

# Phân tích, dự đoán giá chứng khoán bằng mô hình thống kê

Trần Thị Mỹ Xoan - 21522815

Lê Anh Tuấn Dũng - 21521974

Lê Thị Ánh Hồng - 21520245

Nguyễn Thị Mai Liên - 21522283

Đỗ Sĩ Đạt - 21521932

Ngày 20 tháng 6 năm 2024

## Tóm tắt nội dung:

Khi thực hiện đầu tư vào các cổ phiếu chứng khoán, việc sử dụng các mô hình dự đoán giá cổ phiếu trước khi thực hiện một giao dịch đầu tư có thể giúp cho ta có các quyết định đúng đắn hơn khi ra quyết định thực hiện 1 giao dịch. Bài báo này trình bày phương pháp dự đoán giá cổ phiếu của ba công ty lớn Samsung, LG và Sony. Chúng tôi sẽ thực hiện dự đoán bằng cách sử dụng các mô hình thống kê, học máy, học sâu: VARMA, XGBoost, NBeats, Gradient Boosting, LightGBM. Bộ dữ liệu được sử dụng lấy từ ngày 1/3/2019 đến ngày 1/6/2024. Sau khi thực hiện dự đoán thì ta sẽ thực hiện sử dụng các độ đo MAPE, MSE, RMSE để đánh giá các mô hình. Cuối cùng, sử dụng các mô hình để thực hiện dự đoán giá chứng khoán cho 30, 60 và 90 ngày tiếp theo.

**Từ khoá:** VARMA, XGBoost, NBeats, Gradient Boosting, LightGBM. Mô hình thống kê, mô hình học máy, mô hình học sâu, dự đoán giá chứng khoán.

## 1 GIỚI THIỆU

Dự đoán giá cổ phiếu là một vấn đề quan trọng trong lĩnh vực đầu tư tài chính. Việc dự đoán chính xác giá cổ phiếu có thể giúp các nhà đầu tư đưa ra quyết định đầu tư sáng suốt và giảm thiểu rủi ro. Có nhiều phương pháp khác nhau để dự đoán giá cổ phiếu, bao gồm phân tích kỹ thuật, phân tích cơ bản và học máy. Samsung, LG, SONY là những công ty hàng đầu trong ngành công nghệ, giá cổ phiếu của 3 công ty đóng vai trò là chỉ số quan trọng về động lực thị trường, khiến việc phân tích của họ trở nên cần thiết đối với các nhà đầu tư.

Báo cáo này nhằm mục đích dự đoán giá cổ phiếu của các công ty có ảnh hưởng bằng cách sử dụng 5 thuật toán VARMA, XGBoost, NBeats, Gradient Boosting, LightGBM để đưa ra dự đoán cho giá cổ

phiếu. Nhờ đó các nhà đầu tư có thể lựa chọn chính xác hạng mục đầu tư vào giá cổ phiếu.

Bằng cách tận dụng những phương pháp này, chúng tôi mong muốn cung cấp hướng dẫn có giá trị cho các nhà đầu tư trong việc định hướng bối cảnh năng động của lĩnh vực công nghệ.

## 2 NGHIÊN CỨU LIÊN QUAN

[1] Thực hiện giả định có hai cách tiếp cận giá trị đầu vào cho mô hình, dữ liệu liên tục và dữ liệu nhị phân. Ở hướng tiếp cận thứ nhất với dữ liệu liên tục, kết quả cho ra là RNN và LSTM là hai yếu tố dự đoán hàng đầu (khoảng 86% điểm F1). Ở hướng tiếp cận thứ hai với dữ liệu nhị phân thì hai yếu tố dự đoán tốt nhất là RNN và LSTM (với khoảng 90% điểm F1) và quá trình dự đoán cho tất cả các mô hình

đều nhanh hơn. Các công trình thử nghiệm cho thấy sự cải thiện đáng kể về hiệu suất của các mô hình khi sử dụng dữ liệu nhị phân thay vì dữ liệu liên tục.

[2]: Bài báo này đã tham gia vào cuộc tranh luận về tính hữu ích của phân tích tâm lý đối với việc dự đoán diễn biến thị trường chứng khoán. Bài báo đã sử dụng dữ liệu Twitter làm kho thông tin của mình để dự đoán những thăng trầm của sáu công ty NASDAQ nổi tiếng. Bài báo đã đề xuất một phương pháp bắt đầu bằng cách trích xuất nhiều đặc điểm dựa trên văn bản để làm phong phú thêm việc thể hiện tình cảm. Sau đó kết quả thực nghiệm đã cho ra kết luận rằng sự tăng giảm giá cổ phiếu của một công ty bị ảnh hưởng bởi quan điểm hoặc công chúng thể hiện trên Twitter.

[3]: Bài báo này nhằm mục đích dự đoán hướng đi của giá cổ phiếu Mỹ bằng cách tích hợp entropy truyền hiệu quả thay đổi theo thời gian (ETE) và các thuật toán học máy khác nhau. Đầu tiên, bài báo khám phá rằng ETE dựa trên thời hạn biến động 3 và 6 tháng có thể được coi là biến giải thích thị trường bằng cách phân tích mối liên hệ giữa các cuộc khủng hoảng tài chính và mối quan hệ nhân quả Granger giữa các cổ phiếu. Sau đó, bài báo phát hiện ra rằng hiệu suất dự đoán theo hướng giá cổ phiếu có thể được cải thiện khi biến điều khiển ETE được tích hợp như một tính năng mới trong hồi quy logistic, perceptron đa lớp, rừng ngẫu nhiên, XGBoost và mạng bộ nhớ ngắn hạn dài. Cuối cùng, bài báo xác nhận rằng mạng perceptron đa lớp và bộ nhớ ngắn hạn dài phù hợp hơn cho việc dự đoán giá cổ phiếu. Nghiên cứu này là nỗ lực đầu tiên nhằm dự đoán hướng giá cổ phiếu bằng cách sử dụng ETE, có thể áp dụng thuận tiện vào thực tế.

