mô hình hóa yếu tố xác định trong dữ liệu chuỗi thời gian. DeterministicProcess kết hợp các hàm thời gian tuyến tính hoặc đa thức với các thành phần Fourier. [11]

$$Y = \beta_0 + B_1 X_1 t + \beta_2 X_2 t + \dots + \beta_n X_n t$$

Trong đó:

- Y là biến phụ thuộc.
- β_0 là số hạng bị chặn hoặc số hạng không đổi.
- $B_1, \beta_2, \dots, \beta_n$ là các hệ số gắn với các biến độc lập tương ứng X_1, X_2, \dots, X_n .
- t là biến độc lập.

Biểu diễn tổng quát mô hình hồi quy tuyến tính áp dụng CalendarFourier, DeterministicProcess

$$Y = \beta_0 + \sum_{i=1}^n \beta_i X_i + \sum_{j=1}^m \left(\alpha_j \cos\left(\frac{2\pi k_j t}{T}\right) + \gamma_j \sin\left(\frac{2\pi k_j t}{T}\right) \right) + \epsilon$$

Trong đó:

- Y là biến phụ thuộc.
- X_i là các biến đặc trưng.
- $\cos\left(\frac{2\pi k_j t}{T}\right), \sin\left(\frac{2\pi k_j t}{T}\right)$ là các biến đặc trưng Fourier

- α_j, γ_j là các hệ số hồi quy có thành phần Fourier

E. XGBoost

XGBoost (eXtreme Gradient Boosting) là một mô hình máy học dựa trên Gradient Boosting nhưng được tối ưu hóa và xử lý song song giúp cải thiện đáng kể thời gian đào tạo mô hình. [12] XGBoost thực hiện tìm nhiều cây quyết định khác nhau đơn giản sau đó dùng kết quả của cây quyết định và độ lỗi trước đó làm đầu vào cho bước tìm cây quyết định tiếp theo. Sau số lần lặp hoặc ngừng chấp nhận nhất định thì dừng thuật toán. [12]

Ví dụ với tập dữ liệu n dòng và m thuộc tính, ta có:

$$D = \{(x_i, y_i) \mid i = 1, 2, \dots, n, \quad x_i \in \mathbb{R}^m, \quad y_i \in \mathbb{R}\}$$

Mô hình tập hợp cây dựa trên K cây nhỏ để dự đoán đầu ra:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F}$$

Trong đó:

- $\mathcal{F} = \{f(x) = \omega_{q(x)} \mid q : \mathbb{R}^m \rightarrow T, \quad \omega \in \mathbb{R}^T\}$ ứng với các cây hồi quy
- q đại diện cho cấu trúc mỗi cây với số lá tương ứng $\rightarrow T$.
- Mỗi f_k ứng với một cấu trúc cây độc lập q và trọng số lá ω
- ω_i đại diện cho lá thứ i . Đổi với từng dòng dữ liệu, sử dụng quy tắc cây quyết định để phân loại và tính toán cộng điểm các lá tương ứng.

Để tối ưu qua từng cây được sử dụng trong mô hình, tối ưu hóa mục tiêu điều chỉnh như sau:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

Trong đó: $\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2$

Với:

- l : Hàm mất mát.
- Ω : Hàm tính độ phức tạp.
- T : Số lượng lá.
- ω : Trọng số lá.
- γ : Tham số điều chỉnh số lượng lá.
- λ : Tham số điều chỉnh chuẩn $L2$ của các điểm số lá.

Việc tính toán $L(\phi)$ bao gồm các tham số được huấn luyện bổ sung, để dự đoán $\hat{y}_i^{(t)}$ ở dòng dữ liệu thứ i và bước lặp thứ t , cần thêm f_t để tối ưu hóa mục tiêu.

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

$$L^{(t)} \approx \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

Với:

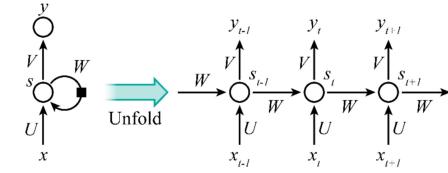
- $g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$: Gradient descent bậc 1.
- $h_i = \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)})$: Gradient descent bậc 2.

F. RNN

RNN (Mạng nơ-ron hồi quy) là một mạng nơ-ron kết hợp dữ liệu đầu vào có độ dài biến đổi trong khi có trạng thái ẩn, trạng thái này phụ thuộc vào các bước thời gian trước đó để tạo ra dữ liệu đầu ra. Thông qua các kết nối giữa các đơn vị ẩn liên quan đến độ trễ thời gian, mô hình có thể ghi nhớ thông tin từ quá khứ, cho phép nó nắm bắt các tương quan theo thời gian giữa

các sự kiện cách xa nhau trong dữ liệu.[13]

Mô hình chuẩn của RNN được minh họa như trong hình dưới đây:



Hình 7: Mô hình RNN chuẩn (Nguồn: [14])

Trạng thái ẩn S_t tại bước t được tính dựa trên đầu vào X_t tại bước t và trạng thái ẩn S_{t-1} tại bước trước:

$$s_t = f(U_{x_t} + W_{s_{t-1}})$$

Trong đó:

- s_t : Trạng thái ẩn tại thời điểm t .
- x_t : Đầu vào tại thời điểm t .
- s_{t-1} : Trạng thái ẩn tại thời điểm $t - 1$.
- U : Ma trận trọng số từ đầu vào đến trạng thái ẩn.
- W : Ma trận trọng số từ trạng thái ẩn trước đến trạng thái ẩn hiện tại.
- f : Hàm kích hoạt (thường là hàm tanh hoặc sigmoid).

