Sử dụng kỹ thuật phân tích chuỗi thời gian vào  
 bài toán dự đoán giá cổ phiếu

Trần Thị Mỹ Nhung   
IS304.N21.HTTT  
Đại học Công nghệ   
Thông tin  
[20520267@gm.uit.edu.vn](mailto:20520267@gm.uit.edu.vn)

Nguyễn Anh Kiệt   
IS304.N21.HTTT  
Đại học Công nghệ  
 Thông tin  
[20521498@gm.uit.edu.vn](mailto:20521498@gm.uit.edu.vn)

Nguyễn Thành Phát   
IS304.N21.HTTT  
Đại học Công nghệ  
 Thông tin[20520270@gm.uit.edu.vn](mailto:20520270@gm.uit.edu.vn)

Nguyễn Hoài Linh  
IS304.N21.HTTT  
Đại học Công nghệ   
Thông tin20521534@gm.uit.edu.vn

Nguyễn Đạt  
IS304.N21.HTTT  
Đại học Công nghệ  
 Thông tin20520434@gm.uit.edu.vn

*Tóm tắt* — Bài báo sử dụng kỹ thuật phân tích chuỗi thời gian trong bài toán dự đoán giá cổ phiếu. Dự đoán giá cổ phiếu là một tác vụ khó khăn do tính phức tạp và biến động của thị trường tài chính. Đó là lí do cho sự quan tâm ngày càng tăng về việc sử dụng các phương pháp máy học và mạng nơ-ron sâu để cải thiện độ chính xác của dự đoán giá cổ phiếu. Trong bài báo này, chúng tôi sử dụng các mô hình: Linear Regression, Gated Recurrent Unit (GRU), Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing (ETS), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Gradient Boosted Tree (GBT), Hidden Markov Model (HMM), Extreme Gradient Boosting (XGBoost), Deep Feedforward Neural Network (DFNN), Temporal Convolutional Attention Networks (TCANs) để dự đoán giá cổ phiếu trên ba bộ dữ liệu Catalent, Intel Corporation và Nutrien theo hai tỉ lệ của train:validate:test là 7:1:2 và 6:2:2. Sau đó thực hiện so sánh hiệu suất của các mô hình dựa trên ba độ đo: MAPE, RMSE, MLSE. Cuối cùng, chúng tôi sử dụng hai mô hình tốt nhất để dự đoán giá đóng cửa cổ phiếu 30 ngày tiếp theo đối với hai mô hình đạt hiệu suất tốt nhất. Kết quả cho thấy mô hình LSTM và HMM có hiệu suất tốt hơn so với các mô hình còn lại. Các kết quả nghiên cứu này có thể cung cấp thông tin hữu ích và hướng dẫn cho các nhà đầu tư và chuyên gia tài chính trong việc đưa ra các quyết định dựa trên dự đoán giá cổ phiếu.

Từ khóa – Cổ phiếu, Linear Regression, GRU, ARIMA, ETS, LSTM, RNN, GBT, HMM, XGBoost, DFNN, TCANs

# **GIỚI THIỆU**

Cổ phiếu được coi là hình thức đầu tư trọng điểm trong ngành tài chính, đại diện cho quyền sở hữu một phần trong tổ chức phát hành. Theo Investopedia [1], công ty Đông Ấn Hà Lan phát hành cổ phiếu đầu tiên vào năm 1602 tại Sở Giao dịch Chứng khoán Amsterdam, đồng thời cũng là công ty đầu tiên phát hành cổ phiếu và trái phiếu. Điều này đã được coi là một sự tiến bộ quan trọng trong lĩnh vực tài chính và đã mở ra thời kỳ phát triển của thị trường cổ phiếu.

Định giá cổ phiếu là quy trình xác định giá trị thị trường thực sự của cổ phiếu tại một thời điểm nhất định, nhằm hiểu rõ tiềm năng của cổ phiếu để đưa ra quyết định đầu tư phù hợp. Đối với doanh nghiệp, việc định giá cổ phiếu được coi là một trong những bước tiên quyết khi công ty cổ phần dự định phát hành cổ phiếu, huy động vốn và tăng cường ảnh hưởng của mình trên thị trường. Từ góc độ của nhà đầu tư, việc định giá cổ phiếu giúp họ xác định cổ phiếu nào đáng để đầu tư và có tiềm năng mang lại lợi nhuận tối đa.

Một phương pháp tiếp cận sơ bộ trong việc định giá cổ phiếu là đánh giá giá trị cổ phiếu. Nếu giá cổ phiếu hiện tại thấp hơn giá trị đã định giá, nhà đầu tư có thể xem xét mua cổ phiếu. Ngược lại, nếu giá cổ phiếu vượt quá giá trị đã định giá và nhà đầu tư hiện đang sở hữu cổ phiếu, họ có thể bán cổ phiếu để thu về lợi nhuận.

Thực tế cho thấy có nhiều thuật toán và kỹ thuật hỗ trợ việc dự báo giá cổ phiếu của ba doanh nghiệp bao gồm: Catalent, Intel Corporation, Nutrien. Trong nghiên cứu này, chúng tôi sẽ áp dụng 11 mô hình: GRU, ARIMA, LSTM, Linear Regression, ES, HMM, GBT, DFNN, XGBoost, DNN, TCA, RNN để tiến hành đánh giá hiệu suất các mô hình, sau đó sử dụng hai mô hình tốt nhất thực hiện dự đoán giá đóng cửa của cổ phiếu trong 30 ngày tiếp theo.

# **CÁC NGHIÊN CỨU LIÊN QUAN**

Nhóm tác giả Vaishnavi Gururaj, Shriya V R và Dr. Ashwini K [2] đã nghiên cứu về thị trường chứng khoán bằng mô hình Linear Regression. Dataset mà nhóm tác giả sử dụng là một năm dữ liệu cổ phiếu của Công ty Coca-Cola, từ 01/2017-2018. Các kết quả về độ đo bao gồm: 3.22 (RMSE), 2.53 (MAE), 10.37 (MSE) và 0.73 (R-Squared).

Nghiên cứu "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique"[3], tác giả Raisa Abedin Disha và S. Waheed đã thử nghiệm và đánh giá hiệu suất của các mô hình học máy như GRU và GBT trong việc phân loại trong hệ thống phát hiện xâm nhập. Các mô hình này đã được huấn luyện và kiểm tra trên hai tập dữ liệu UNSW-NB 15 và Network TON\_IoT. Để tăng cường hiệu suất của các mô hình, tác giả đã sử dụng kỹ thuật lựa chọn đặc trưng gọi là Gini Impurity-based Weighted Random Forest (GIWRF). Kỹ thuật này giúp tác giả chọn ra một tập hợp tối ưu các đặc trưng từ dữ liệu. Kết quả thực nghiệm cho thấy mô hình Cây quyết định (DT) hoạt động tốt hơn so với các mô hình khác trong thí nghiệm này khi sử dụng kỹ thuật lựa chọn đặc trưng GIWRF.

Box và Jenkins đã giới thiệu mô hình ARIMA vào năm 1970. Đây còn được gọi là phương pháp Box-Jenkins, bao gồm một tập hợp các hoạt động để xác định, ước lượng và chẩn đoán các mô hình ARIMA với dữ liệu chuỗi thời gian [4]. Các mô hình ARIMA đã chứng minh khả năng tạo ra dự đoán ngắn hạn hiệu quả. ARIMA liên tục vượt trội so với các mô hình phức tạp khác trong dự đoán ngắn hạn [5].