[4] Mô hình VARMA thường được sử dụng để dự báo dữ liệu chuỗi thời gian đa biến. Mô hình VARMA là sự mở rộng của mô hình ARMA trong chuỗi thời gian đơn biến (Lütkepohl, 2005; Wei, 1990) và được sử dụng với điều kiện rằng dữ liệu phải ổn định theo thời gian (Lütkepohl, 2005). Mô hình VARMA (p,q) là sự kết hợp của mô hình VAR (p) và mô hình vector trung bình động (q) (VMA (q)). Bài báo này xác định 4 mô hình VARMA (p,q) lần lượt là (1,1), (2, 1), (3, 1), (4, 1), Việc lựa chọn mô hình tốt nhất được thực hiện bằng cách sử dụng một số tiêu chí thông tin (AICC, HQC, AIC, và SBC). Các giá trị nhỏ nhất của những tiêu chí này cho biết mô hình tốt nhất. Mô hình phù hợp nhất được chọn là VARMA(2, 1). Kết quả dự báo cho thấy sai số chuẩn tăng lên theo thời gian; sai số chuẩn trong tháng đầu tiên tương đối nhỏ so với dự đoán trung bình, nhưng tăng lên theo thời gian đến dự báo cho 12 tháng tiếp theo. Điều này cho thấy mô hình là hợp lý khi dự báo cho các khoảng thời gian ngắn, nhưng kết quả không ổn định (do sai

số chuẩn cao hơn) khi dự báo cho các khoảng thời gian dài.

[5] Và để tìm ra tham số tối ưu cho mô hình VARMA, ta sử dụng `auto_arima` để thực hiện tìm các tham số tối ưu ( $p, q$ ) cho mỗi biến. Cặp tham số của một biến có AIC nhỏ nhất là phù hợp nhất. Sau đó, huấn luyện với các cặp tham số ( $p, q$ ) tìm được. Sau khi chạy, ta sử dụng chỉ số đánh giá RMSE để xem cặp tham số ( $p, q$ ) nào phù hợp nhất với mô hình. RMSE càng nhỏ thì cặp tham số đó càng phù hợp.

### 3 NGUYÊN LIỆU

#### 3.1 DATASET

Trọng tâm của bài viết này là dự đoán giá cổ phiếu. Vì vậy, cả ba tập dữ liệu chúng tôi sử dụng trong nghiên cứu đều là giá cổ phiếu của các công ty lớn. Các công ty này là SONY, SAMSUNG và LG. Dữ liệu của SAMSUNG, LG, SONY được thu thập từ ngày 1 tháng 3 năm 2019 đến ngày 1 tháng 6 năm 2024.

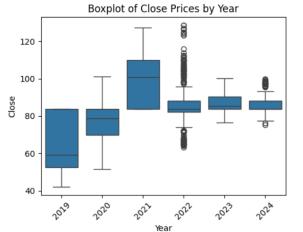
Mô tả từng cột trong dữ liệu :

- + Date : Ngày giao dịch.
- + Open: Giá mở cửa của cổ phiếu vào ngày giao dịch.
- + High: Giá cao nhất của cổ phiếu trong ngày giao dịch.
- + Low: Giá thấp nhất của cổ phiếu trong ngày giao dịch.
- + Close: Giá đóng cửa của cổ phiếu vào ngày giao dịch.

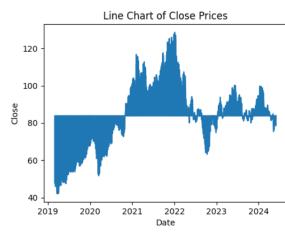
#### 3.2 THỐNG KÊ MÔ TẢ

|       | SONY      | SAMSUNG    | LG            |
|-------|-----------|------------|---------------|
| Count | 1920      | 1920       | 1920          |
| Mean  | 83.842011 | 83.842011  | 101952.316347 |
| Std   | 15.828603 | 15.828603  | 26596.447698  |
| Min   | 42.030000 | 42.030000  | 41850         |
| 25%   | 79.987500 | 79.987500  | 87200         |
| 50%   | 83.842011 | 83.842011  | 101952.316347 |
| 75%   | 90.442500 | 90.442500  | 114000        |
| Max   | 128.59    | 128.590000 | 185000        |

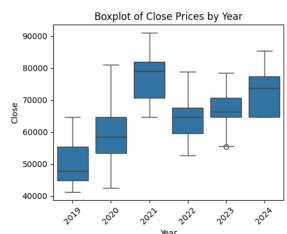
Bảng 1: SONY, SAMSUNG, LG's Descriptive Statistics



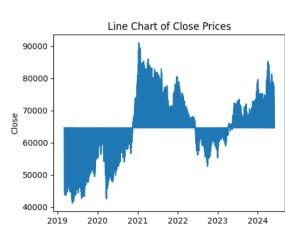
Hình 1: SONY stock price's boxplot



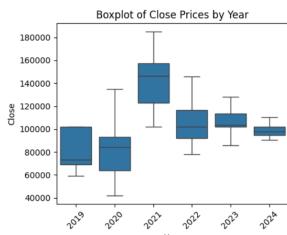
Hình 2: SONY stock price's histogram



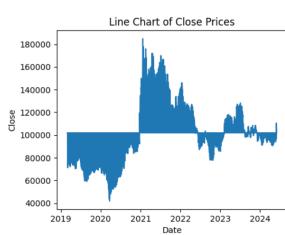
Hình 3: SAMSUNG stock price's boxplot



Hình 4: SAMSUNG stock price's histogram



Hình 5: LG stock price's boxplot



Hình 6: LG stock price's histogram

### 3.3 TỶ LỆ PHÂN CHIA TẬP DỮ LIỆU

Tỷ lệ phân chia dữ liệu là một quyết định quan trọng trong việc xây dựng mô hình học máy. Trong bài nghiên cứu này, chúng tôi sẽ chia dữ liệu theo ba tỷ lệ khác nhau: 6:4, 7:3 và 8:2. Tỷ lệ 6:4, cho phép chúng ta có một tập kiểm thử tương đối lớn, giúp đánh giá mô hình một cách toàn diện hơn. Tuy nhiên, điều này cũng có nghĩa là chúng ta có ít dữ liệu hơn để huấn luyện mô hình, có thể ảnh hưởng đến khả năng học của mô hình. Tỷ lệ 7:3 cho phép chúng ta huấn luyện mô hình trên một lượng dữ liệu lớn hơn, trong khi vẫn đảm bảo có đủ dữ liệu để kiểm thử mô hình. Điều này giúp cải thiện hiệu suất của

mô hình và khả năng tổng quát hóa. Tỷ lệ 8:2 cung cấp một sự cân nhắc giữa việc huấn luyện mô hình và đánh giá toàn diện trên tập dữ liệu kiểm thử, cung cấp một lựa chọn linh hoạt cho nhiều tình huống khác nhau. Nghiên cứu của chúng tôi nhấn mạnh tầm quan trọng của việc xem xét tỷ lệ phân chia dữ liệu để đảm bảo hiệu quả trong cả việc huấn luyện và đánh giá mô hình. Phương pháp tiếp cận tinh tế này giúp đảm bảo rằng mô hình của chúng tôi có khả năng cung cấp dự đoán chính xác và đáng tin cậy trong các tình huống thực tế. Đây không chỉ là một phần quan trọng của quá trình nghiên cứu của chúng tôi mà còn là một bước quan trọng để đảm bảo rằng kết quả của chúng tôi có thể được áp dụng và triển khai hiệu quả trong thực tế.

