ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÀI TẬP LỚN
Khai phá dữ liệu (055131)

ĐỀ TÀI:
Hệ thống dự đoán giá chứng khoán

GVHD: Trần Minh Quang
Lê Hồng Trang
Lớp: 1
Nhóm: 4
1913621 – Bùi Đắc Hưng
1911314 – Lương Thị Quỳnh Hương
2170535 – Lê Vũ Minh Huy

Tp. Hồ Chí Minh, Tháng 11/2022

Mục lục

A. Giới thiệu đề tài

Ngày nay ngành công nghệ thông tin đang không ngừng phát triển, đặc biệt là nhờ mạng Internet mà mọi người có thể tiếp cận được vô vàn thông tin bất kể khoảng cách không gian và vị trí địa lý. Tuy nhiên không phải tất cả các thông tin mà mỗi người tiếp xúc đều có ích cho chính họ. Vì vậy mà ta cần phải tổng hợp và phân tích dữ liệu để phục vụ cho từng mục đích riêng. Trên thực tế, những dữ liệu khổng lồ thu thập được nếu được phân loại và khai thác đúng cách sẽ mang lại được những ý nghĩa to lớn. Đó là lý do mà việc khai phá dữ liệu đã và đang không ngừng phát triển, tận dụng lượng dữ liệu đang có để ứng dụng vào nhiều vấn đề thực tiễn ở mọi lĩnh vực trong đời sống xã hội. Một ứng dụng quan trọng trong khai phá dữ liệu là dự báo, được áp dụng rộng rãi trong nhiều ngành nghề.

Đầu tư chứng khoán hiện nay đang là một ngành kinh tế rất hot. Những người chơi chứng khoán phải đoán trước được xu hướng thay đổi của thị trường thì mới thu được lợi nhuận tối đa. Tuy nhiên điều này không là không dễ vì thị trường chứng khoán luôn biến động và thay đổi từng ngày. Vì vậy cần phải nghiên cứu một công cụ có thể giúp cho việc dự đoán giá chứng khoán. Nhờ sự phát triển của khai phá dữ liệu thì điều tưởng chừng như không thể lại có thể thực hiện được.

Trong bài tập lớn này, nhóm sẽ thực hiện áp dụng một số mô hình dự đoán để xây dựng hệ thống dự đoán giá chứng khoán. Từ đó sẽ phân tích và đánh giá từng mô hình cũng như so sánh giữa các mô hình dự đoán khác nhau. Dữ liệu để huấn luyện cũng như kiểm tra mô hình sẽ lấy từ dữ liệu chứng khoán của một số công ty ở Việt Nam.

B. Chuẩn bị dữ liệu

I. Thu thập dữ liệu

Nguồn dữ liệu được thu thập bằng cách sử dụng API có sẵn của 1 github https://github.com/phamdinhkhanh/vnquant được 276 stars. Nhóm sử dụng API để thu thập dữ liệu của 5 loại cổ phiếu từ năm 01012015 đến 01112022 từ sàn giao dịch VNDirect:

- **BVH**: Tập đoàn Bảo Việt
- **BID**: Ngân hàng thương mại cổ phần đâu tư và phát triển Việt Nam
- **CTG**: Ngân hàng thương mại cổ phần công thương Việt Nam
- **ACB**: Ngân hàng Thương Mại cổ phần Á Châu
- **FPT**: Công ty cổ phần FPT

Dữ liệu đã được nhóm kiểm tra lại bằng cách so với giá trị trên https://banggia.vndirect.com.vn/chung-khoan/danh-muc
Mỗi bảng dữ liệu gồm 1955 dòng dữ liệu và 5 cột:

- **date**: ngày có thông tin về giá cổ phiếu
- **low**: giá trị thấp nhất của cổ phiếu trong ngày
- **high**: giá trị cao nhất của cổ phiếu trong ngày
- **close**: giá cổ phiếu lúc đóng cửa
- **open**: giá cổ phiếu lúc mở cửa