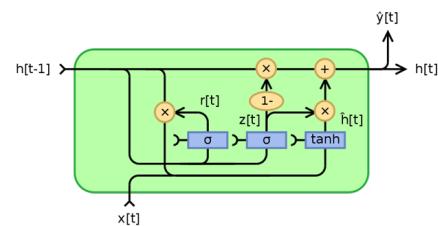
$$y_t = \sigma(V_{s_t})$$

Trong đó:

- y_t : Đầu ra tại thời điểm t .
- s_t : Trạng thái ẩn tại thời điểm t .
- V : Ma trận trọng số từ trạng thái ẩn đến đầu ra.
- σ : Hàm kích hoạt.

G. GRU

Kyunghyun Cho giới thiệu GRU vào năm 2014, là một thuật toán dựa trên RNN tương tự như LSTM nhưng có cấu trúc đơn giản hơn. RNN gặp phải vấn đề Vanishing và Exploding Gradient trong quá trình lan truyền ngược qua thời gian (Back-propagation Through Time). GRU giải quyết vấn đề này bằng cách sử dụng hai cổng: cổng cập nhật và cổng đặt lại. Khác với LSTM, GRU không duy trì trạng thái tế bào (cell state) bên trong và tích hợp thông tin từ trạng thái tế bào vào trạng thái ẩn (hidden state) của nó[15].



Hình 8: Kiến trúc của GRU

Reset gate: xác định có bao nhiêu thông tin trong quá khứ phải được giữ lại.

$$r_t = \sigma(W_r x_t + [h_{t-1}, x_t] + b_r)$$

Update gate: xác định lượng thông tin trước đó sẽ bị xóa và kết hợp đầu vào với thông tin cũ.

$$z_t = \sigma(W_z x_t + [h_{t-1}, x_t] + b_z)$$

Candidate hidden state: được Reset gate sử dụng để giữ lại thông tin quan trọng từ quá khứ.

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h)$$

Hidden state: là đầu ra của quá trình.

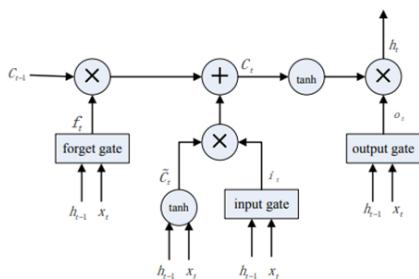
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Trong đó:

- W_r, W_z, W_h là những ma trận trọng số.
- x_t là đầu vào tại bước thời gian t.
- h_{t-1} là trạng thái ẩn trước đó.
- h_t là trạng thái ẩn hiện tại.

H. LSTM

LSTM (Long short-term memory) là một phiên bản cải tiến của RNN có một cấu trúc phức tạp được gọi là ô LSTM (LSTM cell) này trong lớp ẩn (hidden layer) của nó. Khối LSTM có 3 cổng tên là cổng vào (input gate), cổng quên (forget gate) và cổng ra (output gate). Ba cổng này có tác dụng điều khiển các luồng thông tin đi qua các ô cũng như mạng nơron. [16]



Hình 9: Kiến trúc của LSTM

Kiến trúc LSTM cần truyền trạng thái ra của ô C_t (cell output state) và đầu ra lớp ẩn h_t (hidden layer output) tới các nơron tiếp theo trong mạng. Để tính các giá trị này tại thời điểm t ta thực hiện các bước theo thứ tự:

Bước 1: Tính trạng thái của 3 cổng, trạng thái đầu vào của ô:

- Input gate: $i_t = \sigma(W_1^i \cdot x_t + W_h^i \cdot h_{t-1} + b_i)$
- Forget gate: $f_t = \sigma(W_1^f \cdot x_t + W_h^f \cdot h_{t-1} + b_f)$
- Output gate: $o_t = \sigma(W_1^o \cdot x_t + W_h^o \cdot h_{t-1} + b_o)$
- Đầu vào của ô: $\tilde{C}_t = \tanh(W_1^C \cdot x_t + W_h^C \cdot h_{t-1} + b_C)$

Trong đó:

