TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

"Sử dụng mô hình thống kê và học máy để dự đoán giá cổ phiếu."

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

MEMBERS

Đỗ Thảo Quyên (20520295), Nguyễn Ngọc Thảo (20521933)
Nguyễn Tiến Nhân (20521702), Dương Ngọc Hải (20521275)
Lê Trung Hiếu (19520541)

Dự đoán giá cổ phiếu của các sàn thương mại điện tử dựa trên các mô hình học máy

1st Đỗ Thảo Quyên
IS403.N21
Trường Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh
20520295@gm.uit.edu.vn

2st Nguyễn Ngọc Thảo
IS403.N21
Trường Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh
20521933@gm.uit.edu.vn

3st Nguyễn Tiến Nhân
IS403.N21
Trường Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh
20521702@gm.uit.edu.vn

4th Dương Ngọc Hải
IS403.N21
Trường Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh
20521275@gm.uit.edu.vn

5th Lê Trung Hiếu
IS403.N21
Trường Đại học Công Nghệ Thông Tin
Thành phố Hồ Chí Minh
19520541@gm.uit.edu.vn

***Tóm tắt*—** Dự đoán giá cổ phiếu là một bài toán quan trọng trong lĩnh vực tài chính và đầu tư. Trong bài báo này, chúng em nghiên cứu việc dự đoán giá cổ phiếu của các sàn thương mại điện tử bằng các thuật toán học máy như ARIMA, ARIMAX, ARIMAR, Random Forest, RNNs, LSTM, GRU, LR, SVR, Deep FeedForward Neural Network. Chúng em thu thập dữ liệu lịch sử giá cổ phiếu từ các sàn thương mại điện tử và sử dụng các thuật toán học sâu để dự đoán giá cổ phiếu trong tương lai.

Chúng em đánh giá hiệu quả của các thuật toán bằng các độ đo như MSE, RMSE và MAPE. Kết quả thực nghiệm cho thấy rằng các thuật toán GRU, DNN, RNN, ARIMA đều cho kết quả dự đoán tốt hơn so với các mô hình truyền thống. Đặc biệt, mô hình GRU cho kết quả dự đoán tốt hơn so với các mô hình còn lại với độ chính xác cao và độ đo RMSE thấp nhất.

Kết quả của nghiên cứu này có thể được áp dụng để giúp các nhà đầu tư và các công ty đưa ra các quyết định đầu tư thông minh và hiệu quả hơn. Ngoài ra, nghiên cứu này cũng đóng góp cho việc phát triển các mô hình dự đoán giá cổ phiếu sử dụng các thuật toán học máy.

Từ khóa—mô hình học máy, dự đoán giá cổ phiếu, arima, arimax, arimar, random forest, LSTN, GRU, RNNs, linear regression, SVR, deep feedforward neural network.

I. GIỚI THIỆU

Trong những năm gần đây, thị trường chứng khoán đã trở thành một nền tảng ngày càng phổ biến cho việc đầu tư và tạo lợi nhuận. Đây là thị trường có tính chất rủi ro cao, và các nhà đầu tư cần phải đưa ra các quyết định đầu tư thông minh để giảm thiểu rủi ro và tối đa hóa lợi nhuận. Sử dụng mô hình học máy để dự đoán giá cổ phiếu có thể giúp phân tích và đánh giá các yếu tố rủi ro và tìm ra các chiến lược đầu tư tối ưu. Việc sử dụng các mô hình học máy trong đầu tư cũng đang trở thành một xu hướng mới và được nhiều nhà đầu tư quan tâm.

Để giải quyết bài toán này, nhóm đã sử dụng nhiều thuật toán khác nhau, bao gồm: ARIMA, ARIMAX, ARIMAR, Random Forest, RNNs, LSTM, GRU, LR, SVR và Deep FeedForward Neural Network. Mỗi thuật toán có ưu điểm và hạn chế riêng, tùy thuộc vào bối cảnh và mục đích sử dụng. Ví dụ, các mô hình ARIMA, ARIMAX và ARIMAR thường được sử dụng để mô hình hóa các chuỗi thời gian và dự đoán giá cổ phiếu trong tương lai. Trong khi đó, các mô hình Random Forest và Deep FeedForward Neural Network được sử dụng để xử lý các dữ liệu phi thời gian và tìm ra các mối quan hệ phức tạp giữa các yếu tố tác động đến giá cổ phiếu. Các mô hình RNNs, LSTM và GRU được sử dụng để xử lý dữ liệu chuỗi thời gian và tìm ra các mẫu và xu hướng ẩn trong các yếu tố này. Ngoài ra, các mô hình LR và SVR được sử dụng để tìm ra các mối quan hệ tuyến tính giữa các yếu tố tác động và giá cổ phiếu. Tùy thuộc vào bối cảnh và mục đích sử dụng, các nhà nghiên cứu có thể chọn một hoặc nhiều thuật toán để giải quyết bài toán dự đoán giá cổ phiếu. Tuy mỗi thuật toán có cách tiếp cận và ứng dụng riêng, nhưng chung quy lại đều nhằm mục đích dự đoán giá cổ phiếu hoặc xu hướng của giá cổ phiếu sau n-ngày.

Mục tiêu của bài toán này là dự đoán giá cổ phiếu trong tương lai, giúp các nhà đầu tư và các công ty đưa ra quyết định đầu tư thông minh và tối ưu hóa lợi nhuận. Trong bài báo này, nhóm đặt ra ba mục tiêu chính: thử nghiệm và so sánh hiệu quả của các thuật toán học máy được đề cập trên bài toán dự đoán giá cổ phiếu; đánh giá độ chính xác và ổn định của các mô hình dự đoán; và chọn ra mô hình có dự đoán với độ chính xác cao nhất để dự đoán giá cổ phiếu cho những ngày sắp tới.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Trong phần này, chúng em cung cấp một số tóm tắt liên quan đến các công trình đã được công bố về dự đoán giá cổ phiếu được sử dụng các thuật toán học máy. Chúng em bàn luận về các nghiên cứu đề xuất các phương pháp khác nhau để giải quyết cùng một vấn đề, cũng như tham gia vào các cuộc thảo luận bao quát một số vấn đề liên quan trong lĩnh vực dự đoán giá cổ phiếu.

Hiện nay, các mô hình mạng học sâu đang phát triển mạnh mẽ và giải quyết nhiều vấn đề phức tạp. Trong đó, công trình của tác giả QIAN CHEN, WENYU ZHANG [1] về bài toán dự đoán giá cổ phiếu bằng phương pháp Multi-layer Perceptron trên lịch sử giá đóng cửa trên Yahoo Finance. Kết