### 3.4 CHỈ SỐ ĐÁNH GIÁ MÔ HÌNH

#### 3.4.1 Mean Squared Error(MSE)

Mean Squared Error (MSE) là một số liệu phổ biến được sử dụng trong các bài toán hồi quy. Về cơ bản, MSE tính sai số bình phương trung bình giữa các giá trị được dự đoán và giá trị thực tế. Đây là một thước đo chất lượng của mô hình dự đoán, luôn không âm và các giá trị càng gần 0 càng tốt.

Công thức tính MSE:

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

trong đó:

- $n$  là số điểm dữ liệu
- $y_i$  là giá trị quan sát
- $\hat{y}_i$  là giá trị dự đoán

Trong phân tích hồi quy, việc vẽ biểu đồ là một cách trực quan để xem xu hướng chung của dữ liệu. MSE cho bạn biết mức độ gần của đường hồi quy với tập hợp các điểm dữ liệu. Cụ thể, MSE đo khoảng cách từ các điểm dữ liệu đến đường hồi quy (các khoảng cách này là "sai số") và bình phương chúng. Việc bình phương sai số là rất quan trọng vì nó loại bỏ các dấu âm và làm tăng trọng số của các sai số lớn.

Để giảm thiểu MSE, mô hình cần dự đoán chính xác hơn, tức là gần với dữ liệu thực tế hơn. Một ví dụ về hồi quy tuyến tính sử dụng phương pháp này là phương pháp bình phương nhỏ nhất. Phương pháp này đánh giá sự phù hợp của mô hình hồi quy tuyến tính với tập dữ liệu hai biến. Tuy nhiên, nó có giới hạn liên quan đến phân phối dữ liệu đã biết.

MSE càng thấp thì dự báo càng tốt.

### 3.4.2 Root Mean Square Error(RMSE)

Root Mean Square Error (RMSE) là căn bậc hai của mức trung bình các sai số bình phương. RMSE là độ lệch chuẩn của các phần dư (sai số dự đoán). Phần dư đo khoảng cách từ các điểm dữ liệu đến đường hồi quy, RMSE đo mức độ phân tán của các phần dư này, tức là mức độ tập trung của dữ liệu xung quanh đường hồi quy.

Công thức tính RMSE:

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

trong đó:

- $n$  là số điểm dữ liệu
- $y_i$  là giá trị thực tế
- $\hat{y}_i$  là giá trị dự đoán

Ảnh hưởng của mỗi lỗi đối với RMSE tỷ lệ với kích thước của lỗi bình phương, do đó các sai số lớn hơn có ảnh hưởng lớn đến RMSE một cách không cân xứng. Vì lý do này, RMSE nhạy cảm với các yếu tố ngoại lai. RMSE thường được sử dụng trong khí hậu học, dự báo, và phân tích hồi quy để xác minh kết quả thực nghiệm.

Khi các quan sát và dự báo được chuẩn hóa, RMSE có mối quan hệ trực tiếp với hệ số tương quan. Ví dụ, nếu hệ số tương quan là 1, RMSE sẽ bằng 0 vì tất cả các điểm nằm trên đường hồi quy (không có sai số).

RMSE luôn không âm, và giá trị 0 (hầu như không bao giờ đạt được trong thực tế) chỉ ra sự phù hợp hoàn hảo với dữ liệu. Nói chung, RMSE thấp hơn cho thấy mô hình tốt hơn so với RMSE cao hơn.

### 3.4.3 Mean Absolute Percentage Error(MAPE)

Mean Absolute Percentage Error (MAPE) là một số liệu được sử dụng để đo lường độ chính xác của các dự đoán trong các mô hình dự báo. MAPE tính toán sai số tuyệt đối trung bình dưới dạng phần trăm của giá trị thực tế, giúp đánh giá mức độ chính xác của dự đoán một cách dễ hiểu và so sánh dễ dàng giữa các tập dữ liệu khác nhau.

Công thức tính MAPE:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

trong đó:

- $n$  là số điểm dữ liệu
- $y_i$  là giá trị thực tế
- $\hat{y}_i$  là giá trị dự đoán

Ưu điểm của MAPE là nó biểu diễn sai số dưới dạng phần trăm, làm cho việc diễn giải và so sánh dễ dàng hơn. Tuy nhiên, MAPE có một số hạn chế, chẳng hạn như việc không xác định khi giá trị thực tế  $y_i$  bằng 0 và có thể bị ảnh hưởng mạnh bởi các giá trị cực nhỏ của  $y_i$ , làm cho sai số phần trăm trở nên rất lớn.

MAPE thường được sử dụng trong các lĩnh vực như dự báo kinh tế, tài chính và quản lý chuỗi cung ứng để đánh giá độ chính xác của các mô hình dự báo.

## 4 PHƯƠNG PHÁP

### 4.1 THUẬT TOÁN RNN

Mạng nơ-ron tái phát (RNN) là một loại mạng nơ-ron nhân tạo được thiết kế đặc biệt để xử lý các chuỗi dữ liệu có thứ tự, như chuỗi thời gian hoặc văn bản. Điểm đặc biệt của RNN là khả năng ghi nhớ và sử dụng thông tin từ các bước trước đó trong chuỗi, làm cho chúng rất hữu ích trong các bài toán như dịch máy, nhận dạng giọng nói và phân tích chuỗi thời gian.

#### 4.1.1 Cấu trúc và hoạt động của RNN

RNN có cấu trúc đặc biệt với các kết nối hồi quy (recurrent connections), cho phép truyền thông tin ngược trở lại từ bước hiện tại về các bước trước đó. Điều này có nghĩa là tại mỗi bước thời gian  $t$ , trạng thái ẩn  $a^{(t)}$  được tính toán dựa trên trạng thái ẩn từ bước trước đó  $a^{(t-1)}$  và đầu vào hiện tại  $x^{(t)}$ .