- x_t là dữ liệu đầu vào tại thời điểm t
- h_t là đầu ra của lớp ẩn, h_{t-1} là đầu ra của trước đó
- \tilde{C}_t là trạng thái đầu vào của ô
- $W_1^i, W_1^f, W_1^o, W_1^C$ là các ma trận trọng số để kết nối x_t tới 3 cổng và đầu vào của ô
- $W_h^i, W_h^f, W_h^o, W_h^C$ là các ma trận trọng số để kết nối h_{t-1} tới 3 cổng và đầu vào của ô
- b_i, b_f, b_o, b_C là các chỉ số bias

- σ là hàm sigmoid với $\sigma = \frac{1}{1+\exp(-x)}$
- \tanh là hàm hyperbolic tangent với $\tanh = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$

Bước 2: Tính toán trạng thái đầu ra của ô:

$$C_t = i_t \cdot \tilde{C}_t + f_t \cdot C_{t-1}$$

Trong đó:

- C_t là trạng thái đầu ra của ô
- C_{t-1} là trạng thái đầu ra của trước đó

Bước 3: Tính toán đầu ra của lớp ẩn:

$$h_t = o_t \cdot \tanh(C_t)$$

I. CNN-LSTM

CNN-LSTM (Convolutional Neural Network - Long Short-Term Memory) kết hợp hai kiến trúc mạng nơ-ron quan trọng trong lĩnh vực xử lý dữ liệu hình ảnh và chuỗi. [1]

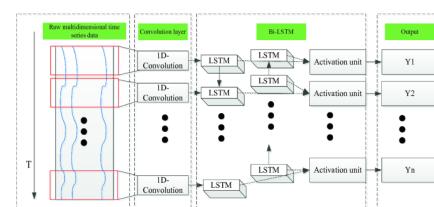
CNN-LSTM gồm 2 phần là CNN và LSTM:

1) *Convolutional Neural Network (CNN)*: Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một loại mạng nơron đặc biệt để xử lý dữ liệu có cấu trúc dạng lưới-ma trận. Có 3 loại CNN: 1D CNNs dùng trên dữ liệu chuỗi thời gian, 2D CNNs dùng để phân loại hình ảnh, 3D CNNs thường dùng trong xử lý ảnh 3 chiều hoặc video. Một số lớp cơ bản của CNN:

- Convolutional layer: sử dụng các filter để thực hiện phép tích chập. Các siêu tham số của các filter này bao gồm kích thước bộ lọc F (filter) và độ trượt S (stride). Output của convolutional layer sẽ qua hàm activation function trước khi trở thành input của convolutional layer tiếp theo.
- Pooling layer: thường được dùng giữa các lớp Convolutional layer để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Việc giảm kích thước dữ liệu giúp giảm tính toán trong mô hình. Có 2 loại Pooling Layer phổ biến là: Max Pooling và Average Pooling.
- Fully connected layer: Sau khi dữ liệu được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của dữ liệu. Fully connected layer nhận đầu vào là các dữ liệu đã được làm phẳng mà mỗi node trong hidden layer được kết nối với tất cả các node trong layer trước.

2) *Long Short-Term Memory (LSTM)*: Đã được nhắc đến trong mục LSTM bên trên.

3) *Kiến trúc mô hình CNN-LSTM*: Kiến trúc CNN-LSTM liên quan đến việc sử dụng các lớp CNN để trích xuất các đặc trưng cục bộ từ dữ liệu đầu vào, sau đó đưa các đặc trưng vào các lớp LSTM để mô hình có thể hiểu các mối quan hệ không gian và thời gian giữa các đặc trưng này. [1].



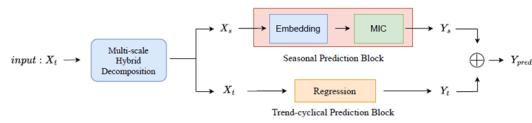
Hình 10: Kiến trúc của CNN-LSTM (Nguồn: [17])

Kiến trúc mô hình CNN-LSTM bao gồm các thành phần sau:

- Input Layer: Nhận dữ liệu đầu vào, ví dụ giá đóng cửa, giá sàn,...
- CNN Layer: Sử dụng các lớp tích chập để trích xuất đặc trưng giúp mô hình học được những đặc trưng quan trọng cho dự đoán giá cổ phiếu.
- LSTM Layer: Nhận các đặc trưng đã được trích xuất từ CNN và sử dụng các memory cell để xử lý và mô hình hóa dữ liệu chuỗi. LSTM giúp mô hình ghi nhớ thông tin lịch sử quan trọng và dự đoán dựa trên quá khứ.
- Output Layer: Tạo ra dự đoán về giá cổ phiếu dựa trên trạng thái ẩn cuối cùng của mô hình.