quả MSE, MAE thu được từ phương pháp Multi-layer Perceptron lần lượt là 0.002882 và 0.041324. Một công trình khác của tác giả Mr. Subba Rao Polamuri, Dr. Kudipudi Srinivas và Dr. A.Krishna Mohan [2] đã đề cập đến sử dụng LR, SVR, RF, Decision Tree, Extra Tree Regressor để dự báo giá cổ phiếu. Kết quả của bài báo đã chỉ ra rằng RF và Extra Tree Regressor là các mô hình tốt nhất. Bên cạnh đó, Vaishnavi Gururaj, Shriya V R và Dr. Ashwini K [3] đã sử dụng mô hình LR và SVR để dự đoán giá cổ phiếu của Công ty Coca-Cola. Bộ dữ liệu cho sẵn là dữ liệu chứng khoán trong 1 năm của Công ty Coca-Cola, từ tháng 1 2017 đến 2018 được lấy từ trang web Quandl. Kết quả cho thấy thông qua các chỉ số MAE, RMSE, MSE mô hình SVR hoạt động tốt hơn LR. Ngoài ra, Rajat Patil [4] đã sử dụng ARIMA, ARIMAX và LSTM để nghiên cứu với mục đích điều tra thị trường chứng khoán trong việc cải thiện nền kinh tế Ấn Độ bằng các dữ liệu từ năm 2000 đến năm 2020. Kết quả của nghiên cứu cho thấy mô hình ARIMAX vượt trội hơn cả. Đồng thời còn có, M K Ho1, Hazlina Darman1 and Sarah Musa1 [5] sử dụng ARIMA, Neural Network (NN) và LSTM đã được sử dụng để dự đoán dữ liệu giá đóng cửa của Bursa Malaysia từ 2/1/2020 đến 19/1/2021. Kết quả cho thấy LSTM có thể tạo ra độ chính xác hơn 90% trong việc dự đoán giá cổ phiếu trong giai đoạn đại dịch này. Tại công trình của DiasSATRIA [6] đã dự báo giá cổ phiếu của 4 ngân hàng lớn ở Indonesia từ năm 2013 đến 2022. Kết quả cho thấy mô hình ARIMA Box-Jenkins không phù hợp để dự đoán giá cổ phiếu BRI, BNI, BCA và Bank Mandiri. Ông cũng so sánh giữa 3 mô hình RNN, LSTM, và GRU, trong đó GRU là mô hình dự báo giá cổ phiếu tốt nhất dựa vào giá trị RMSE.

III. MATERIALS

A. BỘ DỮ LIỆU

Như đã đề cập ở trên, trong nghiên cứu này chúng em sẽ chọn dự báo giá chứng khoán. Với mục đích này, chúng em sẽ chọn bộ dữ liệu về giá chứng khoán của các sàn thương mại điện tử lớn trên thế giới được lấy từ Yahoo Finance, đây là một trong những nền tảng lớn và uy tín. Bộ dữ liệu bao gồm 8 thuộc tính và 1112 dòng được thu thập từ ngày 01/01/2019 đến ngày 01/06/2023.

Chúng em chọn 3 sàn thương mại điện tử lớn là Amazon, Alibaba, Ebay vì đây là 3 sàn thương mại lớn và uy tín đồng thời có sự ổn định và tiềm năng phát triển rộng lớn.

Mô tả thuộc tính dữ liệu:

TT	Tên thuộc tính	Ý nghĩa
1	Date	Cho biết ngày thực hiện giao dịch.
2	Open	Giá mở cửa của cổ phiếu trong ngày giao dịch đó, là giá cổ phiếu được giao dịch đầu tiên trong ngày.
3	High	Giá cổ phiếu cao nhất trong ngày giao dịch đó.
4	Low	Giá cổ phiếu thấp nhất trong ngày giao dịch đó.
5	Close	Giá đóng cửa của cổ phiếu trong ngày giao dịch đó, là giá cổ phiếu cuối cùng được giao dịch trong ngày
6	Volume	Số lượng cổ phiếu đã được giao dịch trong ngày đó. Nó đại diện cho tổng số cổ phiếu đã được mua và bán trong ngày
7	Dividends	Thu nhập cổ tức (nếu có) được trả cho cổ đông trong ngày đó
8	Stock Splits	Chia cổ phiếu (nếu có) trong ngày đó

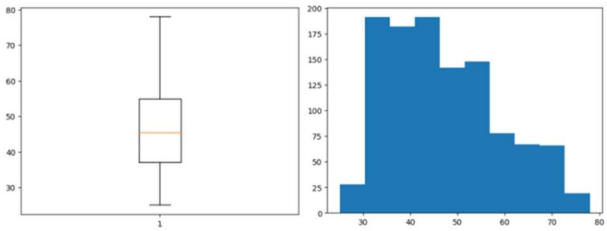
Bảng 1: Bảng mô tả dữ liệu.

Mô tả thống kê các bộ dữ liệu:

	AMZN	BABA	EBAY
Độ dài	1112	1112	1112
Min	75.01	63.15	25.03
Max	186.57	317.14	78.003
Trung bình	126.6	169.94	47.14
Trung vị	120.25	173.625	45.39
Phương sai	1083.01	3856.19	138.8
Độ lệch chuẩn	32.91	62.1	11.78
Skew	0.14	0.14	0.54
Kur	-1.6	-0.95	-0.59
Q1	94.23	109.73	37.03

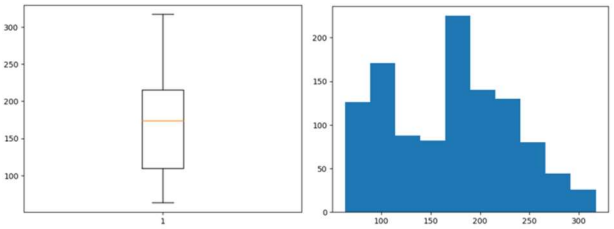
Bảng 2: Bảng thống kê mô tả dữ liệu.

- AMZN



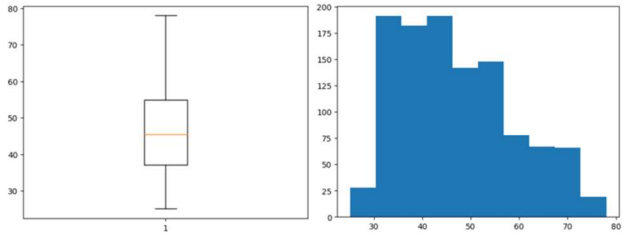
Hình 1: Biểu đồ Boxplot và Histogram của tập dữ liệu AMZN

- BABA



Hình 2: Biểu đồ Boxplot và Histogram của tập dữ liệu BABA

- EBAY



Hình 3: Biểu đồ Boxplot và Histogram của tập dữ liệu EBAY.

❖ Nhận xét:

o AMZN có giá trị trung bình tương đối cao cùng với độ lệch chuẩn tương đối thấp. Điều này cho thấy cổ phiếu này có khả năng giữ ổn định và không nhiều biến động ngẫu nhiên.

o BABA có giá trị trung bình và độ lệch chuẩn cao nhất, cho thấy sự biến động lớn trong cổ phiếu này.

o EBAY có giá trị trung bình và độ lệch chuẩn thấp nhất, cho thấy cổ phiếu này có sự ổn định và ít biến động

B. TOOLS

Trong nghiên cứu này, chúng em sử dụng Visual Studio 2022, Google Colab, Yahoo Finance để trợ giúp trong quá tình nghiên cứu.

Thông qua việc sử dụng Visual Studio 2022 và Google Colab, chúng em tiến hành xây dựng và phát triển các thuật toán và mô hình phân tích dữ liệu, như đào tạo mô hình học máy hoặc thực hiện tính toán. Yahoo Finance là nguồn cung cấp dữ liệu cho nghiên cứu của chúng em, giúp chúng em thu thập thông tin về các giá cổ phiếu cần thiết cho quá trình phân tích.

C. ĐỘ CHIA DATASET

Trong nghiên cứu này giới thiệu ba tập dữ liệu mới cho việc dự đoán giá cổ phiếu. Các tập dữ liệu này được chia thành ba tỉ lệ train : test : val khác nhau là 7 : 2 : 1, 6 : 2 : 2 và 6 : 3 : 1. Việc chia tập dữ liệu theo các tỉ lệ khác nhau giúp cho nghiên cứu có thể thử nghiệm và so sánh hiệu quả của các thuật toán khác nhau trên các tập dữ liệu khác nhau. Tập dữ liệu được thu thập từ nhiều nguồn khác nhau và được xử lý để đảm bảo tính nhất quán và độ tin cậy của dữ liệu. Kết quả cho thấy rằng các tập dữ liệu này có thể được sử dụng để xây dựng và đánh giá các mô hình dự đoán giá cổ phiếu có hiệu quả cao.