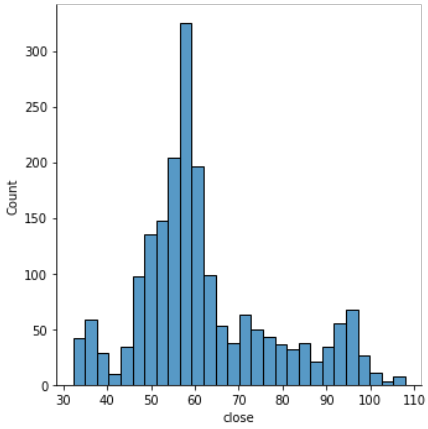
Trong bài tập lớn này, nhóm dự đoán giá trị cổ phiếu lúc đóng cửa, tức cột **close**.

II. Phân tích dữ liệu

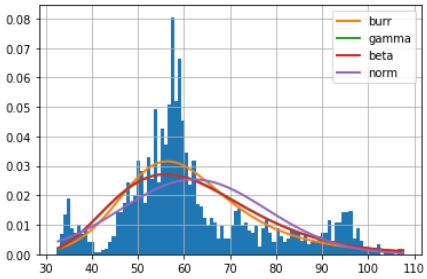
1. BVH: Tập đoàn Bảo Việt

Biểu đồ giá cổ phiếu lúc đóng của của tập đoàn Bảo Việt được minh họa ở Hình 1. Cổ phiếu của công ty BVH tăng nhanh ở cuối năm 2017 đến tháng 3-2018. Sau đó, có dấu hiệu đi xuống 3 tháng sau rồi tăng lên lại đến cuối năm 2018. Tuy nhiên, từ năm 2019 đến tháng 3-2020, cổ phiếu tập đoàn Bảo Việt BVH giảm mạnh. Từ tháng 3-2020 đến nay, nhìn chung cổ phiểu BVH ổn định.
Phân bố dữ liệu BVH được mô tả ở Hình 2. Phân bố dữ liệu lệch về phía bên trái, tạo nên sự không cân đối. Có thể đoán rằng vào giai đoạn 2018-2019 tập đoàn Bảo Việt đã có sự kiện gì đó khiến cổ phiếu tăng vọt.

Hình 1: *Biểu đồ giá cổ phiếu lúc đóng của tập đoàn Bảo Việt*



(a) *Biểu đồ Histogram của giá cổ phiếu lúc đóng cửa BVH*



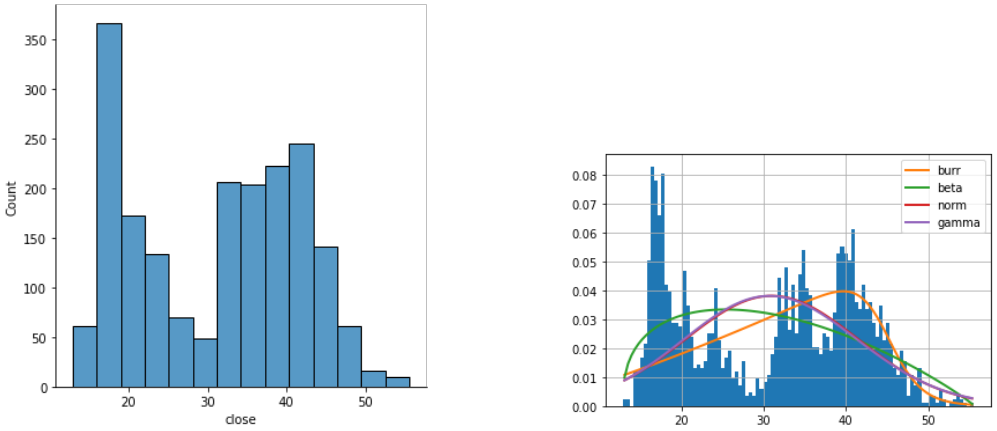
(b) *Tìm phân bố của dữ liệu (phân bố burr là tốt nhất)*