J. MICN

MICN (Multi-Scale Isometric Convolution Network) là mô hình giúp dự đoán hiệu quả và tiết kiệm chi phí cho dự đoán dài hạn bằng cách sử dụng phương pháp tách nhau đầu vào và chọn các thuật toán thích hợp để xử lý từ phần nhỏ. Kiến trúc tổng quan của mô hình MICN được thể hiện như sau: [5]



Hình 11: Kiến trúc của MICN

Mô hình bao gồm 3 phần chính: một khối phân rã đa tầng kết hợp (Multi-scale hybrid decomposition – MHDecomp), một khối dự đoán cho dữ liệu có tính mùa vụ, một khối dự đoán có dữ liệu có tính xu hướng theo chu kỳ. Sau khi xử lý từng phần xong mô hình sẽ kết hợp kết quả của 2 quá trình lại để cho ra kết quả cuối cùng.

1) **MULTI-SCALE HYBRID DECOMPOSITION:** Khối phân rã đa tầng kết hợp được xây dựng dựa theo thuật toán phân rã của Haixu Wu là áp dụng trung bình trượt để làm mượt các biến động nhỏ và làm nổi bật các xu hướng dài hạn. Tuy nhiên thuật toán này có điểm yếu khi sử dụng tham số kernel nên ở mô hình MICN đã cải tiến bằng cách sử dụng nhiều kernel khác nhau kết hợp với việc sử dụng giá trị trung bình để xử lý. Với dữ liệu đầu vào là $X \in \mathbb{R}^{I \times d}$, quá trình phân rã được xử lý như sau:

$X_t = \text{AvgPool}(\text{Padding}(X))_{\text{kernel}_1}, \dots, \text{AvgPool}(\text{Padding}(X))_{\text{kernel}_n}$

Trong đó:

- X_t là phần dữ liệu có xu hướng chu kỳ.
- X_s là phần dữ liệu có tính mùa vụ.

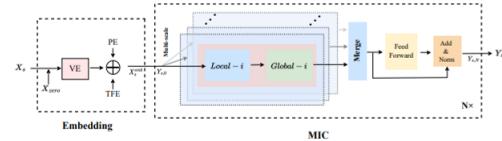
2) **TREND-CYCLICAL PREDICTION BLOCK:** Mô hình MICN sử dụng thuật toán hồi quy tuyến tính đơn giản để dự đoán cho dữ liệu có xu hướng theo chu kỳ. Cụ thể với dữ liệu có tính xu hướng theo chu kỳ được xử lý bằng cách:

$$Y_t^{\text{regre}} = \text{regression}(X_t)$$

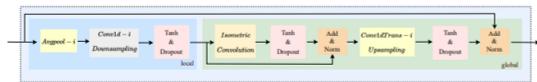
Trong đó $Y_t^{\text{regre}} \in \mathbb{R}^{O \times d}$ là kết quả dự đoán của phần này dựa trên thuật toán hồi quy tuyến tính.

3) **SEASONAL PREDICTION BLOCK:** Khối dự đoán cho dữ liệu có tính mùa vụ tập trung vào việc xử lý phần dữ liệu có tính mùa vụ phức tạp. Đầu tiên thực hiện quá trình Embedding chuỗi đầu vào X_s , thuật toán tiếp tục áp dụng “Multi-scale isometric convolution” (MIC) để lấy ra những đặc trưng cục bộ và tương

quan toàn thể, và các nhánh với các tỷ lệ khác nhau, tức là các mạng con hoặc các tham số khác nhau được sử dụng để mô hình hóa các mẫu cơ bản khác nhau trong chuỗi thời gian. Mỗi nhánh có thể tập trung vào việc xác định mẫu ở một mức độ chi tiết khác nhau. Cuối cùng kết hợp các kết quả từ các nhánh khác nhau để hoàn thiện việc sử dụng thông tin toàn diện của chuỗi.



Hình 12: Khối dự đoán dữ liệu mùa vụ



Hình 13: Kiến trúc module cục bộ-toàn cầu

Các quá trình được tổng kết lại bằng các công thức sau:

$$\begin{aligned} X_s^{\text{emb}} &= \text{Embedding}(\text{Concat}(X_s, X_{\text{zero}})) \\ Y_s^0 &= X_s^{\text{emb}} \\ Y_{s,l} &= \text{MIC}(Y_{s,l-1}), \quad l \in \{1, 2, \dots, N\} \\ Y_s &= \text{Truncate}(\text{Projection}(Y_{s,N})) \end{aligned}$$

Trong đó:

- $X_{\text{zero}} \in \mathbb{R}^{O \times d}$: các chỗ trống được điền bằng zero
- $X_s^{\text{emb}} \in \mathbb{R}^{(I+O) \times D}$: các dữ liệu đã được nhúng từ X_s
- $Y_{s,l} \in \mathbb{R}^{(I+O) \times D}$: kết quả dự đoán của lớp MIC thứ l
- Y_s : kết quả dự đoán cuối cùng cho dữ liệu có tính mùa vụ

V. KẾT QUẢ

A. CÁC PHƯƠNG PHÁP ĐÁNH GIÁ

MAPE (Mean Absolute Percentage Error): tính toán sự sai lệch trung bình theo tỷ lệ phần trăm giữa giá trị dự báo và giá trị thực tế. Tuy nhiên khi giá trị thực tế gần bằng 0, MAPE có thể dẫn đến phép chia cho 0 gây sai lệch trong kết quả đánh giá.