Hai tác giả F. Sidqi và I. D. Sumitra [6] đã nghiên cứu và áp dụng hai phương pháp Single Exponential Smoothing (SES) và Double Exponential Smoothing (DES) để dự báo doanh số bán hàng sản phẩm. Kết quả cho thấy, MAPE của SES là 20% và MAPE của DES khoảng 24%. Phương pháp Single Exponential Smoothing có sai số thấp hơn.

Ba tác giả Murtaza Roondiwala, Harshal Patel và Shraddha Varma đã áp dụng mô hình LSTM (Long Short-Term memory) trong việc dự đoán giá trị của cổ phiếu của NIFTY 50 [7]. Họ đã sử dụng 500 epochs để train và kết quả đạt được là vô cùng tốt. Mô hình cho ra được các kết quả RMSE trên tập test rơi vào

khoảng 0.00859. Sai số vô cùng thấp cho thấy giá trị dự đoán cực kì tốt.

Tác giả Yongqiong Zhu [8] đã sử dụng mô hình RNN để dự đoán giá cố phiếu của Apple với dữ liệu huấn luyện là giá cổ phiểu của Apple (AAPL) trong 10 năm (từ 9/8/2009 đến 12/8/2020 với tập train chiếm 65% và tập test chiếm 35% còn lại. Tác giả đã xây dựng mô hình mạng RNN hai lớp với lớp thứ nhất có 50 nút đơn vị và lớp thứ hai chứa 100 nút đơn vị. Tác giả sử dụng 50 epochs, tối ưu hóa bằng Adam và dùng hàm mất mát là MSE đã cho ra được một kết quả rất tốt. Mô hình cho kết quả độ chính xác của dự đoán lên đến hơn 95% và giá trị mất mát là 0.1%.

Ba tác giả Raihan Tanvir, Md Tanvir Rouf Shawon và Md. Golam Rabiul Alam đã dự đoán giá cổ phiếu của DSE bằng cách sử dụng mô hình HMM. Sau quá trình huấn luyện, nhóm tác giả sử dụng phương pháp tối đa hóa hậu nghiệm (MAP) [9] để dự đoán giá cổ phiếu của Dhaka Stock Exchange (DSE) cho ngày tiếp theo. Tiến hành đánh giá độ chính xác của mô hình bằng các độ đo MAE, RMSE và MAPE cho ra các kết quả lần lượt là 2.5064, 3.4003 và 1.0265. Điều đó cho thấy mô hình cho ra kết quả đáng tin cậy trong việc dự đoán giá cổ phiếu của DSE.

Ba tác giả Kyung Keun Yun, Sang Won Yoon và Daehan Won đã dự đoán giá cổ phiểu KOSPI của bằng cách sử dụng mô hình XgBoost [10]. Tiến hành đánh giá độ chính xác của mô hình đạt được độ chính xác là 93.75 trên tập dữ liệu về cổ phiếu của KOSPI. Mô hình đạt được kết quả vô cùng tốt.

Tác giả Orimoloye và đồng nghiệp [11] trong nghiên cứu so sánh hiệu quả của mô hình Deep Feedforward Neural Network và Shallow architectures để dự đoán chỉ số giá cổ phiếu dựa trên trên ba mươi bốn chỉ số tài chính từ 32 quốc gia trong vòng sáu năm. Tác giả sử dụng Kiểm định Tukey's Standardized Range (HSD) về độ chính xác của các mô hình với kết quả 0.607 DNN (ReLU), 0.582 DNN (Tanh) và nhận thấy rằng hàm kích hoạt ReLU hoạt động tốt hơn hàm kích hoạt Tanh trên tất cả các thị trường và khoảng thời gian khi sử dụng DNN để dự đoán chỉ số giá cổ phiếu.

Y. Lin và các cộng sự [12] đã nghiên cứu và áp dụng mô hình Temporal Convolutional Attention Neural Networks cho dự báo năng lượng mặt trời. Kết quả cho thấy TCANs cho kết quả tốt trên cả dự báo điểm (0.5) và dự báo xác suất (0.9). Các kết quả 0.5/0.9 bao gồm: 0.062/0.031 (Sanyo), 0.068/0.035 (Hanery) và 0.209/0.081 (Solar).

# **TÀI NGUYÊN**

Bài báo sử dụng bộ dữ liệu được lấy từ dữ liệu chứng khoán của 3 công ty CTLT[[1]](#footnote-2)(Dữ liệu của công ty Catalent được lấy về từ trang web finance.yahoo.com có 1119 dòng dữ liệu), INTC[[2]](#footnote-3) (Dữ liệu của công ty Intel Corporation được lấy về từ trang web finance.yahoo.com có 1119 dòng dữ liệu) và NTR[[3]](#footnote-4)(Dữ liệu của công ty Nutrien được lấy về từ trang web finance.yahoo.com có 1119 dòng dữ liệu). Với dữ liệu được thu thập từ ngày 1/1/2019 đến ngày 12/6/2023 và được tải về vào ngày 12/6/2023. Ở cả 3 bộ dữ liệu đều có các cột thuộc tính liên quan đến cổ phiếu.

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Date | Ngày diễn ra giao dịch |
| Open | Giá cổ phiếu đầu tiên được giao dịch |
| High | Giá cao nhất của cổ phiếu được giao dịch |
| Low | Giá thấp nhất của cổ phiếu được giao dịch |
| Close | Giá đóng cửa của cổ phiếu |
| Adj Close | Giá đóng của điều chỉnh |
| Volume | Khối lượng giao dịch |

Bảng . Mô tả thuộc tính trong ba bộ dữ liệu

Thông tin dữ liệu thu thập được mô tả chi tiết trong bảng 2. Bảng 2 thể hiện đánh giá tổng quan trên thuộc tính Close – Thuộc tính được lựa chọn để thực hiện dự đoán giá đóng cửa cổ phiếu.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CTLT** | **INTC** | **NTR** |
| Count | 1119 | 1119 | 1119 |
| Mean | 80.051 | 48.129 | 60.496 |
| Std | 30.003 | 10.864 | 18.428 |
| Min | 29.84 | 24.9 | 25.1 |
| Max | 142.35 | 68.47 | 114.5 |
| 25% | 51.33 | 44.265 | 48.495 |
| 50% | 78.55 | 50.31 | 55.35 |
| 75% | 106.025 | 55.98 | 73.38 |
| Skewness | 0.1411 | -0.647. | 0.566 |
| Kurtosis | -1.321 | -0.533 | -0.282 |
| Histogram |  |  |  |
| Box Plot |  |  |  |
| Nhận xét | - Giá trị Skewness gần bằng 0 cho thấy phân phối dữ liệu có hình dạng gần đối xứng.  - Giá trị Kurtosis âm phân phối có độ nhọn thấp hơn bình thường  - Biểu đồ Histogram cho thấy tần suất xuất hiện của giá cổ phiếu có giá trị từ 41000 đến 51000 là nhiều nhất.  - Biểu đồ Boxplot cho thấy dữ liệu cổ phiếu CTLT không có giá trị ngoại lai và độ phân tán lớn. | - Giá trị Skewness âm cho thấy phân phối dữ liệu lệch mạnh về bên trái so với trung vị.  - Giá trị Kurtosis âm tuy nhiên không có sự khác biệt lớn nên có độ nhọn thấp hơn bình thường.  - Biểu đồ Histogram cho thấy tần suất xuất hiện của giá cổ phiếu có giá trị từ 46000 đến 51000 là nhiều nhất.  - Biểu đồ Boxplot cho thấy dữ liệu cổ phiếu INTC có giá trị ngoại lai nằm ngoài giá trị min và độ phân tán ít. | - Giá trị Skewness dương cho thấy phân phói dữ liệu lệch nhẹ về bên phải.  - Giá trị kurtosis này nhỏ hơn 0 cho thấy phân phối có độ nhọn thấp hơn bình thường.  - Biểu đồ Histogram cho thấy tần suất xuất hiện của giá cổ phiếu có giá trị từ 43000 đến 53000 là nhiều nhất.  - Biểu đồ Boxplot cho thấy dữ liệu cổ phiếu NTR có giá trị ngoại lai nằm ngoài giá trị max và độ phân tán ít. |