D. CÁC ĐỘ ĐO ĐÁNH GIÁ

Trong nghiên cứu này tập trung vào việc sử dụng ba độ đo quan trọng trong dự báo và đánh giá các mô hìnhMAPE (Mean Absolute Percentage Error) và RMSE (Root Mean Squared Error).

Mean Squared Error (MSE) được sử dụng để đo lường mức độ sai lệch giữa giá trị dự đoán và giá trị thực tế. MSE thường được sử dụng trong các bài toán dự đoán và học máy để đánh giá chất lượng của mô hình dự đoán. Công thức của MSE:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_1)$$

Trong đó: $\hat{y}i$ là giá trị dự đoán, y_i là giá trị thực tế và n là tổng số quan sát.

Mean Absolute Percentage Error (MAPE) là một độ đo phần trăm được sử dụng để đánh giá sự sai lệch tương đối giữa giá trị dự đoán và giá trị thực tế. MAPE cho phép đánh giá khả năng dự báo tương đối của mô hình. Công thức của MAPE:

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Trong đó: A_t là giá trị thực tế, F_t là giá trị dự đoán và n là tổng số quan sát.

Root Mean Squared Error (RMSE) là độ đo đánh giá sự khác biệt giữa các giá trị dự đoán và giá trị thực tế dưới dạng giá trị tuyệt đối, được tính bằng căn bậc hai của độ lỗi trung bình bình phương. Công thức của RMSE như sau:

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

Trong đó: $\hat{y}i$ là giá trị dự đoán, y_i là giá trị thực tế và n là tổng số quan sát.

IV. PHƯƠNG PHÁP

A. ARIMA

Mô hình ARIMA là một mô hình thống kê dùng để phân tích và dự đoán các chuỗi thời gian. Nó là một trong những mô hình dự báo phổ biến và mạnh mẽ trong lĩnh vực kinh tế, tài chính, khoa học xã hội và nhiều lĩnh vực khác.

Mô hình sử dụng đầu vào chính là những tín hiệu quá khứ của chuỗi được dự báo để dự báo nó. Các tín hiệu đó bao gồm: chuỗi tự hồi qui AR (auto regression) và chuỗi trung bình trượt MA (moving average). Hầu hết các chuỗi thời gian sẽ có xu hướng tăng hoặc giảm theo thời gian, do đó yếu tố chuỗi dừng thường không đạt được. Trong trường hợp chuỗi không dừng thì ta sẽ cần biến đổi sang chuỗi dừng bằng sai phân. Khi đó tham số đặc trưng của mô hình sẽ có thêm thành phần bậc của sai phân d và mô hình được đặc tả bởi 3 tham số ARIMA(p, d, q) [7].

- AR: Đây là thành phần tự hồi qui bao gồm tợp hợp các độ trễ của biến hiện tại. Độ trễ bậc p chính là giá trị lùi về quá khứ p bước thời gian của chuỗi.

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

- MA: Quá trình trung bình trượt được hiểu là quá trình dịch chuyển hoặc thay đổi giá trị trung bình của chuỗi theo thời gian

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

- I: Sai phân chỉ sự khác nhau giữa giá trị hiện tại và giá trị trước đó. Tức là hiệu giữa giá trị hiện tại và d giá trị trước đó.

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

  Sai phân lần 2 I(2):

  $\Delta(\Delta y_t) = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$

  Sai phân lần d được kí hiệu là I(d)

  ⇨ ARIMA(p,d,q) =AR(p) + I(d)+ MA(q)

B. ARIMAX

Mô hình ARIMAX là một dạng mở rộng của model ARIMA. Mô hình sẽ có thêm một vài biến độc lập khác và cũng được xem như một mô hình hồi qui động. Về bản chất ARIMAX tương ứng với một mô hình hồi qui đa biến nhưng chiếm lợi thế trong dự báo nhờ xem xét đến yếu tố tự tương quan được biểu diễn trong phần dư của mô hình. Nhờ đó cải thiện độ chính xác.

$$\Delta P_t = c + \beta X + \phi_1 * \Delta P_{t-1} + \theta_1 * \varepsilon_{t-1} + \varepsilon_t$$

βX là viết tắt của hệ số β của sự kết hợp tuyến tính với biến ngoại sinh X [8].

C. ARIMAR (Hybrid ARIMA and Random Walk)

Mô hình Hybrid ARIMA and Random Walk là một mô hình kết hợp giữa mô hình ARIMA và mô hình Random Walk trong việc dự đoán chuỗi thời gian. Mô hình này được sử dụng để giải quyết các vấn đề trong việc dự đoán chuỗi thời gian, bao gồm khó khăn trong việc mô hình hóa các thành phần trend và seasonality của chuỗi thời gian, và sự biến động ngẫu nhiên.

Mô hình ARIMA là một mô hình dự đoán chuỗi thời gian dựa trên cả thành phần autoregressive và moving average của chuỗi. Nó giúp mô hình hóa và dự đoán sự biến động của chuỗi thời gian, bao gồm các yếu tố trend, seasonality và sự biến động ngẫu nhiên. Tuy nhiên, mô hình ARIMA có thể không phù hợp khi dữ liệu của chuỗi thời gian không có tính ổn định và có sự thay đổi lớn trong thời gian.

Mô hình Random Walk là một mô hình dự đoán chuỗi thời gian dựa trên sự dự đoán giá trị hiện tại phụ thuộc vào giá trị của chuỗi thời gian tại thời điểm trước đó và một thành phần ngẫu nhiên. Mô hình này không mô hình hóa trend và seasonality của chuỗi thời gian và chỉ tập trung vào sự biến động ngẫu nhiên của chuỗi thời gian.Ta cũng có thể nhận thấy rằng Random Walk chính là mô hình ARIMA(0,1,0). Công thức của mô hình

Random Walk:

$$X(t) = X(t-1) + \varepsilon(t)$$

Trong đó:

- X(t) là giá trị tại thời điểm t.
- X(t-1) là giá trị tại thời điểm trước đó (t-1).
- ε(t) là thành phần ngẫu nhiên, có thể tuân theo phân phối chuẩn (Gaussian) với trung bình 0 và độ lệch chuẩn σ.

Bằng sự kết hợp cả hai mô hình trên ta sẽ tận dụng được lợi thế của cả hai. Ý tưởng chính của mô hình này là sử dụng mô hình **ARIMA** để mô hình hóa và dự đoán xu hướng chính của chuỗi thời gian, trong khi mô hình **Random Walk** được sử dụng để mô phỏng các yếu tố ngẫu nhiên và các biến động nhỏ không thể được mô hình **ARIMA** dự báo chính xác.

Cách sử dụng mô hình:

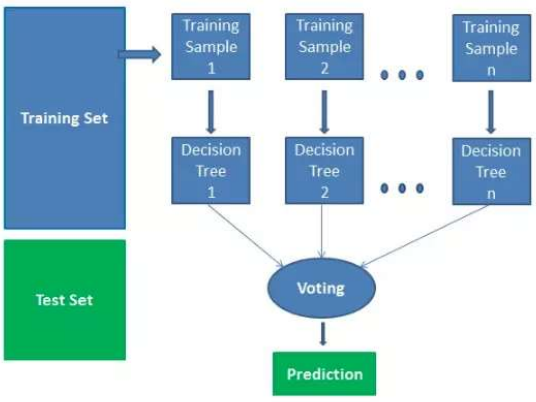
- Trước tiên, cần kiểm tra tính dừng của tập dữ liệu.
- Thực hiện xây dựng mô hình ARIMA, tìm các tham số: **p,d,q**
- Xây dựng mô hình Random Walk bằng cách xây dựng mô hình ARIMA(0,1,0).
- Thực hiện dự đoán bằng mô hình ARIMA và RandomWalk trên tập dữ liệu.
- Kết hợp dự đoán của cả hai mô hình theo trọng số (a)**ARIMA** và (1-a) **Random Walk** để có thể có được giá trị dự đoán của mô hình **Hybrid ARIMA and Random Walk**.