#### 4.1.2 Công thức của RNN

Công thức tính toán trong RNN có thể được biểu diễn như sau:

1. Trạng thái ẩn  $a^{(t)}$ :

$$a^{(t)} = g_1 \left( W_{aa} a^{(t-1)} + W_{ax} x^{(t)} + b_a \right)$$

2. Đầu ra  $y^{(t)}$ :

$$y^{(t)} = g_2 \left( W_{ya} a^{(t)} + b_y \right)$$

Trong đó:

- $W_{ax}$  và  $W_{aa}$  là ma trận trọng số cho đầu vào và trạng thái ẩn.

- $W_{ya}$  là ma trận trọng số cho đầu ra.
- $b_a$  và  $b_y$  là các bias tương ứng.
- $g_1$  và  $g_2$  là các hàm kích hoạt, thường là hàm sigmoid hoặc hàm tanh cho trạng thái ẩn và hàm softmax cho đầu ra.

RNN rất mạnh mẽ nhưng cũng gặp khó khăn trong việc xử lý các chuỗi dài do vấn đề vanishing gradient. Để giải quyết vấn đề này, các biến thể của RNN như LSTM (Long Short-Term Memory) và GRU (Gated Recurrent Unit) đã được phát triển.

RNN là một công cụ quan trọng trong học máy và trí tuệ nhân tạo, đặc biệt khi làm việc với dữ liệu tuần tự và có cấu trúc thời gian.

## 4.2 THUẬT TOÁN LIGHTGBM

LightGBM là một thuật toán học máy dựa trên cây quyết định, nó là một phần của họ các thuật toán tăng cường gradient.

Công thức cơ bản của LightGBM là:

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

Trong đó:

$\hat{y}_i$  là giá trị dự đoán cho mẫu  $i$

$K$  là số lượng cây (trees) được sử dụng

$f_k$  là cây thứ  $k$  trong tập hợp các cây  $\mathcal{F}$

$x_i$  là vector đặc trưng cho mẫu  $i$

LightGBM tối ưu hóa hàm mất mát bằng cách sử dụng thuật toán Gradient Boosting (tăng cường gradient).

Hàm mất mát thường được định nghĩa như sau:

$$\text{loss}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^n L(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

Trong đó:

$\mathbf{y}$  là vector các giá trị thực của các mẫu

$\hat{\mathbf{y}}$  là vector các giá trị dự đoán

$n$  là số lượng mẫu

$L$  là hàm mất mát cho dự đoán và giá trị thực

$\Omega$  là hàm điều chỉnh (regularization function) cho cây

LightGBM sử dụng các kỹ thuật như Gradient Boosting Decision Trees (GBDT) và histogram-based algorithm để tối ưu hóa hiệu suất tính toán và bộ nhớ.

## 4.3 THUẬT TOÁN VARMA

Mô hình VARMA thường được sử dụng để dự báo dữ liệu chuỗi thời gian đa biến. Mô hình VARMA là một mở rộng của mô hình ARMA trong chuỗi thời gian đơn biến (Lutkepohl, 2005; Wei, 1990) và được sử dụng với điều kiện dữ liệu phải là dừng theo thời gian (Lutkepohl, 2005). Mô hình VARMA ( $p,q$ ) là sự kết hợp của mô hình VAR ( $p$ ) và mô hình trung bình trượt vector ( $q$ ) (VMA ( $q$ )).

- Công thức:

$$\mathbf{y}_t = \mathbf{c} + \sum_{i=1}^p \Phi_i \mathbf{y}_{t-i} + \sum_{j=1}^q \Theta_j \epsilon_{t-j} + \epsilon_t \quad (1)$$

Trong đó:

- $\mathbf{y}_t$  là vector của các biến tại thời điểm  $t$ .
- $\mathbf{c}$  là một vector hằng số.
- $\Phi_i$  là ma trận hệ số của chuỗi thời gian (độ trễ, lag)  $i$  của mô hình AR.
- $\Theta_j$  là ma trận hệ số của chuỗi thời gian (độ trễ, lag)  $j$  của mô hình MA.
- $\epsilon_t$  là vector của các nhiễu ngẫu nhiên tại thời điểm  $t$ .

## 4.4 ARIMA

ARIMA model là viết tắt của cụm từ Autoregressive Intergrated Moving Average. Mô hình sẽ biểu diễn phương trình hồi qui tuyến tính đa biến (multiple linear regression) của các biến đầu vào (còn gọi là biến phụ thuộc trong thống kê) là 2 thành phần chính:

- Auto regression: Kí hiệu là AR. Đây là thành phần tự hồi qui bao gồm tịnh hợp các độ trễ của biến hiện tại. Độ trễ bậc  $p$  chính là giá trị lùi về quá khứ bước thời gian của chuỗi. Độ trễ dài hoặc ngắn trong quá trình AR phụ thuộc vào tham số trễ.
- Moving average: Quá trình trung bình trượt được hiểu là quá trình dịch chuyển hoặc thay đổi giá trị trung bình của chuỗi theo thời gian. Do chuỗi của chúng ta được giả định là dừng nên quá trình thay đổi trung bình thường như là một chuỗi nhiễu trắng. Quá trình moving average sẽ tìm mối liên hệ về mặt tuyến tính giữa các phần tử ngẫu nhiên (stochastic term).

- Intergrated: Là quá trình đồng tích hợp hoặc lấy sai phân. Yêu cầu chung của các thuật toán trong time series là chuỗi phải đảm bảo tính dừng. Hầu hết các chuỗi đều tăng hoặc giảm theo thời gian. Do đó yếu tố tương quan giữa chúng chưa chắc là thực sự mà là do chúng cùng tương quan theo thời gian. Khi biến đổi sang chuỗi dừng, các nhân tố ảnh hưởng thời gian được loại bỏ và chuỗi sẽ dễ dự báo hơn. Để tạo thành chuỗi dừng, một phương pháp đơn giản nhất là chúng ta sẽ lấy sai phân. Một số chuỗi tài chính còn qui đổi sang logarit hoặc lợi suất. Độ của sai phân để tạo thành chuỗi dừng còn gọi là bậc của quá trình đồng tích hợp (order of intergration).

- Công thức:

$$\mathbf{y}_t = \mathbf{c} + \epsilon_t + \sum_{i=1}^p \Phi_i \mathbf{y}_{t-i} + \sum_{j=1}^q \Theta_j \epsilon_{t-j} \quad (2)$$

Trong đó:

- $\mathbf{y}_t$  là giá trị của chuỗi thời gian tại thời điểm  $t$ .
- $\mathbf{c}$  là hằng số.
- $\epsilon_t$  là thành phần lỗi tại thời điểm thời điểm  $t$ .
- $\Phi_i$  là hệ số của thành phần tự hồi quy (AR).
- $\Theta_j$  là hệ số của thành phần trung bình trượt (MA).