Bảng 2. Mô tả chi tiết về dữ liệu thu thập

Để quá trình thực nghiệm đạt hiệu quả tốt nhất, chúng tôi sử dụng một số công cụ bao gồm: *Google Colab -* Colaborator hay còn gọi là Google Colab, là một sản phẩm từ Google Research, nó cho phép chạy các dòng lệnh code python qua trình duyệt. Có thể sử dụng tài nguyên máy tính để chạy nhanh hơn như CPU tốc độ cao, GPUs, TPUs, …*Visual Studio Code -* Là một trình soạn thảo mã nguồn mở gọn nhẹ nhưng có khả năng vận hành mạnh mẽ trên 3 nền tảng là Windows, Linux và macOS được phát triển bởi Microsoft. Bên cạnh đó, chúng tôi sử dụng một số thư viện trong quá trình thực nghiệm như*:* numpy, pandas, matplotlib, scikit-learn, keras, minmaxscaler,…

# **PHƯƠNG PHÁP**

## Linear Regression

Trong lĩnh vực thống kê, Linear Regression là một phương pháp dùng để xác định mối quan hệ giữa một biến phụ thuộc có giá trị số và một hoặc nhiều biến độc lập.

Khi chỉ có một biến độc lập, chúng ta gọi là hồi quy tuyến tính đơn giản (Simple Linear Regression). Trong trường hợp có nhiều biến độc lập, ta gọi là hồi quy tuyến tính đa biến (Multiple Linear Legression) [13].

Simple Linear Regression được mô tả qua công thức:

Trong đó:

: biến phụ thuộc (dependent variable) cần dự đoán.

: biến độc lập (independent variable) được sử dụng để dự đoán giá trị của .

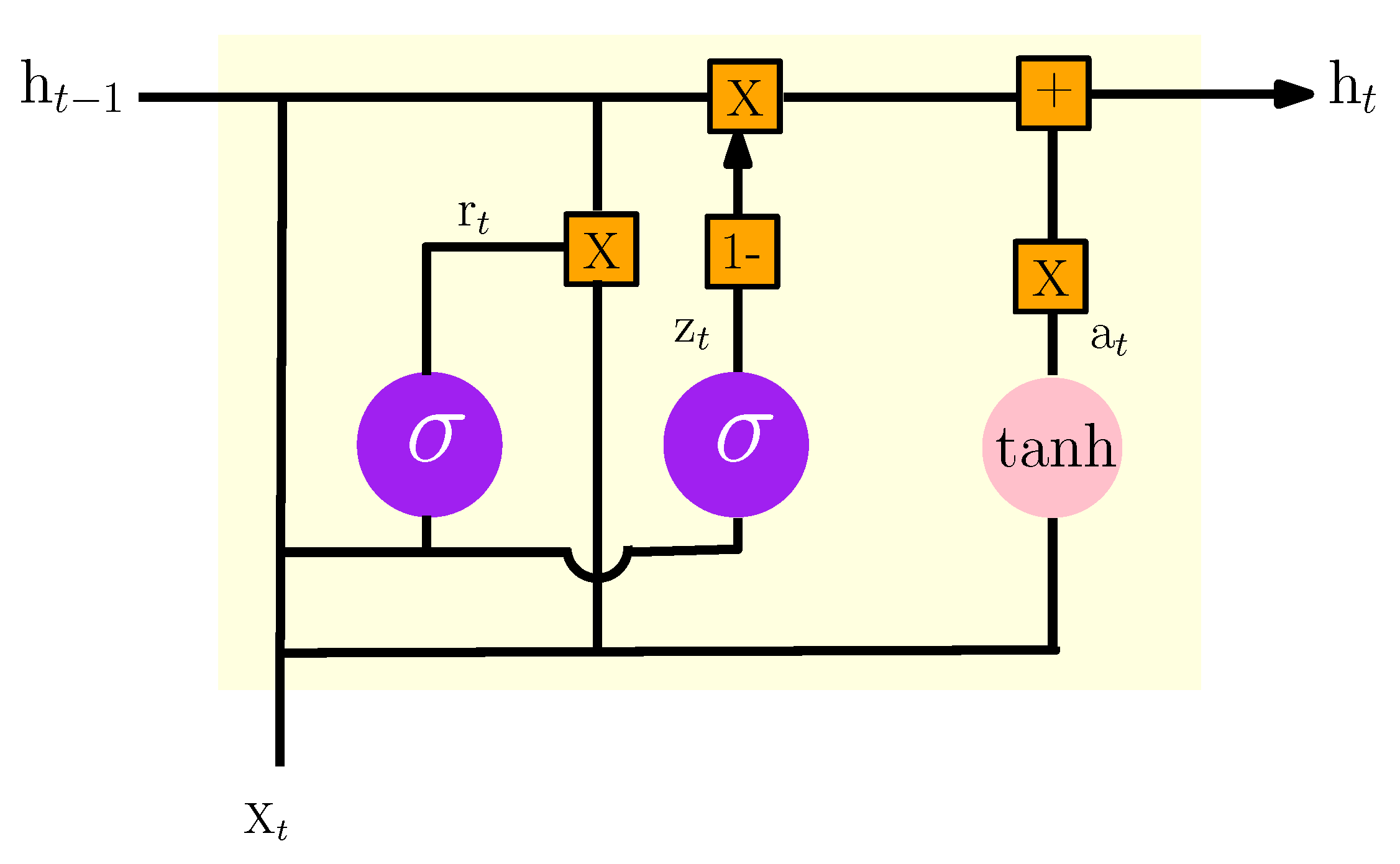
: hệ số góc (intercept) của đường hồi quy, đại diện cho giá trị dự đoán của khi .

: hệ số hồi quy (regression coefficient), đại diện cho mức độ thay đổi của dựa trên mỗi đơn vị thay đổi của .

: lỗi ngẫu nhiên (random error), biểu thị sự không thể tránh khỏi của mô hình trong việc mô phỏng dữ liệu thực tế.

## GRU

Theo Mahjoub và đồng nghiệp [14], mô hình Gated Recurrent Unit (GRU) là một biến thể được tối ưu hóa của Recurrent Neural Network (RNN), dựa trên mô hình Long Short-Term Memory (LSTM). GRU gộp cổng đầu vào và cổng quên của LSTM thành một cổng cập nhật duy nhất, giảm độ phức tạp của mô hình.