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100\%$$

Root Mean Squared Error (RMSE): là độ lệch chuẩn của các phần dư (sai số dự đoán). RMSE được tính bằng cách lấy căn bậc hai của trung bình của bình phương các sai số giữa giá trị dự đoán và giá trị thực tế. Tuy nhiên, RMSE có thể bị ảnh hưởng bởi các giá trị ngoại lai (outliers) trong dữ liệu.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Mean Absolute Error (MAE): dùng để so sánh hiệu suất và đánh giá mức độ chính xác của các mô hình dự báo. MAE đo lường độ lỗi trung bình tuyệt đối giữa giá trị dự đoán và giá trị thực tế. MAE càng nhỏ thì mô hình dự đoán càng chính xác.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Trong đó:

- n : số lượng mẫu
- y_i : giá trị thực tế của mẫu thứ i
- \hat{y}_i : giá trị dự đoán tương ứng của mô hình cho mẫu thứ i

B. TRỰC QUAN HÓA

I) ĐÁNH GIÁ TRÊN BỘ DỮ LIỆU BIDV:



Hình 14: LR 8:2

BỘ DỮ LIỆU BIDV:



Hình 15: LR 9:1

Hình 16: Holt-Winter 8:2



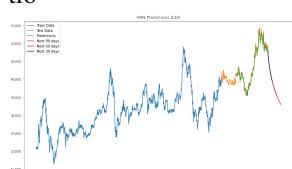
Hình 18: ARIMA 8:2



Hình 22: Linear Regression CalendarFourier, DeterministicProcess 8:2



Hình 24: RNN with 8:2 ratio



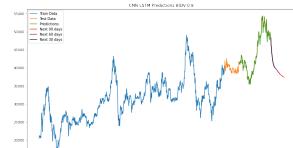
Hình 26: RNN 8:2



Hình 28: LSTM 8:2



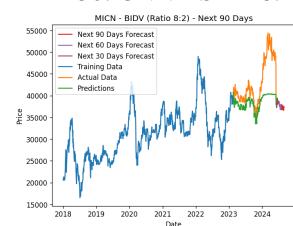
Hình 29: LSTM 9:1



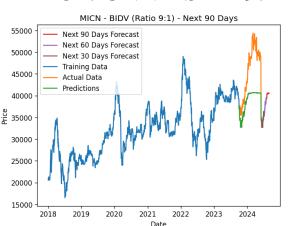
Hình 30: CNN-LSTM 8:2



Hình 31: CNN-LSTM 9:1



Hình 32: MICN 8:2



Hình 33: MICN 9:1



Hình 34: LR 8:2



Hình 35: LR 9:1



Hình 36: Holt-Winter 8:2



Hình 37: Holt-Winter 9:1



Hình 38: ARIMA 8:2



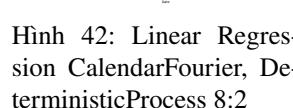
Hình 39: ARIMA 9:1



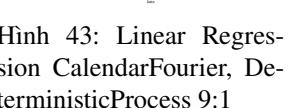
Hình 40: XGBoost 8:2



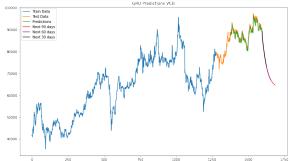
Hình 41: XGBoost 9:1



Hình 42: Linear Regression CalendarFourier, DeterministicProcess 8:2



Hình 43: Linear Regression CalendarFourier, DeterministicProcess 9:1



Hình 44: GRU 8:2



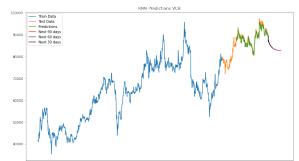
Hình 45: GRU 9:1



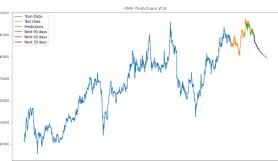
Hình 60: XGBoost 8:2



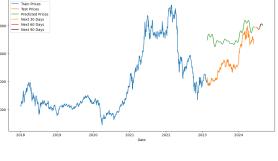
Hình 61: XGBoost 9:1



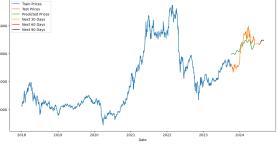
Hình 46: RNN 8:2



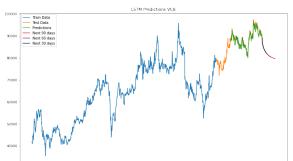
Hình 47: RNN 9:1



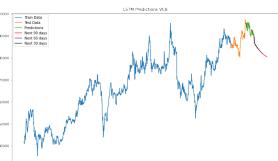
Hình 62: Linear Regression
CalendarFourier, DeterministicProcess 8:2