D. RANDOM FOREST

Random Forests là thuật toán học có giám sát (supervised learning). Nó có thể được sử dụng cho cả phân lớp và hồi quy. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất. Một khu rừng bao gồm cây cối. Người ta nói rằng càng có nhiều cây thì rừng càng mạnh. Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Nó cũng cung cấp một chỉ báo khá tốt về tầm quan trọng của tính năng. Random forests có nhiều ứng dụng, chẳng hạn như công cụ đề xuất, phân loại hình ảnh và lựa chọn tính năng. Nó có thể được sử dụng để phân loại các ứng viên cho vay trung thành, xác định hoạt động gian lận và dự đoán các bệnh. Nó nằm ở cơ sở của thuật toán Boruta, chọn các tính năng quan trọng trong tập dữ liệu.

Random Forest hoạt động theo 4 bước:

- Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.
- Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi cây quyết định.
- Bỏ phiếu cho mỗi kết quả dự đoán
- Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng

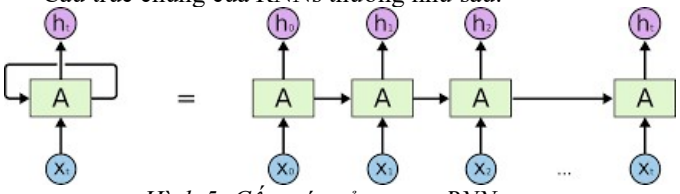


Hình 4: Cách hoạt động của RF.

E. RECURRENT NEURAL NETWORKs - RNNs

Mạng nơ-ron tái phát (RNNs) là mạng thần kinh cho phép sử dụng các đầu ra trước đó như đầu vào trong quá trình có các trạng thái ẩn.

Cấu trúc chung của RNNs thường như sau:



Hình 5: Cấu trúc của mạng RNNs.

Trong mạng RNNs, thông tin chuyển qua một vòng lặp đến lớp ẩn giữa. Tầng đầu vào (input layer) x_t nhận đầu vào vào mạng nơ-ron và xử lý nó, sau đó chuyển tiếp thông tin đến tầng ẩn (hidden layer). Tầng ở giữa 'A' có thể bao gồm nhiều tầng ẩn, mỗi tầng có các hàm kích hoạt và trọng số và bias riêng của nó.

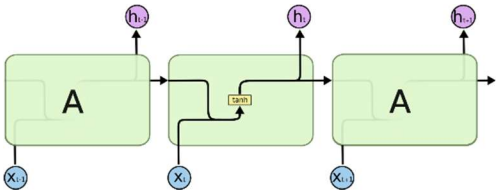
Mạng nơ-ron tái phát sẽ chuẩn hóa các hàm kích hoạt và trọng số và bias khác nhau để mỗi tầng ẩn có cùng các tham số. Sau đó, thay vì tạo nhiều tầng ẩn, nó sẽ tạo một tầng và lặp lại nó nhiều lần tùy theo yêu cầu.

F. LONG SHORT TERM MEMORY

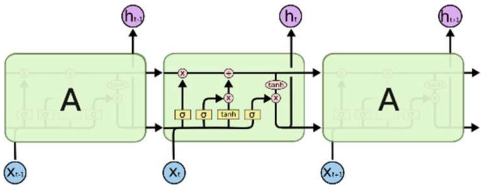
Mạng bộ nhớ dài-ngắn (Long Short-Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mỗi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh.



LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



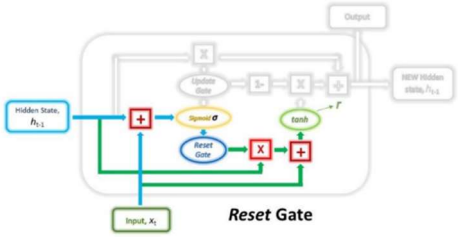
G. GATED RECURRENT UNIT – GRU

Mô hình GRU (Gated Recurrent Unit) là một kiến trúc mạng nơ-ron hồi quy (RNN) được sử dụng để xử lý và dự đoán dữ liệu chuỗi theo thời gian. Nó được giới thiệu bởi Cho et al. vào năm 2014 như là một biến thể của mô hình LSTM (Long Short-Term Memory).

GRU được thiết kế để giải quyết vấn đề biến mất gradient trong mạng RNN truyền thống và đồng thời giảm số lượng tham số so với LSTM. Mô hình GRU cung cấp một cơ chế cổng (gate) linh hoạt để kiểm soát và điều chỉnh thông tin trong quá trình chuỗi.

Cách hoạt động của mô hình GRU được thể hiện qua cơ chế cổng (gate) linh hoạt. GRU sử dụng hai cổng chính: cổng cập nhật (update gate) và cổng đặt lại (reset gate). Các cổng này cho phép mô hình điều chỉnh và kiểm soát thông tin trong quá trình chuỗi.

- **Cổng đặt lại** (Reset gate)



Hình 6: Cổng đặt lại.

Công thức tính toán của cổng này:

$$gate_{reset} = \sigma\,(W_{input\ reset} \cdot x_t + W_{hidden\ reset} \cdot h_{t-1})$$

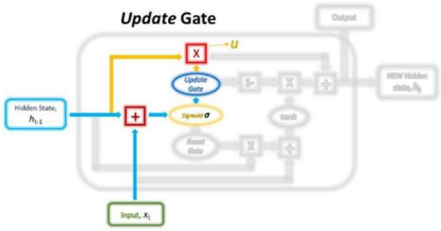
Trong đó $gate_{reset}$ là giá trị của cổng đặt lại; σ là hàm sigmoid; $W_{input\ reset}$ là trọng số kết nối giữa đầu vào (x_t) và cổng đặt lại; $W_{hidden\ reset}$ là trọng số kết nối giữa trạng thái ẩn trước đó (h_{t-1}) và cổng đặt lại.

Sau khi tính được giá trị của cổng đặt lại, chúng ta sẽ tính toán giá trị được cập nhật cho trạng thái ẩn hiện tại (h_t) thông qua công thức sau:

$$r = \tanh(gate_{reset} \odot (W_{h1} \cdot h_{t-1}) + W_{x1} \cdot x_t)$$

Trong đó ⊙ là phép nhân từng phần tử (element-wise multiplication); W_{h1} là trọng số của trạng thái ẩn trước đó(h_{t-1}); W_{x1} là trọng số của đầu vào(x_t).

- **Cổng cập nhật** (Update gate)



Hình 7: Cổng cập nhật.

Cũng giống như **cổng Đặt lại**, **cổng Cập nhật** được tính bằng cách sử dụng trạng thái ẩn trước đó và dữ liệu đầu vào hiện tại:

$$gate_{update} = \sigma\ (W_{input\ update} \cdot x_t + W_{hidden\ update} \cdot h_{t-1})$$

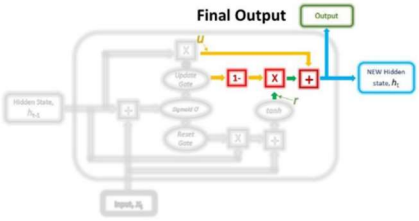
Trong đó: σ là hàm sigmoid; $W_{input\ update}$ và $W_{hidden\ update}$ lần lượt là trọng số của đầu vào input và trạng thái ẩn trước đó.