Hình . Kiến trúc GRU

Trong GRU, có hai cổng chính. Bao gồm cổng cập nhật và cổng đặt lại được tính toán dựa trên dữ liệu đầu vào hiện tại và trạng thái ẩn từ bước thời gian trước đó . Công thức được tính như sau:

1. Cổng cập nhật:
2. Cổng đặt lại:

Sau đó, tính toán trạng thái ẩn ứng viên bằng hàm tanh với cổng đặt lại, dữ liệu đầu vào hiện tại và trạng thái ẩn từ bước thời gian trước đó:

1. Trạng thái ẩn ứng viên:

Cuối cùng, trạng thái ẩn hiện tại được tính toán bằng cách kết hợp trạng thái ẩn ứng viên với trạng thái ẩn từ bước thời gian trước đó, điều chỉnh bởi cổng cập nhật:

1. Trạng thái ẩn hiện tại:

Để tối ưu hoá tham số, nhóm sử dụng kỹ thuật Random search và các phương pháp ngăn ngừa overfitting để tìm các ngẫu nhiên để tìm các tham số để huấn luyện trên tập train và đánh giá độ chính xác dựa trên tập validate dựa vào phương pháp đánh giá MAPE để chọn tham số tốt nhất trong mạng lưới tham số được thiết lập.

## ARIMA

Trong một mô hình ARIMA, giá trị tương lai của một biến được giả định là một hàm tuyến tính của một số quan sát quá khứ cộng với các lỗi ngẫu nhiên. Hàm tuyến tính này dựa trên ba thành phần tham số: tự hồi quy (AR), tích hợp sai phân (I) và trung bình trượt (MA) [15]. Mô hình ARIMA có thể được ký hiệu là ARIMA(p, d, q), trong đó p là số lượng thành phần tự hồi quy, d là số lượng sự khác biệt không mùa và q là số lượng lỗi dự báo trễ trong phương trình dự đoán [16]. Giá trị tương lai của một biến trong ARIMA được biểu thị như sau:

Trong đó:

: Giá trị thực tế.

: Sai số ngẫu nhiên tại thời điểm t.

và : Là các hệ số.

## ETS

Theo Zhi-Peng và đồng nghiệp(2008) [17] Exponential Smoothing (ETS) là một phương pháp dự báo trong phân tích chuỗi thời gian, nơi các chuỗi thời gian quan sát được gán trọng số không đều.

Simple Exponential Smoothing (SES) gán các trọng số giảm dần theo thời gian. Cụ thể, phương pháp này tính toán dự báo bằng cách lấy trọng số lớn nhất cho quan sát gần nhất và trọng số nhỏ hơn cho các quan sát xa hơn.

Công thức cập nhật dự đoán của phương pháp SES như sau:

trong đó, là giá trị dự đoán tại thời điểm t, là giá trị quan sát ở thời điểm t, là giá trị dự đoán tại thời điểm t – 1, là một tham số trơn (smoothing parameter) nằm trong khoảng từ 0 đến 1 và nó quyết định mức độ ảnh hưởng của mỗi quan sát.

Nhóm sử dụng phương pháp tìm kiếm lưới (grid search) để tìm giá trị alpha tối ưu trong mô hình Simple Exponential Smoothing dựa trên mean squared error trên dữ liệu validate. Phương pháp này giúp tìm ra giá trị alpha làm giảm Mean Squared Error (MSE) và cải thiện độ chính xác của mô hình trên dữ liệu validate.

## LSTM

Theo Alex Graves và đồng nghiệp (2005) [18] LSTM là viết tắt của "Long Short-Term Memory", một loại mạng nơ-ron học sâu hay còn được biến đến là một loại đặc biệt của mạng Recurrent Neural Network (RNN).

Một LSTM layer bao gồm một tập hợp các khối nhớ được kết nối theo chu kỳ. Mỗi khối chứa một hoặc nhiều ô nhớ được kết nối theo chu kỳ thông qua ba cổng nhân tích - cổng đầu vào, cổng đầu ra và cổng quên. Chúng cung cấp các phép ghi, đọc và đặt lại liên tục cho ô nhớ.

Sự ra đời của LSTM đã giúp hạn chế phần nào vấn đề phụ thuộc xa mà RNN mắc phải nhờ khả năng học các phụ thuộc dài hạn.

Cấu trúc của LSTM:

A picture containing screenshot, diagram

Description automatically generated

Hình 2. Kiến trúc LSTM

Công thức tính toán các cổng trong LSTM:

*Forget gate:*

*Input gate:*

*Cell gate:*

*Ouput gate:*

*Cell state:*

## RNN

Recurrent Neural Network (RNN) là một mô hình mạng neural hồi quy gồm 3 thành phần chính là Input layer, Hidden layer và Output layer [19].

Mô hình tổng hợp và lan truyền thông tin theo trình tự xâu chuỗi. Mô hình được gọi là hồi quy vì thực hiện tính toán ở thời điểm hiện tại sẽ phụ thuộc vào các kết quả tính toán ở những thời điểm trước đó. Do đó, RNN phải nhớ thông tin đã được đưa vào để tính toán trước đó để phục vụ cho các bước tính toán sau này.

Sau nhiều bước biến đổi, các thông tin ở những bước đầu tiên bị biến đổi nhiều lần dẫn đến bị triệt tiêu. Do đó, RNN chỉ có nhớ được một vài bước trước đó để mô hình được tối ưu nhất.

Kiến trúc của RNN [STT]:

A picture containing screenshot, diagram, text, line

Description automatically generated

Hình 3. Kiến trúc RNN

RNN được biểu diễn bằng công thức sau:

và

Trong đó:

: giá trị đầu vào tại thời điểm t

: giá trị đầu ra tại thời điểm t

: giá trị kích hoạt

: các ma trận trọng số

: vector độ lệch

: các hàm kích hoạt

## GBT

Gradient Boosted Tree (GBT) là một thuật toán machine learning, được sử dụng trong các tác vụ hồi quy và phân loại. Gradient boosting được diễn giải như một thuật toán tối ưu hóa dựa trên một hàm chi phí phù hợp, dựa trên quan sát của Leo Breiman[20]. Cụ thể hơn, các thuật toán Gradient Boosting đã được phát triển dưới góc nhìn hạ gradient trong không gian hàm số - nghĩa là các thuật toán tối ưu hóa một hàm chi phí qua không gian hàm số bằng cách lựa chọn lặp đi lặp lại một hàm (giả thuyết yếu) chỉ vào hướng gradient âm.

Công thức chính của GBT:

Trong đó:

​ là mô hình mạnh tại bước thứ i.

là mô hình yếu tại bước thứ i.

γ là giá trị thu nhỏ. Kiểm soát tốc độ học của mô hình mạnh và giúp giới hạn quá khớp

Để tối ưu thuật toán Grandient Boosted Trees nhóm sử dụng phương pháp RandomSearch cùng với các phương pháp ngăn ngừa overfitting để tìm kiếm ngẫu nhiên tham số trên mạmg được thiết lập sẵn, điều này giúp tiết kiệm thời gian, chi phí đồng thời ngăn ngừa overfitting nhưng vẫn tìm kiếm được tham số tốt nhất.

## HMM

Theo Raihan Tanvir và đồng nghiệp (2023) [9], Hidden Markov Model (HMM) là một mô hình thống kê được xử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, phân tích chuỗi DNA, xử lý ảnh. Trong đó, mô hình cho phép dự đoán trạng thái ẩn dựa trên các quan sát được từ thực tế.

Mô hình gồm 3 tham số chính chính (): ma trận sác xuất chuyển đổi trạng thái (*A*), ma trận sác xuất quan sát (*B*) (cung cấp xác suất bj của quan sát Ot) và phân phối xác suất ban đầu của các trạng thái ẩn ().