## 4.5 THUẬT TOÁN GRADIENT BOOSTING REGRESSION

Gradient Boosting Regressor là một thuật toán học máy mạnh mẽ được sử dụng cho các bài toán hồi quy. Nó kết hợp nhiều mô hình hồi quy đơn giản (thường là các cây quyết định) để tạo ra một mô hình dự đoán mạnh mẽ hơn. Thuật toán này dựa trên phương pháp boosting, một kỹ thuật học tập ensemble, trong đó các mô hình con được xây dựng liên tiếp nhau, với mỗi mô hình mới nhằm mục đích sửa lỗi của mô hình trước đó.

Gradient Boosting là xây dựng mô hình dần dần bằng cách thêm vào các mô hình con mà mỗi mô hình mới tập trung vào việc sửa lỗi (residual) của các dự đoán trước đó. Điều này được thực hiện bằng cách tối ưu hóa gradient của hàm lỗi.

Các bước thực hiện

Khởi tạo mô hình ban đầu:

Bắt đầu với một mô hình đơn giản, thường là giá trị

trung bình của biến mục tiêu  $y$ .

Tính toán residuals:

Residuals là sai số giữa giá trị thực tế và giá trị dự đoán của mô hình hiện tại.

Huấn luyện mô hình con:

Huấn luyện một mô hình con (thường là một cây quyết định) để dự đoán residuals.

Cập nhật mô hình hiện tại:

Cập nhật mô hình hiện tại bằng cách thêm mô hình con mới với một trọng số (learning rate).

Lặp lại:

Lặp lại quá trình cho đến khi đạt được số lượng mô hình con mong muốn hoặc sai số giảm đến mức chấp nhận được.

Ưu điểm và nhược điểm

Ưu điểm:

Hiệu quả cao trong việc giảm sai số và dự đoán chính xác. Linh hoạt với khả năng xử lý nhiều loại dữ liệu khác nhau. Có thể điều chỉnh nhiều tham số để tối ưu hóa hiệu suất.

Nhược điểm:

Tốn nhiều thời gian và tài nguyên tính toán. Dễ bị overfitting nếu không điều chỉnh cẩn thận các tham số. Khó khăn trong việc giải thích mô hình do tính phức tạp cao.

## 4.6 THUẬT TOÁN XGBoost

Thuật toán XGBoost là một thuật toán máy học thuộc loại ensemble learning, cụ thể là Gradient Boosting Framework. Nó sử dụng cây quyết định làm người học cơ sở và sử dụng các kỹ thuật chính quy hóa để nâng cao khả năng khai thác của mô hình. XGBoost được sử dụng rộng rãi cho các tác vụ như hồi quy, phân loại và xếp hạng.

XGBoost là một thuật toán máy học theo phương pháp học tập tổng hợp. Nó là xu hướng cho các nhiệm vụ học tập có giám sát, chẳng hạn như hồi quy và phân loại. XGBoost xây dựng mô hình dự đoán bằng cách kết hợp các dự đoán của nhiều mô hình riêng lẻ, thường là cây quyết định.

XGBoost tối ưu hàm mục tiêu tại lần lặp t là:

$$\text{Obj}(t) = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (3)$$

Trong đó:

- $l$  là hàm mất mát (loss function, ví dụ như MSE cho regression, logistic loss cho classification).
- $y_i$  là giá trị thực tế của quan sát thứ  $i$ .
- $\hat{y}_i^{(t-1)}$  là dự đoán tại vòng lặp thứ  $t - 1$ .
- $f_t$  là cây quyết định được thêm vào tại vòng lặp thứ  $t$ .
- $\Omega(f_t)$  là hàm phạt (regularization term) giúp ngăn ngừa overfitting.

Các lợi ích và đặc điểm của mô hình XGBoost:

- Độ chính xác cao: Bộ phân loại XGBoost nổi tiếng với độ chính xác cao và đã được chứng minh vượt trội hơn so với các thuật toán máy học khác trong nhiều nhiệm vụ dự đoán.
- Khả năng mở rộng: XGBoost có khả năng mở rộng cao và có thể xử lý các bộ dữ liệu lớn với hàng triệu hàng và cột.
- Hiệu quả: Nó được thiết kế để tính toán hiệu quả và có thể nhanh chóng huấn luyện các mô hình trên các bộ dữ liệu lớn.
- Linh hoạt: XGBoost hỗ trợ nhiều loại dữ liệu và mục tiêu khác nhau, bao gồm hồi quy, phân loại và các vấn đề xếp hạng.
- Chính quy hóa: Nó tích hợp các kỹ thuật chính quy để tránh hiện tượng quá khớp và cải thiện hiệu suất tổng quát.

## 4.7 THUẬT TOÁN N-BEAT

Thuật toán N-BEATS là một mô hình dự đoán chuỗi thời gian mạnh mẽ dựa trên một cấu trúc kiến trúc mạng nơ-ron đa tầng sâu. N-BEATS được thiết kế để có khả năng học được các biến đổi không tuyến tính và phức tạp trong dữ liệu chuỗi thời gian. Điểm nổi bật của N-BEATS là khả năng mở rộng và linh hoạt trong việc tùy chỉnh cấu trúc mạng.

Các đặc điểm chính:

+ Cấu trúc mô hình : Mô hình N-BEATS thường bao gồm một hoặc nhiều khối có thể lặp lại. Mỗi khối bao gồm hai mạng nơ-ron: một mạng BackcastNet và một mạng ForecastNet. Cả hai mạng này thường có cấu trúc tương tự như Feedforward Neural Networks hoặc Convolutional Neural Networks.

+ Hàm mất mát : N-BEATS thường sử dụng hàm mất mát như Mean Absolute Error (MAE) hoặc Mean Squared Error (MSE) giữa dự đoán và giá trị thực tế của chuỗi thời gian.

+ Quá trình huấn luyện : Mô hình N-BEATS được huấn luyện thông qua việc tối ưu hóa hàm mất mát bằng các phương pháp tối ưu hóa như gradient descent hoặc các biến thể của nó, như Adam hoặc RMSprop.