Hình 2: *Phân bố dữ liệu BVH*

Hình 3: *Biểu đồ giá cổ phiếu lúc đóng BID*

2. BID: Ngân hàng thương mại cổ phần đầu tư và phát triển Việt Nam

Giá trị cổ phiếu lúc đóng cửa của BID được minh họa ở Hình 3. Từ năm 2017, giá trị cổ phiếu BID tăng và đạt đỉnh vào tháng 4 năm 2018. Sau đó giá trị cổ phiếu đã giảm



(a) *Biểu đồ Histogram của giá cổ phiếu lúc đóng cửa BID*

(b) *Tìm phân bố của dữ liệu (phân bố burr là tốt nhất)*

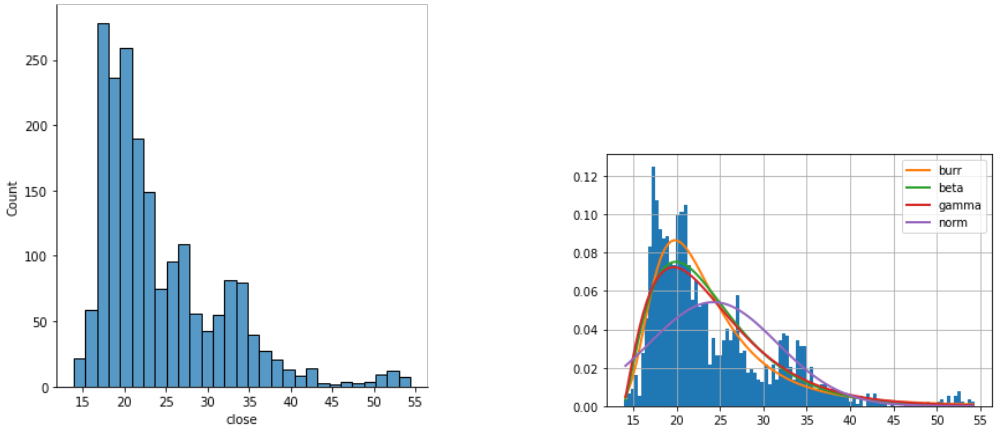
Hình 4: *Phân bố dữ liệu BID*

xuống 2 tháng sau đó. Tuy nhiên, từ tháng 6-2018 giá trị cổ phiếu này có xu hướng tăng và đã đạt đỉnh vào tháng 2 năm 2020. Sau thời gian này, nhìn chung giá trị cổ phiếu BID ổn định.
Phân bố dữ liệu BID ở Hình 4 là phân bố hai đỉnh. Có thể thấy giá trị cổ phiếu lúc đóng cửa trước năm 2018 và sau năm 2018 tuân theo 2 phân bố khác nhau rõ rệt.

3. CTG: Ngân hàng thương mại cổ phần công thương Việt Nam

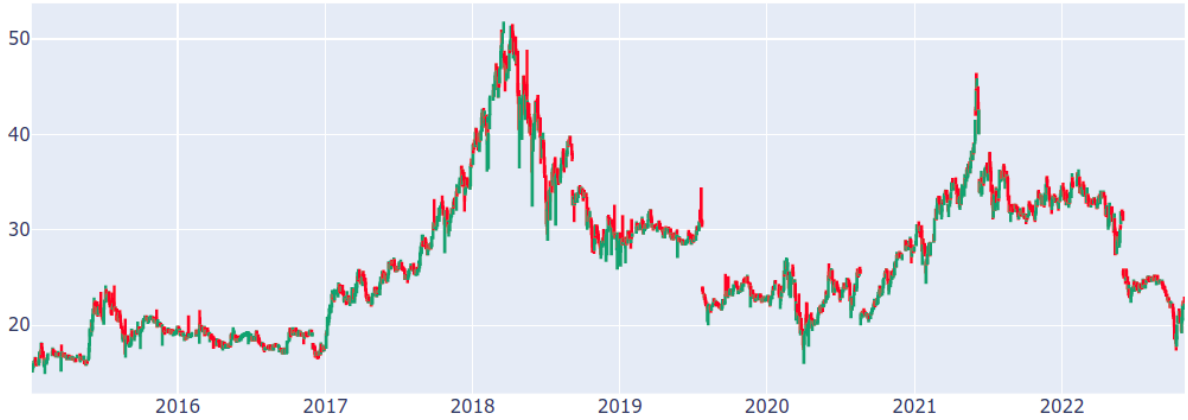
Hình 5: *Biểu đồ giá cổ phiếu lúc đóng CTG*