Giá trị quan sát của mô hình là một vector ba chiều biểu diễn thông tin cổ phiếu hàng ngày:

*Ot =*

*:= (fracChange, fracHigh, fracLow)*

Sử dụng phương pháp tối đa hóa hậu nghiệm (MAP) để kiểm tra mô hình sau khi huấn luyện. Với việc sử dụng dữ liệu từ *d* ngày trước đó và giá mở cửa của ngày thứ *d + 1* để tính toán được *fracChange* cho ngày đó, giá trị MAP của ngày thứ *d + 1* được tính như sau:

*= arg*

Sau quá trình tối đa hóa hậu nghiệm, giá đóng cửa được dự đoán bằng công thức:

*Close = Open x (1 + fracChange)*

Khi áp dụng mô hình HMM vào việc dự đoán giá đóng cửa, nhóm đã tối ưu hóa qua việc chia nhỏ *fracChange* thành các giá trị nằm trong khoảng *fracChange* cao nhất và thấp nhất của *d* ngày với số bước chia nhất định (*number of steps*). Sau đó, tiến hành tìm giá trị *number of steps* tối ưu nhất dựa trên độ đo MAE (Mean Absolute Error) trên tập dữ liệu validate. Bằng cách này, nhóm đã tìm ra được giá trị *number of steps* có thể làm giảm giá trị MAE và cải thiện được độ chính xác của mô hình trên tập dữ liệu validate.

## XGBoost

XGBoost viết tắt của "eXtreme Gradient Boosting". Là một thư viện tăng cường gradient phân tán được tối ưu hóa, được thiết kế để đạt hiệu suất cao, linh hoạt và dễ di động.

XgBoost thực hiện các thuật toán học máy dựa trên khung Gradient Boosting. XGBoost cung cấp một cách tăng cường cây song song (còn được gọi là GBDT, GBM) giải quyết nhiều vấn đề khoa học dữ liệu một cách nhanh chóng và chính xác [21].

A picture containing text, line, diagram, font

Description automatically generated

Hình 4. Mô hình hóa thuật toán XgBoost

Một số công thức được sử dụng trong mô hình XgBoost [22]:

l(x1, x2) – cost function

– Thứ tụ tại cây đang thực thi

= 0 – Khởi tạo giá trị dự đoạn ban đầu

Quá trình tính toán lặp đi lặp lại cho đến khi có được giá trị cuối cùng:

Hàm dự đoán tổng hợp của tất cả các cây trong tập cây (ensemble): h(x) =

## DFNN

Theo Oyedele và đồng nghiệp [23], Deep Feedforward Neural Network (DFNN) là mô hình học sâu điển hình để học các biểu diễn dữ liệu phức tạp và trừu tượng theo cấp bậc. Quá trình học này truyền tải dữ liệu qua nhiều lớp biến đổi.

Kiến trúc của DFNN gồm 3 tầng: Input layer, Hidden layer, Output layer. Trong đó mỗi lớp có nhiều đơn vị xử lý kết nối với nhau. Trong DFNN, mỗi lớp sử dụng một phép biến đổi phi tuyến trên input của nó để tạo ra output. Hình IV.1 thể hiện kiến trúc DFNN, là một đồ thị hướng không chu trình, có nghĩa là luồng thông tin chỉ diễn ra theo hướng tiến về phía trước, đơn giản hóa quá trình huấn luyện backpropagation.

A picture containing diagram, line

Description automatically generated

Hình 5. Kiến trúc mô hình DFNN

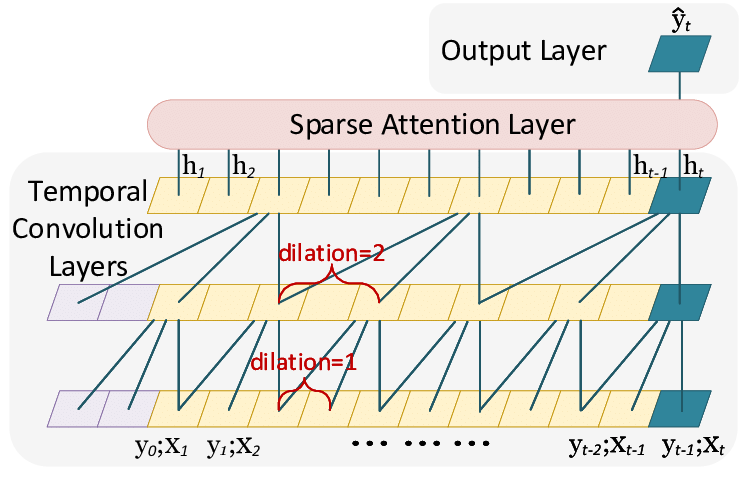
Trong mỗi lớp DFNN, đầu ra của mỗi node được tính toán bằng hàm kích hoạt , với trọng số . Số node trong hidden layer thứ i là . Các node được tính toán như sau:

Trong đó là chỉ số đại diện cho trong lớp ; là chỉ số đại diện cho một node đầu vào từ lớp ; n là tổng số node đầu vào của một layer liền kề; m là số node trong lớp , với . là đầu ra của node thứ j; là đầu vào từ node; là trọng số; là hệ số tự do (bias).

Quá trình thực nghiệm trên mô hình DFNN sử dụng ý tưởng của Grid Search để tối ưu hóa các tham số, bộ tham số tốt nhất được lựa chọn dựa trên độ đo MSE trên tập dữ liệu validate.

## TCANs

Theo Lin và đồng nghiệp (07/2021) [24] Temporal Convolutional Attention Neural Networks (TCANs) gồm 3 phần: Các lớp tích chập thời gian (Temporal Convolution Layers), lớp chú ý thưa thớt (Sparse Attention Layer) và lớp đầu ra (Output Layer).



Hình 6. Kiến trúc TCANs

Các lớp tích chập thời gian áp dụng phép tích chập để trích xuất đặc trưng từ chuỗi thời gian đầu vào, sử dụng cửa sổ và bước nhảy khác nhau để mã hóa mẫu thời gian quan trọng. Lớp chú ý thưa thớt tập trung vào các điểm dữ liệu quan trọng trong chuỗi thời gian và tạo ra trọng số chú ý cho từng điểm dữ liệu để giải thích kết quả dự đoán. Lớp đầu ra kết hợp đặc trưng và trọng số chú ý từ các lớp trước đó để tính toán đầu ra cho bài toán chuỗi thời gian.

TCANs sử dụng kiến trúc tích chập phân cấp để mã hóa chuỗi đầu vào và trích xuất mẫu thời gian dưới dạng biến tiềm ẩn (latent variables).

Bằng việc sử dụng lớp chú ý thưa thớt, TCANs có khả năng truy cập vào tất cả các bước lịch sử của chuỗi đầu vào mà không phụ thuộc vào độ dài của chuỗi. Điều này cho phép TCANs tập trung vào những bước quan trọng nhất của chuỗi đầu vào và cung cấp một hình ảnh trực quan về kết quả, giúp dễ dàng hiểu và giải thích kết quả dự đoán.

Nhóm sử dụng Optuna, một thư viện tối ưu siêu tham số (hyperparameter optimization) theo phương pháp tìm kiếm Bayesian (Bayesian search). Mục tiêu là tìm ra các giá trị siêu tham số tối ưu để cải thiện hiệu suất dự đoán trên dữ liệu validate dựa trên Mean Squared Error (MSE). Quá trình tối ưu hóa được thực hiện bằng cách thử nghiệm và so sánh các giá trị siêu tham số khác nhau và lưu giữ cấu hình tốt nhất để sử dụng trong quá trình đánh giá và triển khai mô hình.

## Độ đo

Để đánh giá khả năng dự đoán của các mô hình đề xuất, nhóm sử dụng ba phép đo hiệu suất bao gồm: Mean Absolute Percentage Error (RMSE), Root Mean Square Error (MAPE) và Mean Squared Logarithmic Error (MSLE).