+ Backcast và Forecast:

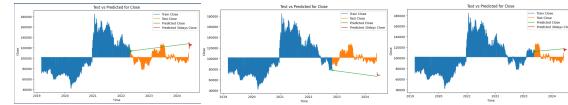
Backcast : Quá trình dự đoán các giá trị trong quá khứ của chuỗi thời gian. Trong quá trình này, mô hình sẽ sử dụng các giá trị quan sát được trong quá khứ để dự đoán giá trị tại một thời điểm cụ thể trong quá khứ. Quá trình này giúp mô hình học được cách biến đổi và ảnh hưởng của dữ liệu đầu vào đối với các giá trị trong quá khứ, từ đó cung cấp thông tin cần thiết để dự đoán tương lai.

Forecast: Quá trình dự đoán các giá trị trong tương lai của chuỗi thời gian. Trong quá trình này, mô hình sẽ sử dụng các giá trị quan sát được trong hiện tại và quá khứ để dự đoán giá trị tại một thời điểm cụ thể trong tương lai. Quá trình này cho phép mô hình học được cách dữ liệu hiện tại và quá khứ ảnh hưởng đến các giá trị trong tương lai, từ đó cung cấp dự đoán cho các giá trị trong tương lai của chuỗi thời gian

## 5 THỰC NGHIỆM

### 5.1 MÔ HÌNH DỰ ĐOÁN

#### 5.1.1 VARMA



Hình 8: LG 30 DAYS 6:4, 7:3, 8:2



Hình 9: LG 60 DAYS 6:4, 7:3, 8:2



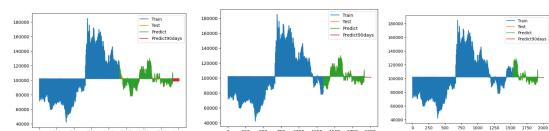
Hình 10: LG 90 DAYS 6:4, 7:3, 8:2



Hình 11: SONY 30 DAYS 6:4, 7:3, 8:2



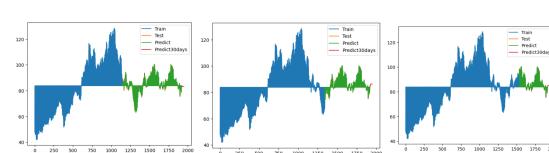
Hình 12: SONY 60 DAYS 6:4, 7:3, 8:2



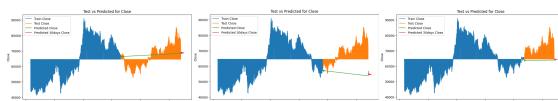
Hình 19: LG 90 DAYS 6:4, 7:3, 8:2



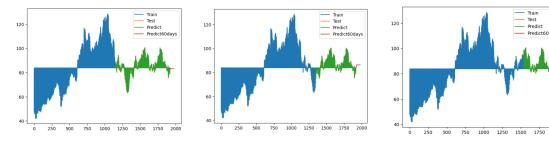
Hình 13: SONY 90 DAYS 6:4, 7:3, 8:2



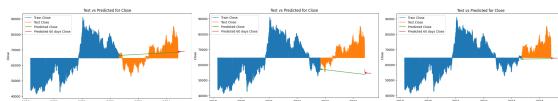
Hình 20: SONY 30 DAYS 6:4, 7:3, 8:2



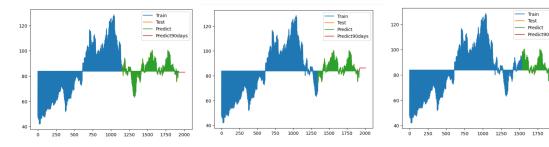
Hình 14: SAMSUNG 30 DAYS 6:4, 7:3, 8:2



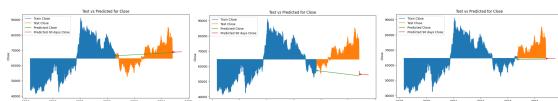
Hình 21: SONY 60 DAYS 6:4, 7:3, 8:2



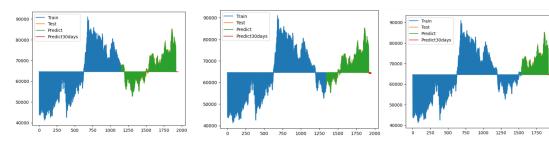
Hình 15: SAMSUNG 60 DAYS 6:4, 7:3, 8:2



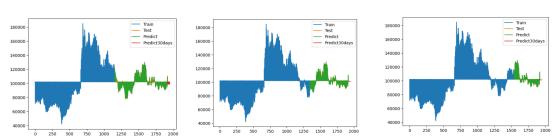
Hình 22: SONY 90 DAYS 6:4, 7:3, 8:2



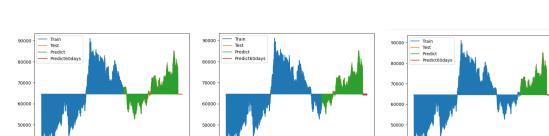
Hình 16: SAMSUNG 90 DAYS 6:4, 7:3, 8:2



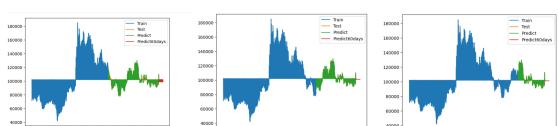
Hình 23: SAMSUNG 30 DAYS 6:4, 7:3, 8:2



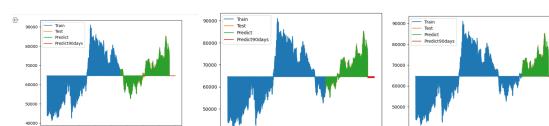
Hình 17: LG 30 DAYS 6:4, 7:3, 8:2



Hình 24: SAMSUNG 60 DAYS 6:4, 7:3, 8:2

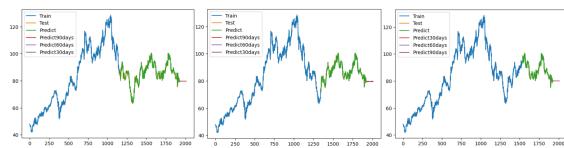


Hình 18: LG 60 DAYS 6:4, 7:3, 8:2



Hình 25: SAMSUNG 90 DAYS 6:4, 7:3, 8:2

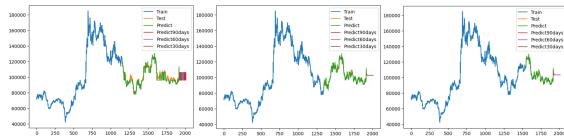
### 5.1.3 XGBOOST



Hinh 26: XGBoost model's result with SONY dataset

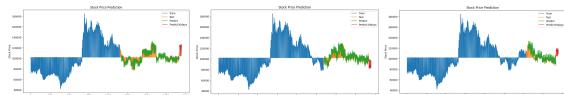


Hinh 27: XGBoost model's result with SAMSUNG dataset

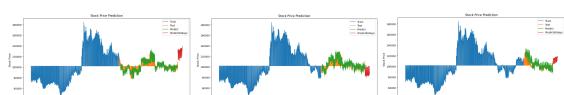


Hinh 28: XGBoost model's result with LG dataset

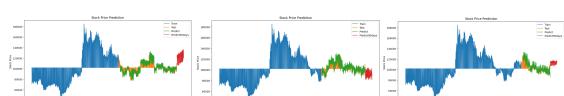
### 5.1.4 N-BEAT



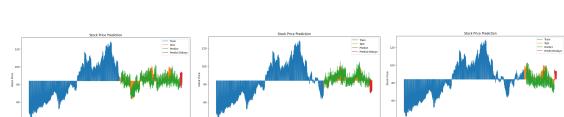
Hinh 29: LG 30 DAYS 6:4, 7:3, 8:2



Hinh 30: LG 60 DAYS 6:4, 7:3, 8:2



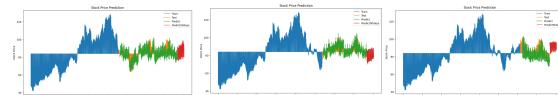
Hinh 31: LG 90 DAYS 6:4, 7:3, 8:2



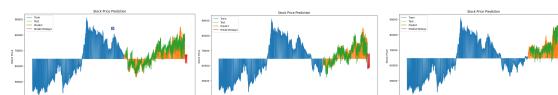
Hinh 32: SONY 30 DAYS 6:4, 7:3, 8:2



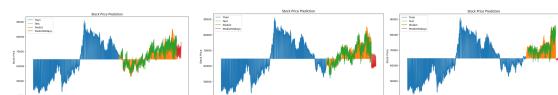
Hinh 33: SONY 60 DAYS 6:4, 7:3, 8:2



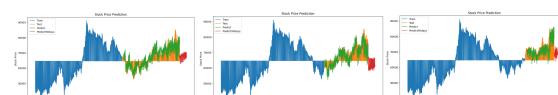
Hinh 34: SONY 90 DAYS 6:4, 7:3, 8:2