Hình 63: Linear Regression
CalendarFourier, DeterministicProcess 9:1



Hình 48: LSTM 8:2



Hình 49: LSTM 9:1



Hình 64: GRU 8:2



Hình 65: GRU 9:1



Hình 50: CNN-LSTM 8:2



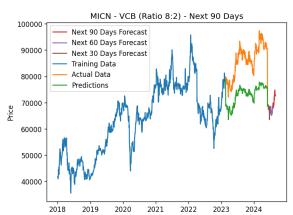
Hình 51: CNN-LSTM 9:1



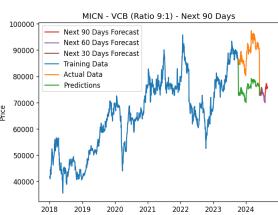
Hình 66: RNN 8:2



Hình 67: RNN 9:1



Hình 52: MICN 8:2



Hình 53: MICN 9:1



Hình 68: LSTM 8:2



Hình 69: LSTM 9:1



Hình 54: LR 8:2



Hình 55: LR 9:1



Hình 70: CNN-LSTM 8:2



Hình 71: CNN-LSTM 9:1



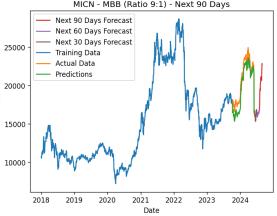
Hình 56: Holt-Winter 8:2



Hình 57: Holt-Winter 9:1



Hình 72: MICN 8:2



Hình 73: MICN 9:1



Hình 58: ARIMA 8:2



Hình 59: ARIMA 9:1

C. SO SÁNH, ĐÁNH GIÁ

Bảng II: ĐÁNH GIÁ TRÊN CÁC BỘ DỮ LIỆU

Tỉ lệ	Mô hình	BIDV			VCB			MBB		
		RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
7:3	Linear Regression	5106.099	4029.223	10.7	13314.847	11406.7	15.909	7754.886	7467.216	44.473
	Holt-Winter	12147.353	10425.494	24.37	19660.67	16922.62	19.86	3030.488	2326.928	12.583
	ARIMA	11644.413	9900.365	23.084	18989.785	16338.678	19.18	2971.378	2288.123	12.492
	XGBoost	1531.03	782.56	1.69	1223.78	793.18	0.96	285.05	217.39	1.2
	Linear									
	CalendarFourier, DeterministicProcess	4643.148	3800.978	10.1	13837.12	11888.17	16.518	7823.866	7539.177	44.75
	RNN	848.758	610.66	1.444	1238.266	903.622	1.1			
	GRU	799.344	565.852	1.3439	1505.262	1187.511	1.42	322.462	230.585	1.284
	LSTM	1140.567	943.25	2.215	1626.77	1346.2	1.636	364.172	262.703	1.48
	CNN-LSTM	818.416	579.635	1.374	1456.85	1133.341	1.349	357.54	269.474	1.515
8:2	MICN	4768.2027	3880.841	8.99	7171.965	6384.823	7.61	1455.6243	1397.8943	8.16
	Linear Regression	5959.578	4485.368	9.618	4906.916	4224.515	4.829	3895.731	4385.594	22.978
	Holt-Winter	5442.702	4004.039	8.67	9729.256	8419.695	9.45	4675.115	3715.823	18.094
	ARIMA	5465.229	3899.99	8.384	9568.17	8264.83	9.26	4690.713	3730.365	18.162
	XGBoost	1835.20	1056.32	2.19	1481.31	894.92	0.99	304.36	196.86	0.98
	Linear									
	CalendarFourier, DeterministicProcess	5034.1	3829.7	8.3	5144.08	4441.966	5.133	4836.991	4364.947	25.582
	RNN	903.898	652.472	1.44	1525.8	1123.53	1.249			
	GRU	778.219	550.714	1.233	1292.515	964.056	1.076	316.972	224.028	1.097
	LSTM	863.893	631.665	1.41	1170.171	869.401	0.969	309.524	219.665	1.075
9:1	CNN-LSTM	819.226	581.721	1.301	342.531	251.857	1.235	333.924	245.384	1.201
	MICN	5851.538	4515.326	9.73	13536.513	13263.265	15.24	700.2225	619.9381	3.57
	Linear Regression	7397.376	6358.285	13.152	4402.437	3575.22	3.911	2318.005	2037.823	10.3597
	Holt-Winter	9974.911	8400.13	17.107	6184.328	5317.27	5.84	3873.053	3092.189	13.75
	ARIMA	10578.949	8939.392	18.199	5997.686	4922.698	5.36	3928.862	3100.638	13.732
	XGBoost	2631.97	1757.3	3.47	1571.18	988.49	1.06	360.88	260.76	1.21
	Linear									
	CalendarFourier, DeterministicProcess	5719.048	4611.46	9.4	4474.24	3779.58	4.179	1851.396	1628.438	8.37
	RNN	1035.685	763.267	1.489	1076.603	801.1	0.856	466.864	370.408	1.604
	GRU	946.578	644.264	1.262	879.522	684.756	0.735	443.137	351.169	1.516
	LSTM	991.127	723.549	1.418	848.022	649.84	0.696	424.367	329.62	1.421
Cuối cùng, thực hiện trực quan kết quả dự báo bằng biểu đồ, trong đó chứa thông tin dự báo trong 30-60-90 ngày tiếp theo trong tương lai.	CNN-LSTM	476.62	375.372	1.633	1153.785	879.134	0.943	542.099	441.47	1.907
	MICN	7823.79	6982.51	14.54	14089.16	13939.425	15.53	1057.8475	952.3056	4.84