A. de Myttenaere [25] đã mô tả công thức MAPE đo lường sai số tuyệt đối trung bình giữa giá trị dự đoán và giá trị thực tế với công thức như sau:

T. Chai [26] đã đề cập đến công thức RMSE đo lường khoảng cách trung bình giữa các giá trị dự đoán và thực tế với công thức như sau:

đo lường khoảng cách trung bình giữa các giá trị dự đoán và thực tế ở dạng logarit với công thức [27]:

Trong đó:

, : Giá trị dự đoán cho một quan sát cụ thể.

Giá trị thực sự (đúng) của mục tiêu đang dự đoán.

: Số lượng quan sát trong dữ liệu.

# **THỰC NGHIỆM**

## Thực nghiệm trên ba bộ dữ liệu

A picture containing text, diagram, line, font

Description automatically generated

Hình 7. Kết quả thực nghiệm Linear Regression

A picture containing text, line, diagram, plot

Description automatically generated

Hình 8. Kết quả thực nghiệm ARIMA

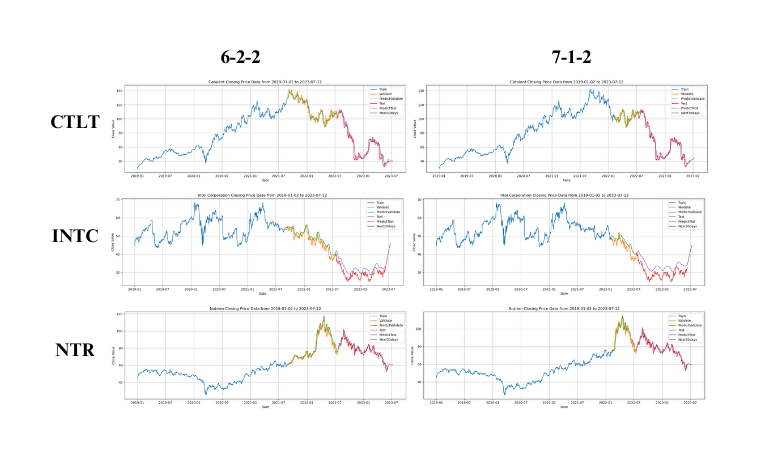


Hình 9. Kết quả thực nghiệm GRU

A picture containing text, diagram, line, plot

Description automatically generated

Hình 10. Kết quả thực nghiệm ETS

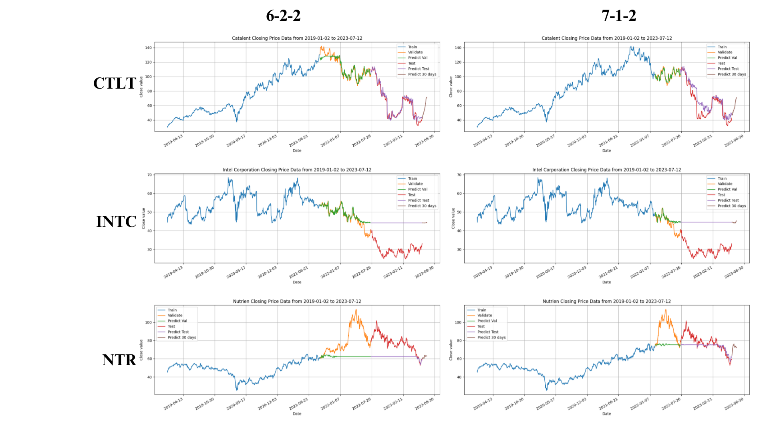


Hình 11. Kết quả thực nghiệm LSTM

A picture containing text, diagram, line, plot

Description automatically generated

Hình 12. Kết quả thực nghiệm RNN



Hình 13. Kết quả thực nghiệm GBT

A picture containing text, diagram, line, plot

Description automatically generated

Hình 14. Kết quả thực nghiệm HMM



Hình 15. Kết quả thực nghiệm XgBoost

A picture containing text, diagram, line, plot

Description automatically generated

Hình 16. Kết quả thực nghiệm DFNN

A picture containing text, diagram, line, plot

Description automatically generated

Hình 17. Kết quả thực nghiệm TCANs

Kết quả thực nghiệm sau khi thu thập được đánh giá dựa trên ba độ đo và đề cập dưới bảng sau:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | | CTLT | | INTC | | NTR | |
| Tỉ lệ chia | | 7:1:2 | 6:2:2 | 7:1:2 | 6:2:2 | 7:1:2 | 6:2:2 |
| Mô hình | Độ đo |  | | | | | |
| LR | MAPE | 1.89 | 1.93 | 0.91 | 1.066 | 0.17 | 0.307 |
| RMSE | 103.1 | 105 | 26.69 | 31.3 | 15.77 | 26.57 |
| MSLE | 1.102 | 1.13 | 0.402 | 0.505 | 0.044 | 0.154 |
| GRU | MAPE | 0.046 | 0.053 | 0.043 | 0.057 | 0.024 | 0.024 |
| RMSE | 3.832 | 4.467 | 1.508 | 1.891 | 2.552 | 2.553 |
| MSLE | 0.004 | 0.006 | 0.002 | 0.003 | 0.001 | 0.001 |
| ARIMA | MAPE | 0.79 | 1.617 | 0.656 | 0.832 | 0.186 | 0.211 |
| RMSE | 43.424 | 88.4 | 19.3 | 24.43 | 21.81 | 19.13 |
| MSLE | 0.372 | 0.92 | 0.246 | 0.353 | 0.061 | 0.069 |
| ETS | MAPE | 0.031 | 0.031 | 0.02 | 0.02 | 0.02 | 0.02 |
| RMSE | 2.853 | 2.853 | 0.77 | 0.77 | 2.1 | 2.1 |
| MSLE | 0.002 | 0.002 | 6e-4 | 6e-4 | 7e-4 | 7e-4 |
| LSTM | MAPE | 0.162 | 0.170 | 0.493 | 0.410 | 0.041 | 0.046 |
| RMSE | 0.048 | 0.049 | 0.1 | 0.072 | 0.033 | 0.036 |
| MSLE | 0.002 | 0.002 | 0.008 | 0.004 | 4e-4 | 5e-4 |
| RNN | MAPE | 0.996 | 0.996 | 0.997 | 0.997 | 0.992 | 0.992 |
| RMSE | 67.75 | 71.65 | 33.96 | 33.56 | 77.63 | 75.54 |
| MSLE | 15.17 | 15.44 | 11.88 | 11.8 | 15.22 | 15.01 |
| GBT | MAPE | 0.146 | 0.060 | 0.512 | 0.500 | 0.068 | 0.192 |
| RMSE | 10.077 | 4.350 | 15.144 | 14.776 | 7.804 | 17.854 |
| MSLE | 0.031 | 0.007 | 0.167 | 0.161 | 0.008 | 0.058 |
| HMM | MAPE | 0.025 | 0.024 | 0.018 | 0.017 | 0.018 | 0.017 |
| RMSE | 2.004 | 1.976 | 0.669 | 0.655 | 1.807 | 1.753 |
| MSLE | 0.001 | 9e-4 | 5e-4 | 5e-4 | 5e-4 | 5e-4 |
| XGBoost | MAPE | 0.043 | 0.041 | 0.312 | 0.318 | 0.105 | 0.245 |
| RMSE | 3.787 | 3.476 | 13.73 | 14.03 | 10.23 | 17.48 |
| MSLE | 0.005 | 0.004 | 0.144 | 0.149 | 0.016 | 0.056 |
| DFNN | MAPE | 0.073 | 0.071 | 0.061 | 0.05 | 0.024 | 0.033 |
| RMSE | 4.956 | 6.317 | 2.044 | 1.807 | 2.125 | 3.364 |
| MSLE | 0.009 | 0.01 | 0.004 | 0.003 | 0.000 | 0.002 |
| TCANs | MAPE | 0.079 | 0.07 | 0.072 | 0.053 | 0.021 | 0.02 |
| RMSE | 5.175 | 5.287 | 2.362 | 1.988 | 2.206 | 2.092 |
| MSLE | 0.01 | 0.011 | 0.006 | 0.005 | 8e-4 | 7e-4 |