Hinh 35: SAMSUNG 30 DAYS 6:4, 7:3, 8:2

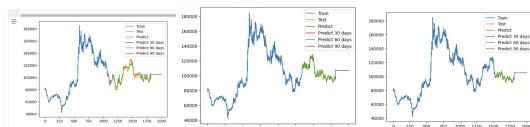


Hinh 36: SAMSUNG 60 DAYS 6:4, 7:3, 8:2

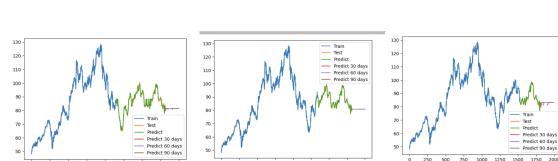


Hinh 37: SAMSUNG 90 DAYS 6:4, 7:3, 8:2

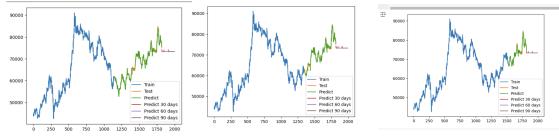
### 5.1.5 RNN + LIGHT GBM



Hinh 38: LG dataset 6:4, 7:3, 8:2



Hinh 39: Sony dataset 6:4, 7:3, 8:2



Hình 40: SAMSUNG dataset 6:4, 7:3, 8:2

## 5.2 ĐÁNH GIÁ VÀ KẾT QUẢ

### 5.2.1 MÔ HÌNH ĐÁNH GIÁ VỚI TẬP DỮ LIỆU LG

| Model             | Train test 6:4 |                    |          | Train test 7:3 |                   |           | Train test 8:2 |                    |          |
|-------------------|----------------|--------------------|----------|----------------|-------------------|-----------|----------------|--------------------|----------|
|                   | RMSE           | MSE                | MAPE     | RMSE           | MSE               | MAPE      | RMSE           | MSE                | MAPE     |
| VARMA             | 1.02015, 3468  | 10419390259, 93193 | 99,9900% | 103930, 0102   | 10780671013, 1514 | 100,0000% | 104018, 4736   | 10181946265, 13071 | 99,9900% |
| XGBoost           | 12134, 0139    | 147234293, 41      | 14,0657% | 138076, 0794   | 139567793, 78     | 9,6       | 1701, 5222     | 2095177, 72        | 1,7378%  |
| Gradient Boosting | 13331, 8321    | 1737376, 7044      | 0,8520%  | 1121, 5964     | 1287979, 1443     | 0,7321%   | 945, 4851      | 631394461, 392     | 0,1912%  |
| NBEats            | 8166, 6858     | 66694757, 100      | 6,1700%  | 8229, 2707     | 67720896, 888     | 6,2600%   | 7335, 1427     | 613959461, 762     | 5,0000%  |
| RNN LightGBM      | 0,0532         | 0,0028             | 10,9619% | 0,0370         | 0,0014            | 6,3418%   | 0,0241         | 0,0006             | 5,0364%  |

### 5.2.2 MÔ HÌNH ĐÁNH GIÁ VỚI TẬP DỮ LIỆU SONY

| Model             | Train test 6:4 |           |          | Train test 7:3 |           |          | Train test 8:2 |           |          |
|-------------------|----------------|-----------|----------|----------------|-----------|----------|----------------|-----------|----------|
|                   | RMSE           | MSE       | MAPE     | RMSE           | MSE       | MAPE     | RMSE           | MSE       | MAPE     |
| VARMA             | 85,2537        | 7269,1967 | 99,9600% | 86,4895        | 7480,4372 | 99,9700% | 87,1740        | 7599,3068 | 99,9700% |
| XGBoost           | 7,4301         | 55,2060   | 8,6995%  | 8,5473         | 73,0568   | 11,4127% | 4,1702         | 17,3905   | 5,2628%  |
| Gradient Boosting | 0,5131         | 0,2632    | 0,3878%  | 0,4100         | 0,1681    | 0,3100%  | 0,3742         | 0,1400    | 0,2870%  |
| NBEats            | 7,4235         | 55,1086   | 6,7500%  | 6,1880         | 38,2908   | 5,4300%  | 6,6931         | 75,5697   | 7,8400%  |
| RNN LightGBM      | 0,0518         | 0,0027    | 9,0061%  | 0,0266         | 0,0007    | 3,9340%  | 0,0252         | 0,0006    | 3,9372%  |