Sau đó, vector Cập nhật (Update) sẽ trải qua phép nhân từng phần với trạng thái ẩn trước đó để thu được **u** trong phương trình dưới đây, nó sẽ được sử dụng để tính toán đầu ra cuối cùng sau này:

$$u = gate_{update} \odot h_{t-1}$$

- **Kết hợp các đầu ra:**

Ở bước cuối cùng, chúng ta sẽ sử dụng lại cổng Cập nhật (Update gate) và thu được trạng thái ẩn được cập nhật.



Hình 8: Tính toán đầu ra cuối cùng.

Lần này, chúng ta sẽ lấy phép nghịch đảo từng phần của cùng một vector Cập nhật (1 - Update gate) và thực hiện phép nhân từng phần với đầu ra từ cổng Đặt lại (Reset gate), r. Mục đích của phép toán này là để cổng Cập nhật xác định phần nào của thông tin mới sẽ được lưu trữ trong trạng thái ẩn.

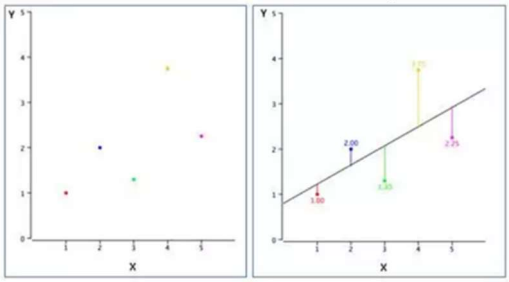
Cuối cùng, kết quả từ các phép toán trên sẽ được tổng hợp với đầu ra từ cổng Cập nhật ở bước trước, **u**. Điều này sẽ cho chúng ta trạng thái ẩn mới và được cập nhật.

$$h_t = r \odot (1 - gate_{update}) + u.$$

H. LINEAR REGRESSION – LR

Hồi quy tuyến tính là thuật toán tìm ra phương trình tuyến tính dựa trên tập dữ liệu quan hệ giữa X (dữ liệu đầu vào) và Y (dữ liệu đầu ra). X là biến giải thích và Y là biến phụ thuộc. Trước khi thử tạo ra mô hình quan hệ, chúng ta nên xác định liệu giữa các mối quan hệ này có liên quan với nhau hay không. Điều này chỉ ra rằng, không nhất thiết phải có sự tương tác giữa các biến, nhưng cần phải có sự liên quan. Nếu không có mối quan hệ nào giữa các biến được đưa vào mô hình, thì mô hình hồi quy tuyến tính sẽ không giúp ích được trong trường hợp này.

Trong khi sử dụng hồi quy tuyến tính, mục tiêu của chúng ta là để làm sao một đường thẳng có thể tạo được sự phân bố gần nhất với hầu hết các điểm. Do đó làm giảm khoảng cách (sai số) của các điểm dữ liệu cho đến đường đó.



Hình 9: Minh họa thuật toán LR.

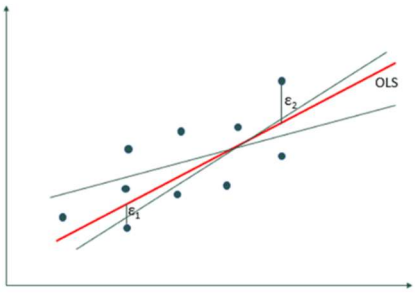
Ví dụ, ở các điểm ở hình trên (trái) biểu diễn các điểm dữ liệu khác nhau và đường thẳng (bên phải) đại diện cho một đường gần đúng có thể giải thích mối quan hệ giữa các trục x & y. Thông qua, hồi quy tuyến tính chúng ta cố gắng tìm ra một đường như vậy. Ví dụ, nếu chúng ta có một biến phụ thuộc Y và một biến độc lập X - mối quan hệ giữa X và Y có thể được biểu diễn dưới dạng phương trình sau:

$$\mathbf{Y = \beta 0 + \beta 1 * X}$$

Trong đó:

- Y: Biến phụ thuộc
- B0: Hằng số
- B1: Hệ số mối quan hệ giữa Y và X
- X: biến độc lập

Một trong các phương pháp ước lượng hồi quy tuyến tính phổ biến là bình phương nhỏ nhất OLS (Ordinary Least Squares). Nguyên tắc của phương pháp hồi quy OLS là làm cho biến thiên phần dư này trong phép hồi quy là nhỏ nhất. Khi biểu diễn trên mặt phẳng Oxy, đường hồi quy OLS là một đường thẳng đi qua đám đông các điểm dữ liệu mà ở đó, khoảng cách từ các điểm dữ liệu (trị tuyệt đối của ε) đến đường hồi quy là ngắn nhất.



Hình 10: Minh họa phương pháp OLS.

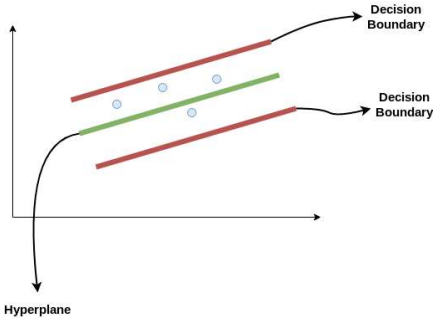
Từ đồ thị scatter biểu diễn mối quan hệ giữa các biến độc lập và biến phụ thuộc, các điểm dữ liệu sẽ nằm phân tán nhưng có xu hướng chung tạo thành dạng một đường thẳng. Chúng ta có thể có rất nhiều đường đường thẳng hồi quy đi qua đám đông các điểm dữ liệu này chứ không phải chỉ một đường duy nhất, vấn đề là ta phải chọn ra đường thẳng nào

mô tả sát nhất xu hướng dữ liệu. Bình phương nhỏ nhất OLS sẽ tìm ra đường thẳng đó dựa trên nguyên tắc cực tiểu hóa khoảng cách từ các điểm dữ liệu đến đường thẳng. Trong hình ở trên đường màu đỏ là đường hồi quy OLS.

I. SUPPORT VECTOR REGRESSION – SVR

Support Vector Regression (SVR) là một thuật toán học máy trong lĩnh vực hồi quy, được phát triển dựa trên Support Vector Machines (SVM). SVR được sử dụng để dự đoán giá trị đầu ra liên tục trong các bài toán hồi quy.

Giống như SVM trong bài toán phân loại, SVR cũng sử dụng các vector hỗ trợ (support vectors) để xác định siêu mặt phẳng tốt nhất phân chia giữa các điểm dữ liệu. Tuy nhiên, mục tiêu của SVR không phải là tìm siêu mặt phẳng chia hai lớp một cách tối ưu, mà là tìm siêu mặt phẳng sao cho khoảng cách giữa các điểm dữ liệu được gán nhãn và siêu mặt phẳng là nhỏ nhất.



Hình 10: Minh họa thuật toán SVR.

SVR có thể xử lý các mối quan hệ phi tuyến tính giữa đầu vào biến và biến mục tiêu bằng cách sử dụng hàm nhân để ánh xạ dữ liệu sang không gian có chiều cao hơn. Điều này làm cho nó trở thành một công cụ mạnh mẽ cho các nhiệm vụ hồi quy, nơi có thể có phức tạp mối quan hệ giữa các biến đầu vào và biến mục tiêu.

Vì SVR chỉ có khả năng nội suy (interpolate) giữa các dữ liệu hiện có trong tập dữ liệu huấn luyện, nên nếu chúng ta đánh giá ước lượng bên ngoài miền giá trị mà chúng ta đã đánh giá, thì kết quả sẽ phụ thuộc trực tiếp vào sự lựa chọn của hàm kernel được sử dụng cho SVR và các thuật toán tối ưu hóa.