VI. KẾT LUẬN

A. KẾT LUẬN TỔNG THỂ

Cổ phiếu mang đến những cơ hội đầu tư tuyệt vời nhưng cũng đi kèm với những rủi ro do tính chất khó lường của chúng. Tuy nhiên, những tiến bộ trong các phương pháp thống kê, thuật toán học máy và học sâu đã mở đường cho sự phát triển của các mô hình dự đoán có độ chính xác cao.

Kết quả nghiên cứu của nhóm cho thấy thuật toán XGBoost, GRU và LSTM là phù hợp nhất trong việc dự đoán giá cổ phiếu trong tương lai của 3 ngân hàng BIDV, VCB, MBB.

Cụ thể, XGBoost dự báo tốt nhất cho ngân hàng MBB trong mọi tỉ lệ. Theo sau là thuật toán LSTM và GRU.

Đối với dữ liệu ngân hàng VCB, các kết quả có sự phân bố đa dạng ở các mô hình và tỉ lệ phân chia tập dữ liệu. Trong đó mô hình XGBoost đạt hiệu suất tốt nhất trên tỉ lệ 7:3, theo sau là mô hình RNN. Mô hình LSTM tốt nhất trên cả 2 tỉ lệ 8:2 và 9:1. Đối với dữ liệu ngân hàng BIDV, GRU lại cho kết quả tốt nhất ở cả ba tỉ lệ. Theo sau là CNN-LSTM ở 2 tỉ lệ 7:3 và 8:2.

Cuối cùng, thực hiện trực quan kết quả dự báo bằng biểu đồ, trong đó chứa thông tin dự báo trong 30-60-90 ngày tiếp theo trong tương lai.

B. KHÓ KHĂN, THÁCH THỨC

Mặc dù trong quá trình thực hiện đồ án nhóm đã cố gắng tìm hiểu đào sâu lý thuyết để triển khai các mô hình một cách tốt nhất có thể nhưng nhóm nghiên cứu vẫn nhận sự bản thân còn thiếu kinh nghiệm và kiến thức để triển khai các thuật toán đó một cách hoàn chỉnh nhất.

- Chưa thông thạo kiến thức về lập trình: Trong quá trình thực hiện đồ án, ngôn ngữ lập trình được sử dụng là Python - một ngôn ngữ mà nhóm nghiên cứu lần đầu sử dụng vì vậy việc vừa tìm hiểu ngôn ngữ mới vừa áp dụng nó để triển khai các thuật toán là một thách thức không nhỏ đối với nhóm nghiên cứu.
- Nguồn dữ liệu có tính phù hợp khác nhau với mỗi thuật toán: Nhóm nghiên cứu nhận thấy với đa số các mô hình nhóm thực hiện dữ liệu của ngân hàng MBB các tính phù hợp cao nhất, cho ra những kết quả đánh giá tốt nhất, còn với 2 ngân hàng còn lại vẫn có những mô hình chứ thế dự đoán ra kết quả tốt, vậy nên có thể phải xử lý những dữ liệu của các ngân hàng tốt hơn để thu được kết quả tốt cho các mô hình và dữ liệu khác nhau.
- Việc sử dụng một thuộc tính giá duy nhất để dự đoán giá

cỗ phiếu khi trên thực tế có rất nhiều yếu tố tác động dẫn đến việc dữ liệu dự đoán cho tương lai có phần chưa chính xác với thực tế cũng là một thách thức cần được xem xét cho nhóm nghiên cứu.

C. ĐỊNH HƯỚNG TRONG TƯƠNG LAI

Bằng kinh nghiệm thực hiện đồ án môn Phân tích dữ liệu kinh doanh, chúng em sẽ tìm hiểu sâu hơn về cách các thuật toán hoạt động, kết hợp những điểm mạnh của các thuật toán để áp dụng với đúng bộ dữ liệu phù hợp.

Nhóm sẽ tập trung cải thiện hiệu suất của các mô hình qua việc tinh chỉnh các siêu tham số cũng như kiến trúc lớp của những mô hình như CNN-LSTM, MICN để nâng cao tính chính xác của mô hình. Nhóm cũng sẽ thực nghiệm thêm với những mô hình mới và thêm nhiều bộ dữ liệu, nhất là với những dữ liệu biến động do các yếu tố bất ngờ (kinh doanh thua lỗ, nợ xấu) để tăng tính thực tế.

Với mục tiêu đảm bảo các mô hình có độ chính xác cao hơn và phù hợp với từng bộ dữ liệu.