### 5.2.3 MÔ HÌNH ĐÁNH GIÁ VỚI TẬP DỮ LIỆU SAMSUNG

| Model             | Train test 6:4 |                |           | Train test 7:3 |                |           | Train test 8:2 |                |           |
|-------------------|----------------|----------------|-----------|----------------|----------------|-----------|----------------|----------------|-----------|
|                   | RMSE           | MSE            | MAPE      | RMSE           | MSE            | MAPE      | RMSE           | MSE            | MAPE      |
| VARMA             | 66591, 4033    | 664952286, 423 | 100,0000% | 60285, 3722    | 4662892058, 22 | 100,0000% | 71049, 4794    | 5046028524, 43 | 100,0000% |
| XGBoost           | 4463, 7737     | 39292775, 32   | 6,3453%   | 5319, 1986     | 20293973, 94   | 0,5871%   | 1807, 9566     | 2208706, 52    | 0,7042%   |
| Gradient Boosting | 447, 2897      | 20069, 0372    | 0,4534%   | 439, 7989      | 193544, 4837   | 0,4303%   | 449, 3986      | 203959, 071    | 0,4114%   |
| NBEats            | 4520, 1126     | 23407333, 32   | 5,4100%   | 5263, 7966     | 27707555, 14   | 5,7500%   | 5764, 7135     | 33231921, 63   | 5,7500%   |
| RNN LightGBM      | 0,0758         | 0,0057         | 11,5631%  | 0,0776         | 0,0095         | 12,9990%  | 0,0374         | 0,0014         | 4,1780%   |

## 6 KẾT LUẬN

### 6.1 KẾT LUẬN TỔNG THỂ

Dự đoán giá cổ phiếu vẫn là một nhiệm vụ phức tạp và đầy thách thức, và việc tìm ra một mô hình chính xác toàn diện vẫn còn khó khăn. Nghiên cứu này đã khám phá các phương pháp thống kê, học máy, và học sâu để dự đoán giá cổ phiếu của các gã khổng lồ công nghệ hàng đầu: LG, Samsung và Sony. Mỗi phương pháp mang lại những hiểu biết và lợi thế độc đáo, nhấn mạnh tầm quan trọng của việc xem xét các phương pháp đa dạng cho phân tích toàn diện.

Nhìn chung, nghiên cứu này chứng minh rằng mặc dù không có mô hình nào có thể dự đoán chính xác giá cổ phiếu, việc sử dụng kết hợp các phương pháp thống kê, học máy và học sâu có thể cung cấp những

hiểu biết quý báu và cải thiện độ chính xác của dự báo. Tuy nhiên, điều quan trọng là phải xem xét các hạn chế của từng phương pháp và diễn giải kết quả một cách thận trọng, thừa nhận sự bất định vốn có liên quan đến dự đoán thị trường chứng khoán.

## 6.2 THÁCH THỨC

Dữ liệu là yếu tố quan trọng nhất trong phân tích thống kê và dự đoán. Một tập dữ liệu phù hợp cần phải đáp ứng các yêu cầu sau: Dài đủ và chính xác cao, đủ lớn để mô hình có thể học và dự đoán chính xác, liên quan chặt chẽ đến đối tượng cần dự đoán. Tuy nhiên, việc tìm kiếm một tập dữ liệu phù hợp thường không dễ dàng. Dữ liệu công khai thường không đáp ứng được các yêu cầu trên hoặc không phù hợp với chủ đề của dự án.

Sau khi thu thập dữ liệu, tôi bắt đầu mã hóa các dự đoán bằng các mô hình. Tôi đã chọn một số mô hình phổ biến như Gradient Boosting, VARMA, XGBoost, và mô hình Deep Learning như NBEATS, Light GBM kết hợp RNN.

Tuy nhiên, việc xử lý các mô hình này không đơn giản. Nhóm cần phải có kiến thức về lý thuyết thống kê, lý thuyết học máy và kỹ năng lập trình. Chúng em đã dành nhiều thời gian nghiên cứu cách các mô hình này hoạt động.

## 6.3 KẾ HOẠCH TƯỞNG LAI

Để nâng cao độ chính xác của dự đoán, chúng em dự định sẽ:

Tinh chỉnh dữ liệu: Nâng cao kỹ năng thu thập và xử lý dữ liệu thông qua các phương pháp mới nhất, cung cấp cho các mô hình của chúng tôi thông tin sạch hơn và đáng tin cậy hơn. Nâng cấp thuật toán: Khám phá các phương pháp tiên tiến như Học sâu (Deep Learning) và Học tăng cường (Reinforcement Learning) để có các mô hình dự đoán tinh vi và chính xác hơn. Dánh giá thông minh hơn: Áp dụng các chỉ số tiêu chuẩn trong ngành như MASE, MAPE và SMAPE để đánh giá toàn diện hiệu quả của các mô hình. Hợp tác: Tích cực hợp tác với các chuyên gia trong các cộng đồng chuyên môn để chia sẻ kinh nghiệm và mở rộng nền tảng kiến thức của chúng tôi.

## 7 NHÌN NHẬN

## 8 TÀI LIỆU THAM KHẢO

[1] : Mojtaba Nabipour; Pooyan Nayyeri; Hamed Jabani; Shahab S.; Amir Mosavi , "Predicting Stock

Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis", 2020.

[2]: Salah Bouktif; Ali Fiaz; Mamoun Awad , "Augmented Textual Features-Based Stock Market Prediction", 2020

[3] : Sondo Kim; Seungmo Ku; Woojin Chang; Jae Wook Song , "Augmented Textual Features-Based Stock Market Prediction", 2020

[4]: Warsono, Edwin Russel, Wamiliana\*, Widiarti, Mustofa Usman, "Modeling and Forecasting by the Vector Autoregressive Moving Average

Model for Export of Coal and Oil Data (Case Study from Indonesia over the Years 2002-2017)". International Journal of Energy Economics and Policy, 2019

[5]: Yugesh Verma, "A Guide to VARMA with Auto ARIMA in Time Series Modelling". analytic-sindiamag.com, 2021

[6]: Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree", 31st Conference on Neural Information Processing Systems (NIPS 2017).