Kernel được sử dụng trong nghiên cứu này là **Linear**, các kernel thông thường được sử dụng trong SVR bao gồm:

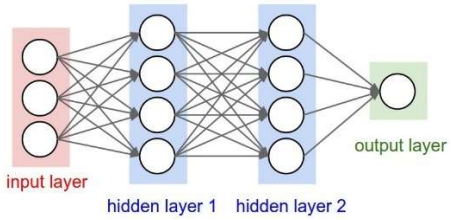
Kernel	Phương trình
Linear	$k(x,y) = x^T y$
RBF	$k(x,y) = \exp\left(\frac{\|\|x-y\|\|^2}{2\sigma^2}\right)$
Sigmoid	$k(x,y) = \tanh(ax^T y + b)$
Polynomial	$k(x,y) = (x * y + 1)^d$

J. DEEP FEEDFROWARD NEURAL NETWORK - DNN

Mạng neuron truyền thẳng (Deep Feedforward Neural Network) là một mạng nơ-ron nhân tạo sử dụng để huấn luyện và dự đoán giá trị đầu ra phụ thuộc vào các giá trị đầu vào. Ngoài ra, dữ liệu chỉ được truyền đi từ lớp đầu vào đến lớp đầu ra thông qua các lớp ẩn giữa chúng.

Một mạng Deep Feed Forward Neural Network gồm 3 tầng chính (Hình 1)

- Tầng vào (Input Layer): Tầng này nằm bên trái cùng của mạng, thể hiện cho các đầu vào của mạng.
- Tầng ẩn (Hidden Layer): Tầng này nằm giữa tầng vào và tầng ra nó thể hiện cho quá trình suy luận logic của mạng.
- Tầng ra (Output Layer): Là tầng bên phải cùng và nó thể hiện cho những đầu ra của mạng.

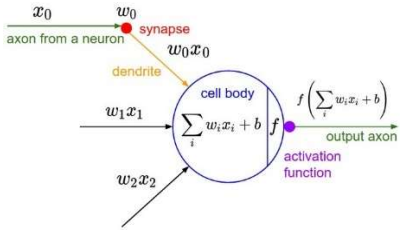


Hình 11: Kiến trúc mạng Deep FeedForward Neural Network.

Trong kiến trúc mạng DNN, node (neurons) trong mạng neural là một đơn vị cơ bản xử lý thông tin và các node trong mỗi lớp (layer) được kết nối chặc chẽ với các node trong lớp khác để truyền thông tin. Mỗi kết nối có trọng số, biểu thị tương quan giữa các node. Dữ liệu được truyền qua mạng, và giá trị đầu ra của mỗi node được tính dựa trên giá trị đầu vào và trọng số tương ứng, với công thức:

$$Output\ of\ neuron = Y = f\left(\sum_i w_i x_i + b\right)$$

Trong đó: w_i là trọng số, x_i là giá trị đầu vào và b là giá trị bias.



Hình 12: Minh họa một neural trong kiến trúc DNN.

V. KẾT QUẢ THỰC NGHIỆM

A. KẾT QUẢ TRÊN BỘ DỮ LIỆU

1. AMZN

1.1. MSE, RMSE, MAPE với độ chia 7:2:1

Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	29.377	5.420	0.0367
RNN	36.671	6.0557	0.045
LSTM	48.066	6.933	5.055
RF	129.440	11.377	9.128
LR	7115.144	84.351	0.695

SVR	68.706	8.288	0.054
ARIMA	1182.614	34.39	0.266
ARIMAX	313.91	17.72	0.128
AIRIMAR	1182.80	34.391	0.266
GRU	16.32	4.04	0.028

Bảng 3: Kết quả trên AMZN bộ chia 7:2:1.

1.2. MSE, RMSE, MAPE với độ chia 6:3:1

Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	74.540	8.634	0.0612
RNN	25.378	5.0376	0.032
LSTM	77.669	8.813	5.940
RF	173.770	13.182	9.551
LR	6555.903	80.968	0.583
SVR	130.778	11.435	0.074
ARIMA	1444.286	38.003	0.259
ARIMAX	292.742	17.11	0.105
AIRIMAR	1444.236	38.003	0.259
GRU	20.42	4.5	0.03

Bảng 4: Kết quả trên AMZN bộ chia 6:3:1.

1.3. MSE, RMSE, MAPE với độ chia 6:2:2

Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	98.190	9.909	0.064
RNN	46.352	6.808	0.042
LSTM	40.304	6.348	4.721
RF	185.879	13.633	8.879
LR	3329.544	57.702	0.361
SVR	111.441	10.556	0.066
ARIMA	738.523	27.176	0.151
ARIMAX	279.18	16.709	0.086
ARIMAR	738.5	27.175	0.151
GRU	27.22	5.22	0.032

Bảng 5: Kết quả trên AMZN bộ chia 6:2:2.

Với bộ dữ liệu của AMZN được chia tỷ lệ train:test:val (7:3:1, 6:3:1, 6:2:2) nhóm chúng em thu được kết quả mô hình tốt nhất là GRU.

2. BABA

2.1. MSE, RMSE, MAPE với độ chia 7:2:1

Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	43.513	6.596	0.065
RNN	18.472	4.298	0.040
LSTM	35.201	5.933	5.812
RF	1153.224	33.959	38.821
LR	14043.859	118.506	1.306
SVR	111.790	10.573	0.11
ARIMA	1325.25	36.404	0.387
ARIMAX	204.082	14.286	0.137
ARIMAR	1315.893	36.275	0.385
GRU	13.57	3.68	0.034

Bảng 6: Kết quả trên BABA bộ chia 7:2:1.

2.2. MSE, RMSE, MAPE với độ chia 6:3:1

Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	74.656	8.640	0.0786
RNN	37.759	6.145	0.049
LSTM	110.625	10.517	9.896
RF	216.737	14.722	10.820
LR	29485.194	171.712	1.695
SVR	209.014	14.457	0.141
ARIMA	4448.105	66.694	0.644
ARIMAX	614.23	24.784	0.222
ARIMAR	1315.89	36.275	0.385
GRU	26.02	5.10	0.042

Bảng 7: Kết quả trên BABA bộ chia 6:3:1.

2.3. MSE, RMSE, MAPE với độ chia 6:2:2

Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	3.1397	1.7719	0.031
RNN	144.029	12.001	0.0959
LSTM	138.675	11.776	9.654
RF	207.361	14.400	9.444
LR	22553.812	150.179	1.288
SVR	193.734	13.918	0.117
ARIMA	2901.458	53.865	0.441
ARIMAX	529.384	23.008	0.169
ARIMAR	2926.64	54.098	0.443
GRU	49.42	7.03	0.057

Bảng 8: Kết quả trên BABA bộ chia 6:2:2.

Với bộ dữ liệu của AMZN được chia tỷ lệ train:test:val là: 7:3:1, 6:3:1, 6:2:2 thì chúng em có mô hình tốt nhất là GRU và DNN.

3. EBAY

3.1. MSE, RMSE, MAPE với độ chia 7:2:1

Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	1.823	1.350	0.0254
RNN	1.856	1.362	0.026
LSTM	1.755	1.325	2.476
RF	138.508	11.768	9.658
LR	277.811	16.667	0.272
SVR	7.985	2.825	0.054
ARIMA	193.17	13.899	0.294
ARIMAX	70.926	8.422	0.17
ARIMAR	193.169	13.898	0.294
GRU	1.416	1.19	0.023

Bảng 9: Kết quả trên EBAY bộ chia 7:2:1.