D. LỜI CẢM ƠN

Với lòng biết ơn sâu sắc nhất, nhóm xin gửi lời cảm ơn chân thành và sâu sắc tới PGS. TS. Nguyễn Đình Thuân, GVTG Nguyễn Minh Nhựt đã tận tâm hướng dẫn, hỗ trợ qua từng buổi học trên lớp, giải đáp kịp thời các thắc mắc của nhóm, đưa ra những góp ý để nhóm có thể hoàn thành đồ án môn học và đạt được kết quả đê ra.

TÀI LIỆU

- [1] A. A. R. R. V. R. S **and** A. M. Bagde, *Predicting Stock Market Time-Series Data using CNN-LSTM Neural Network Model*, 2023. arXiv: 2305.14378 [q-fin.ST].
- [2] P. Manoharan, V. Vijayakumar **and** N. Chilamkurti, “Bitcoin price prediction using ARIMA model,” *International Journal of Internet Technology and Secured Transactions*, **jourvol** 10, **page** 396, **january** 2020. DOI: 10.1504/IJITST.2020.108130.
- [3] Y. Ma, R. Han **and** X. Fu, “Stock prediction based on random forest and LSTM neural network,” **october** 2019, **pages** 126–130. DOI: 10.23919/ICCAS47443.2019.8971687.
- [4] X. Jin **and** C. Yi, “The Comparison of Stock Price Prediction Based on Linear Regression Model and Machine Learning Scenarios,” **indecember** 2022 **pages** 837–842, ISBN: 978-94-6463-029-9. DOI: 10.2991/978-94-6463-030-5_82.
- [5] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen **and** Y. Xiao, “MICN: Multi-scale Local and Global Context Modeling for Long-term Series Forecasting,” **inThe Eleventh International Conference on Learning Representations** 2023. **url:** <https://openreview.net/forum?id=zt53IDUR1U>.
- [6] A. Awajan, M. T. Ismail **and** S. Alwadi, “Stock market forecasting using empirical mode decomposition with holt-winter,” **volume** 2184, **december** 2019, **page** 050 006. DOI: 10.1063/1.5136394.
- [7] D. Maulud **and** A. Mohsin Abdulazeez, “A Review on Linear Regression Comprehensive in Machine Learning,” *Journal of Applied Science and Technology Trends*, **jourvol** 1, **pages** 140–147, **december** 2020. DOI: 10.38094/jastt1457.
- [8] S. Lima, A. M. Gonçalves **and** M. Costa, “Time series forecasting using Holt-Winters exponential smoothing: An application to economic data,” **volume** 2186, **december** 2019, **page** 090 003. DOI: 10.1063/1.5137999.
- [9] P. S. Kalekar, “Time series Forecasting using Holt-Winters Exponential Smoothing,” 2004. **url:** <https://api.semanticscholar.org/CorpusID:13942871>.
- [10] “Triple Exponential Smoothing.” **url:** <https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc435.htm>.
- [11] Vinciusspark, *Store Sales - Time Series Forecasting*, **https://www.kaggle.com/code/lorentzyeung/calendarfourier-deterministicprocess-fourier#DeterministicProcess**, Accessed: 2024-06-20.
- [12] T. Chen **and** C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” **inProceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining jourser KDD ’16**, ACM, **august** 2016. DOI: 10.1145/2939672.2939785. **url:** <http://dx.doi.org/10.1145/2939672.2939785>.
- [13] S.-H. Noh, “Analysis of Gradient Vanishing of RNNs and Performance Comparison,” *Information*, **jourvol** 12, **number** 11, **2021**, ISSN: 2078-2489. DOI: 10.3390/info12110442. **url:** <https://www.mdpi.com/2078-2489/12/11/442>.
- [14] Y. LeCun, Y. Bengio **and** G. Hinton, “Deep learning,” *Nature*, **jourvol** 521, **number** 7553, **pages** 436–444, 2015, ISSN: 1476-4687. DOI: 10.1038/nature14539. **url:** <https://doi.org/10.1038/nature14539>.
- [15] M. Ridwan, K. Sadik **and** F. Afendi, “Comparison of ARIMA and GRU Models for High-Frequency Time Series Forecasting,” *Scientific Journal of Informatics*, **jourvol** 10, **pages** 389 –400, **january** 2024. DOI: 10.15294/sji.v10i3.45965.
- [16] Y. Duan, Y. L.V. **and** F.-Y. Wang, “Travel time prediction with LSTM neural network,” **in2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)** 2016, **pages** 1053–1058. DOI: 10.1109/ITSC.2016.7795686.
- [17] X. Jin, X. Yu, X. Wang, Y. Bai, T. Su **and** J. Kong, “Prediction for Time Series with CNN and LSTM,” **inProceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019)** R. Wang, Z. Chen, W. Zhang **and** Q. Zhu, **editors**, Singapore: Springer Singapore, 2020, **pages** 631–641, ISBN: 978-981-15-0474-7.