Bảng . Đánh giá kết quả thực nghiệm

Bảng trên ghi nhận các giá trị độ đo MAPE, RMSE, MSLE của các mô hình Linear Regression, Gated Recurrent Unit (GRU), Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing (ETS), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Gradient Boosted Tree (GBT), Hidden Markov Model (HMM), Extreme Gradient Boosting (XGBoost), Deep Feedforward Neural Network (DFNN), Temporal Convolutional Attention Networks (TCANs) trên tập test của ba bộ dữ liệu CTLT, INTC và NTR theo hai tỉ lệ của train:validate:test là 7:1:2 và 6:2:2.

Trong 3 bộ dữ liệu đã xét, các giá trị tốt nhất (có giá trị thấp nhất) cho độ đo MAPE, RMSE và MSLE được đánh dấu bằng màu đỏ.

Hidden Markov Model (HMM) đạt được giá trị Mean Absolute Percentage Error (MAPE) thấp nhất trên cả ba bộ dữ liệu và cũng như với hai tỉ lệ train:val:test.

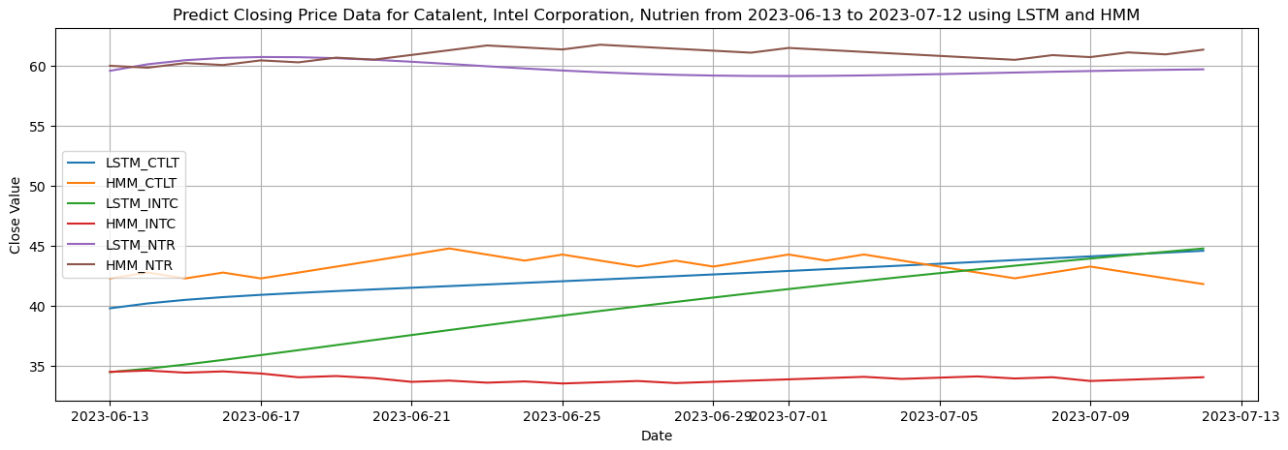
Long Short-Term Memory (LSTM) là mô hình cho giá trị Root Mean Square Error (RMSE) thấp nhất trên cả ba bộ dữ liệu và cũng như với hai tỉ lệ train:val:test.

Trong trường hợp của Mean Squared Logarithmic Error (MSLE), Hidden Markov Model (HMM) là mô hình cho kết quả thấp nhất trên hầu hết ba bộ dữ liệu và cũng như với hai tỉ lệ train:val:test. Tuy nhiên, đối với bộ dữ liệu NTR ở tỉ lệ 7:1:2 của train:val:test, LSTM cho giá trị thấp nhất là 0.00041 (nhỏ hơn 0.00053 của HMM).

Tổng kết lại, từ các phân tích về các độ đo lỗi như MAPE, RMSE và MSLE trên cả ba bộ dữ liệu và hai tỉ lệ train:val:test, nhóm nhận thấy hai mô hình phù hợp nhất là Long Short-Term Memory (LSTM) và Hidden Markov Model (HMM). Cả hai mô hình này đã cho kết quả tốt nhất trong các độ đo lỗi khác nhau trên hầu hết ba bộ dữ liệu và tỉ lệ train:val:test.

Nhóm sẽ sử dụng hai mô hình này để giải quyết bài toán dự đoán giá trị giá đóng cửa (Close) trong 30 ngày tiếp theo, từ ngày 13/6/2023 đến 12/7/2023 trên 3 bộ dữ liệu CTLT, INTC và NTR.

## Thực nghiệm dự đoán giá cổ phiếu 30 ngày tới



Hình 18. Dự đoán giá cổ phiếu trong 30 ngày tới với tỉ lệ 7:1:2

A picture containing line, plot, text, diagram

Description automatically generated

Hình 19. Dự đoán giá cổ phiếu trong 30 ngày tới với tỉ lệ 6:2:2

# **KẾT LUẬN**

Trong bài báo này, chúng tôi đã tiến hành nghiên cứu về việc sử dụng kỹ thuật phân tích chuỗi thời gian để dự đoán giá cổ phiếu. Chúng tôi đã sử dụng các mô hình như Linear Regression, GRU, ARIMA, ETS, LSTM, RNN, GBT, HMM, XGBoost, DFNN, và TCANs trên ba bộ dữ liệu khác nhau để đưa ra dự đoán giá cổ phiếu. Điều này giúp chúng tôi đánh giá hiệu suất và so sánh các mô hình dự đoán. Kết quả thực nghiệm cho thấy mô hình LSTM và HMM đã cho thấy hiệu suất tốt hơn so với các mô hình khác trong việc dự đoán giá cổ phiếu. Điều này đều chỉ ra tiềm năng của các mô hình dựa trên mạng nơ-ron sâu và kỹ thuật phân tích chuỗi thời gian trong lĩnh vực dự đoán giá cổ phiếu.

Mặc dù đã đạt được một số kết quả khả quan, quá trình nghiên cứu không tránh khỏi một số khó khăn. Một trong những khó khăn là tính phức tạp và biến động của thị trường tài chính, làm tăng độ khó trong việc dự đoán giá cổ phiếu.

Trong tương lai, chúng tôi sẽ tiếp tục nghiên cứu, áp dụng các kỹ thuật trong tinh chỉnh mô hình để cải tiến các mô hình dự đoán giá cổ phiếu. Ngoài ra chúng tôi có thể nghiên cứu và áp dụng các mô hình mới nhất và phát triển phương pháp kết hợp giữa các mô hình khác nhau để tăng độ chính xác và tin cậy của dự đoán. Đồng thời, mở rộng phạm vi nghiên cứu bằng cách sử dụng thêm nhiều dữ liệu từ các thị trường tài chính khác nhau như tin tức, sự kiện và thông tin khác để dự đoán chính xác hơn.