3.2. MSE, RMSE, MAPE với độ chia 6:3:1

Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	5.707	2.389	0.044
RNN	1.904	1.380	0.023
LSTM	2.098	1.448	2.495
RF	195.547	13.983	10.238

LR	591.149	24.313	0.458
SVR	6.594	2.567	0.045
ARIMA	1091.249	33.034	0.622
ARIMAX	114.038	10.679	0.201
ARIMAR	1091.25	33.034	0.622
GRU	1.749	1.323	0.022

Bảng 10: Kết quả trên EBAY bộ chia 6:3:1.

3.3. MSE, RMSE, MAPE với độ chia 6:2:2

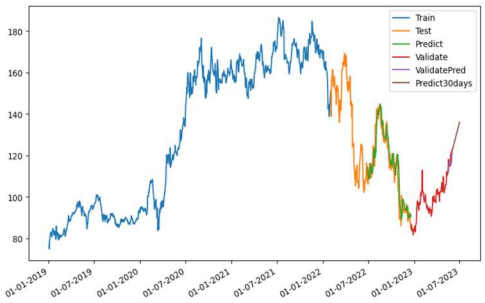
Thuật toán	Test		
	MSE	RMSE	MAPE
DNN	3.139	1.772	0.031
RNN	4.389	2.095	0.0329
LSTM	3.250	1.802	2.968
RF	177.492	13.322	8.642
LR	277.811	16.667	0.272
SVR	4.816	2.194	0.036
ARIMA	563.918	23.747	0.382
ARIMAX	57.329	7.572	0.123
ARIMAR	563.918	23.747	0.382
GRU	1.789	1.337	0.02

Bảng 11: Kết quả trên BABA bộ chia 6:2:2.

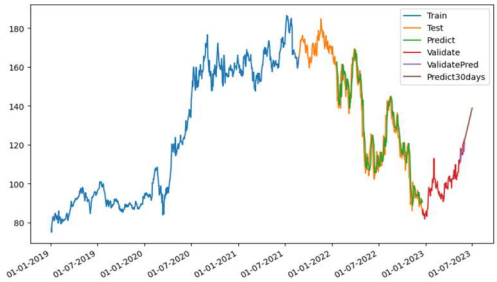
Với bộ dữ liệu của AMZN được chia tỷ lệ train:test:val là: 7:3:1, 6:3:1, 6:2:2 thì chúng em có mô hình tốt nhất là GRU.

B. DỰ ĐOÁN GIÁ CỔ PHIẾU 30 NGÀY TỚI

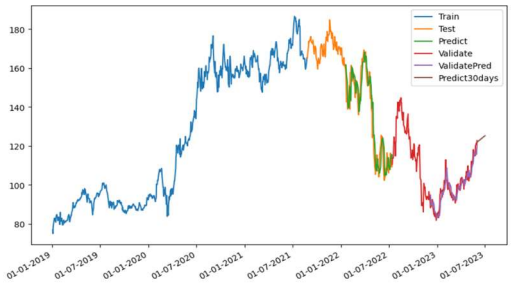
1. AMZN



Hình 13: Mô hình dự đoán GRU theo độ chia 7:2:1.



Hình 12: Mô hình dự đoán GRU theo độ chia 6:3:1.



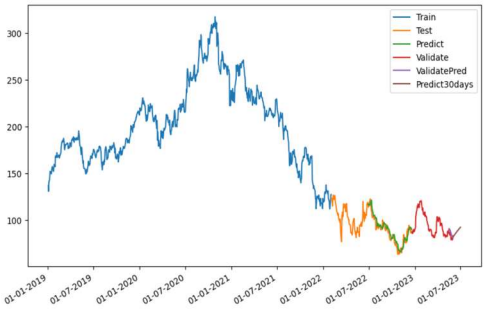
Hình 13: Mô hình dự đoán GRU theo độ chia 6:2:2.

Bảng dưới là kết quả dự đoán giá cổ phiếu và giá thực tế trên bộ dữ liệu AMZN.

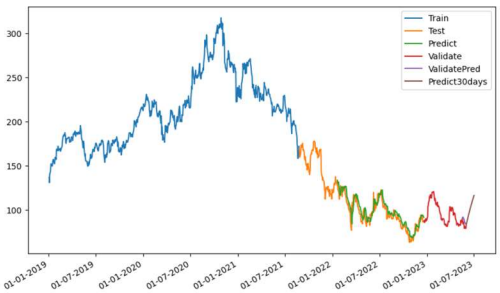
Ngày	Giá thực tế	Giá dự đoán		
		GRU (7:3:1)	GRU (6:3:1)	GRU (6:2:2)
2/6/2023	124.25	122.68	122.35	122.23
5/6/2023	125.3	124.37	123.44	123.22
6/6/2023	126.61	124.92	123.79	123.52
7/6/2023	121.23	125.48	124.14	123.81
8/6/2023	124.25	126.034	124.49	124.11
9/6/2023	123.43	126.59	124.84	124.41
12/6/2023	126.57	128.29	125.90	125.31
13/6/2023	126.66	128.86	126.25	125.61
14/6/2023	126.42	129.43	126.61	125.91
15/6/2023	127.11	130.00	126.96	126.21
16/6/2023	125.49	130.58	127.32	126.52

Bảng 12: Bảng giá thực tế với giá dự đoán cổ phiếu AMZN.

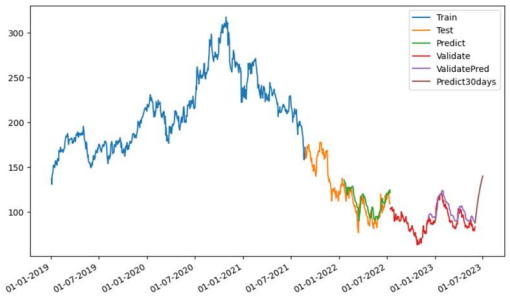
2. BABA



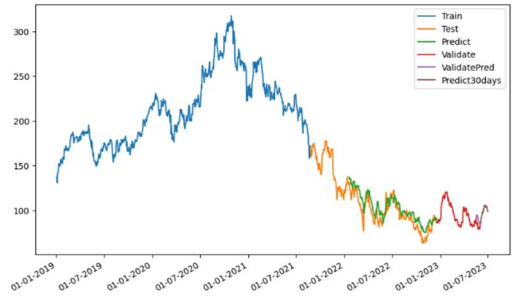
Hình 14: Mô hình dự đoán GRU theo độ chia 7:2:1.



Hình 15: Mô hình dự đoán GRU theo độ chia 6:3:1.



Hình 16: Mô hình dự đoán GRU theo độ chia 6:2:2.



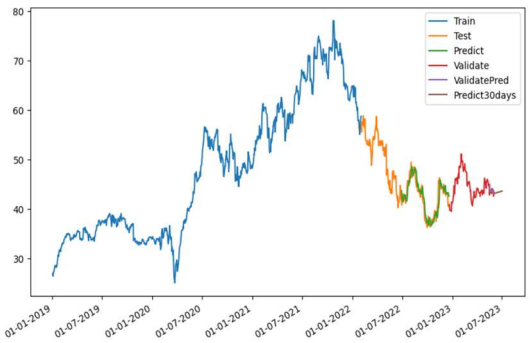
Hình 17: Mô hình dự đoán DNN theo độ chia 6:2:2.

Bảng dưới là kết quả dự đoán giá cổ phiếu và giá thực tế trên bộ dữ liệu BABA.

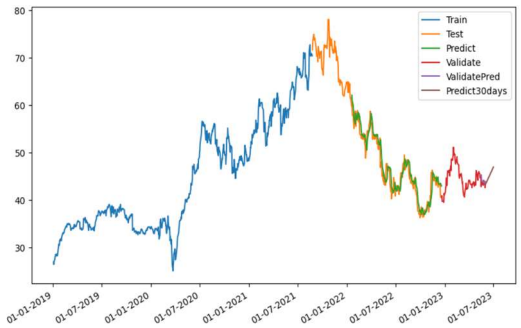
Ngày	Giá thực tế	Giá dự đoán		
		GRU (7:3:1)	GRU (6:3:1)	DNN (6:2:2)
2/6/2023	84.27	82.82	84.45	88.64
5/6/2023	84.40	85.32	89.01	92.70
6/6/2023	86.70	86.12	90.52	93.88
7/6/2023	85.28	86.91	92.00	94.40
8/6/2023	86.14	87.69	93.47	95.83
9/6/2023	85.50	88.46	94.92	97.29
12/6/2023	85.86	90.70	99.16	96.69
13/6/2023	87.51	91.43	100.54	97.38
14/6/2023	89.36	92.15	101.90	99.37
15/6/2023	92.20	92.86	103.24	99.61
16/6/2023	92.10	93.56	104.57	101.38

Bảng 13: Bảng giá thực tế với giá dự đoán cổ phiếu BABA.

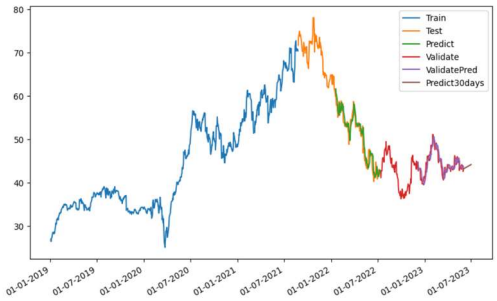
3. EBAY



Hình 18: Mô hình dự đoán GRU theo độ chia 7:2:1.



Hình 19: Mô hình dự đoán GRU theo độ chia 7:2:1.



Hình 20: Mô hình dự đoán GRU theo độ chia 7:2:1.

Bảng dưới là kết quả dự đoán giá cổ phiếu và giá thực tế trên bộ dữ liệu EBAY.

Ngày	Giá thực tế	Giá dự đoán		
		GRU (7:3:1)	GRU (6:3:1)	GRU (6:2:2)
2/6/2023	44.39	43.46	43.56	43.20
5/6/2023	45.29	44.19	44.40	43.38
6/6/2023	45.18	44.45	44.70	43.44
7/6/2023	45.06	44.72	44.99	43.51
8/6/2023	45.61	44.98	45.30	43.58
9/6/2023	45.81	45.25	45.60	43.65
12/6/2023	45.12	46.08	46.53	43.86
13/6/2023	45.52	46.36	46.84	43.94
14/6/2023	45.07	46.64	47.16	44.00
15/6/2023	45.62	46.92	47.48	44.08
16/6/2023	45.06	47.21	47.80	44.15

Bảng 14: Bảng giá thực tế với giá dự đoán cổ phiếu EBAY.

VI. TỔNG KẾT VÀ ĐÁNH GIÁ

Nghiên cứu của chúng em đã chỉ ra rằng GRU (Gated Recurrent Unit) và DNN (Deep Neural Network) là hai mô hình phù hợp nhất để dự đoán giá trong tương lai của các cổ phiếu AMZN, BABA và EBAY trong số mười mô hình được thử nghiệm, bao gồm DNN, RNN, LSTM, RF, LR, SVR, ARIMA, ARIMAX, ARIMAXR và GRU.

Hai mô hình GRU và DNN được biết đến với khả năng mô hình hóa dữ liệu chuỗi thời gian phức tạp và khả năng học được các mẫu phi tuyến tính trong dữ liệu. GRU là một dạng RNN được thiết kế để giảm hiện tượng biến mất đạo hàm trong quá trình huấn luyện và giúp giải quyết vấn đề vanishing gradient. Trong khi đó, DNN là một mô hình sâu với nhiều lớp ẩn, có khả năng học các mức độ biểu diễn phức tạp hơn so với mô hình tuyến tính như LR hoặc SVM.

Các mô hình còn lại như RNN, LSTM, RF, LR, SVR, ARIMA, ARIMAX, ARIMAXR cũng mang ý nghĩa quan trọng trong nghiên cứu này, vì chúng đại diện cho các phương pháp phân tích dữ liệu thống kê, học máy cổ điển và phân tích chuỗi thời gian. Mặc dù không được xác định là phù hợp nhất trong việc dự đoán giá cổ phiếu trong nghiên cứu này, nhưng các phương pháp này có thể có ứng dụng trong các ngữ cảnh khác nhau và có thể mang lại giá trị trong việc phân tích dữ liệu tài chính.

Tuy nhiên, cần lưu ý rằng kết quả của nghiên cứu này chỉ áp dụng cho các cổ phiếu AMZN, BABA và EBAY và không thể tổng quát cho tất cả các cổ phiếu khác. Nên tiếp tục nghiên cứu và kiểm tra hiệu suất của các mô hình trên các tập dữ liệu khác nhau để xác định tính tổng quát và ứng dụng của chúng trong việc dự đoán giá cổ phiếu.

A. ACKNOWLEDMENT

Chúng em xin bày tỏ lòng biết ơn chân thành đến giảng viên, PGS.TS. Nguyễn Đình Thuân, anh Nguyễn Minh Nhựt và chị Nguyễn Thị Việt Hương, vì sự hướng dẫn và hỗ trợ nhiệt tình trong suốt quá trình thực hiện dự án nghiên cứu của chúng em.

Chúng em cũng muốn gửi lời cảm ơn đến Khoa Hệ Thống Thông Tin, Trường Công Nghệ Thông Tin đã cung cấp cho chúng em tài nguyên và cơ sở vật chất cần thiết để hoàn thành công việc của mình.

Cuối cùng, chúng em muốn cảm ơn tất cả các thành viên đã rất nỗ lực dành thời gian và hợp tác để làm cho nghiên cứu này trở thành hiện thực.

B. REFERENCES

[1] QIAN CHEN, WENYU ZHANG, "Forecasting Stock Prices Using a Hybrid Deep Learning Model Integrating Attention Mechanism, Multi-Layer Perceptron, and Bidirectional Long-Short Term Memory Neural Network," *IEEE Access,* 2020.

[2] Subba Rao Polamuri, K. Srinivas, A. Krishna Mohan, "Stock Market Prices Prediction using Random Forest and Extra Tree Regression," *International Journal of Recent Technology and Engineering (IJRTE),* 2019.

[3] S. V. R. a. D. A. K. Vaishnavi Gururaj, "Stock Market Prediction using Linear Regression and Support Vector," *International Journal of Applied Engineering Research ISSN,* 2019.

[4] R. Patil, "Time Series Analysis and Stock Price Forecasting using Machine Learning Techniques," *National College of Ireland,* 2021.

[5] M K Ho, Hazlina Darman and Sarah Musa, "Stock Price Prediction Using ARIMA, Neural Network and LSTM Models," *Journal of Physics: Conference Series.*

[6] D. SATRIA, "PREDICTING BANKING STOCK PRICES USING RNN, LSTM, AND GRU APPROACH," *Applied Computer Science,* 2023.

[7] [Online]. Available: https://phamdinhkhanh.github.io/.

[8] [Online]. Available: https://365datascience.com/tutorials/python-tutorials/arima/?fbclid=IwAR3WEfpDHCpKBe_RTbb_KjjtxdD3KaYb8r-ChzhHX7lisaWDNcAIpjweq54.

[9] "alkaline-ml.com," [Online]. Available: http://alkaline-ml.com/pmdarima/.

[10] "alkaline-ml.com," [Online]. Available: http://alkaline-ml.com/pmdarima/.

[11] DiasSATRIA, "PREDICTING BANKING STOCK PRICES USING RNN, LSTM, AND GRU APPROACH," *Applied Computer Science,* 2023.