# **ĐÓNG GÓP CỦA CÁC THÀNH VIÊN**

**Trần Thị Mỹ Nhung**: Kiểm tra – Xem xét và chỉnh sửa, kết luận, ARIMA, DFNN. **Nguyễn Hoài Linh**:Trực quan hóa dự đoán 30 ngày, Tài nguyên, HMM, RNN. **Nguyễn Thành Phát**: Trực quan hóa dự đoán 30 ngày, Tài nguyên, LSTM, XGBoost. **Nguyễn Đạt**: Giới thiệu, độ đo, GRU, GBT. **Nguyễn Anh Kiệt**: Đánh giá thuật toán, Linear Regression, ETS, TCANs.

# **LỜI CẢM ƠN**

Chúng tôi xin bày tỏ lòng biết ơn sâu sắc đến PGS. TS. Nguyễn Đình Thuân và Kỹ sư Nguyễn Minh Nhựt vì sự hướng dẫn, định hướng và đóng góp quý báu của họ trong quá trình thực hiện bài báo này. Sự chỉ dẫn chuyên nghiệp và kiến thức sâu rộng của PGS. TS. Nguyễn Đình Thuân và Kỹ sư Nguyễn Minh Nhựt đã mang đến cho chúng tôi những cơ sở lý thuyết và phương pháp nghiên cứu quan trọng, giúp chúng tôi tiếp cận và hiểu sâu hơn về kỹ thuật phân tích chuỗi thời gian và dự đoán giá cổ phiếu. Sự tận tâm và sự hỗ trợ từ PGS. TS. Nguyễn Đình Thuân và Kỹ sư Nguyễn Minh Nhựt là một phần không thể thiếu trong thành công của chúng tôi.

Bên cạnh đó, chúng tôi cũng muốn bày tỏ lòng biết ơn đến các thành viên trong nhóm nghiên cứu. Sự đồng lòng, tương trợ và chia sẻ kiến thức giữa chúng tôi đã tạo ra một môi trường làm việc tích cực và động lực để vượt qua các thách thức trong quá trình nghiên cứu. Những ý kiến đóng góp và thảo luận trong nhóm đã giúp chúng tôi mở rộng tầm nhìn và hoàn thiện nội dung bài báo.

Cuối cùng, chúng tôi xin gửi lời cảm ơn tới tất cả những người đã đóng góp và hỗ trợ chúng tôi trong hành trình này. Những đóng góp và hỗ trợ này đã góp phần quan trọng vào thành công của bài báo và mang lại giá trị cho cả lĩnh vực nghiên cứu về dự đoán giá cổ phiếu. Chúng tôi hy vọng rằng công trình này sẽ được lan tỏa và tiếp tục khám phá những tiềm năng và ứng dụng mới trong lĩnh vực này.

# **TÀI LIỆU THAM KHẢO**

1. A. Beattie, "What Was the First Company to Issue Stock?", Investopedia, [Online]. Available: https://www.investopedia.com/ask/answers/08/first-company-issue-stock-dutch-east-india.asp.
2. Gururaj, V., Shriya, V. R., & Ashwini, K. (2019). Stock market prediction using linear regression and support vector machines. Int J Appl Eng Res, 14(8), 1931-1934.
3. Disha, R. A., & Waheed, S. (2022). Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique. Cybersecurity, 5(1), 1-20.
4. A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.
5. A. Meyler, G. Kenny and T. Quinn, “Forecasting Irish Inflation using ARIMA Models”, Central Bank of Ireland Research Department, Technical Paper, 3/RT/1998.
6. Sidqi, F., & Sumitra, I. D. (2019, November). Forecasting product selling using single exponential smoothing and double exponential smoothing methods. In IOP Conference Series: Materials Science and Engineering (Vol. 662, No. 3, p. 032031). IOP Publishing.
7. Roondiwala, M., Patel, H., & Varma, S. (2017, April). Predicting Stock Prices Using LSTM. IJSC publishing.
8. Yongqiong Zhu. 2020. Stock price prediction using the RNN model. 2020 International Conference on Applied Physics and Computing (ICAPC 2020). IOP Publishing.
9. Raihan Tanvir , Md Tanvir Rouf Shawon and Md. Golam Rabiul Alam. (2023, 26 January). DSE Stock Price Prediction using Hidden Markov Model. Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
10. Kyung Keun Yun, Yoon,Sang Won Yoon, & Daehan Won(2021, August). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. Department of Systems Science and Industrial Engineering publishing.
11. Orimoloye, L. O., Sung, M. C., Ma, T., & Johnson, J. E. (2020). Comparing the effectiveness of deep feedforward neural networks and shallow architectures for predicting stock price indices. Expert Systems with Applications, 139, 112828.
12. Lin, Y., Koprinska, I., & Rana, M. (2021, July). Temporal convolutional attention neural networks for time series forecasting. In 2021 International joint conference on neural networks (IJCNN) (pp. 1-8). IEEE.
13. Sharma, A., Bhuriya, D., & Singh, U. (2017, April). Survey of stock market prediction using machine learning approach. In 2017 International conference of electronics, communication and aerospace technology (ICECA) (Vol. 2, pp. 506-509). IEEE.
14. Mahjoub, S., Chrifi-Alaoui, L., Marhic, B., & Delahoche, L. (2022). Predicting Energy Consumption Using LSTM, Multi-Layer GRU and Drop-GRU Neural Networks. Sensors, 22(11), 4062.
15. Box G, Jenkins G. Time Series Analysis, Forecasting and Control. San Francisco, CA: Holden-Day; 1970
16. Wang, J. J., Wang, J. Z., Zhang, Z. G., & Guo, S. P. (2012). Stock index forecasting based on a hybrid model. Omega, 40(6), 758-766.
17. Zhi-Peng, L. I., Hong, Y. U., Yun-Cai, L. I. U., & Fu-Qiang, L. I. U. (2008). An improved adaptive exponential smoothing model for short-term travel time forecasting of urban arterial street. Acta automatica sinica, 34(11), 1404-1409.
18. Alex Graves and Jurgen Schmidhuber, (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks, 18(5-6), pp.602-610.
19. Shervine Amidi. (2018). Recurrent Neural Networks - CS230 Cheatsheet. Available:https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-recurrent-neural-networks.
20. L. Breiman, "Arcing The Edge," Statistics Department, University of California, Berkeley, Technical Report 486, June 1997.
21. Chen, T., & Guestrin, C. (2016, June). XGBoost: A Scalable Tree Boosting System. Carlos Guestrin. KDD ’16. New York, USA: ACM Press, p. 785–794
22. Wójcik, Filip. (2020). XGBoost as a time-series forecasting tool.
23. Oyedele, A. A., Ajayi, A. O., Oyedele, L. O., Bello, S. A., & Jimoh, K. O. (2023). Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. Expert Systems with Applications, 213, 119233.
24. Lin, Y., Koprinska, I., & Rana, M. (2021, July). Temporal convolutional attention neural networks for time series forecasting. In 2021 International joint conference on neural networks (IJCNN) (pp. 1-8). IEEE.
25. A. de Myttenaere, B. Golden, B. Le Grand, và F. Rossi, "Mean Absolute Percentage Error for Regression Models," Neurocomputing​.​
26. T. Chai và R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," Geosci. Model Dev., vol. 7, pp. 1247–1250, 2014.
27. "MSLE - Mean Squared Logarithmic Error," Permetrics 1.2.0 documentation

1. <https://finance.yahoo.com/quote/CTLT/history?p=CTLT> [↑](#footnote-ref-2)
2. https://finance.yahoo.com/quote/INTC/history?p=INTC [↑](#footnote-ref-3)
3. https://finance.yahoo.com/quote/NTR/history?p=NTR [↑](#footnote-ref-4)