Dữ liệu giá cổ phiếu lúc đóng cửa CTG được minh họa ở Hình 5. Từ năm 2015 đến cuối năm 2017, giá trị cổ phiếu CTG ổn định. Từ cuối năm 2017 đến tháng 3 năm 2018, giá trị cổ phiếu CTG có xu hướng tăng và đạt đỉnh vào tháng 3-2018. Giá trị cổ phiếu CTG có xu hướng tăng từ tháng 3-2020 và đạt đỉnh lần 2 vào tháng 6-2021. Sau đó, giá trị cổ phiếu CTG cho đến nay đang có xu hướng giảm.
Phân bố dữ liệu của CTG ở Hình 6 là phân bố bị lệch về phía bên trái.



(a) *Biểu đồ Histogram của giá cổ phiếu lúc đóng cửa CTG* **(b)** *Tìm phân bố của dữ liệu (phân bố burr là tốt nhất)*

Hình 6: *Phân bố dữ liệu CTG*



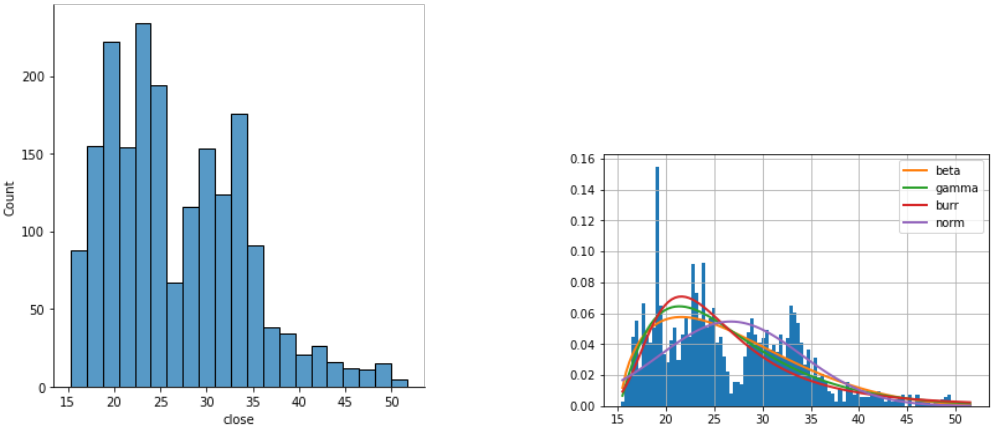
Hình 7: *Biểu đồ giá cổ phiếu lúc đóng ACB*

4. ACB: Ngân hàng Thương Mại cổ phần Á Châu

Giá trị cổ phiếu lúc đóng cửa ACB được minh họa ở Hình 7. Giá trị cổ phiếu của ACB tăng từ đầu năm 2017 và đạt đỉnh vào tháng 3-2018. Sau đó có xu hướng giảm đến tháng 3-2020. Từ tháng 3-2020, giá trị cổ phiếu ACB tăng và đạt đỉnh vào tháng 6-2021. Từ đó đến nay, giá trị cổ phiếu ACB có xu hướng giảm.
Phân bố dữ liệu ACB được mô tả ở Hình 8 là phân bố hai đỉnh. Tuy nhiên nhìn biểu đồ giá trị cổ phiếu ACB bằng mắt thường không phân biệt rõ rệt 2 giai đoạn giống như cổ phiếu BID.

5. FPT: Công ty cổ phần FPT

Giá trị cổ phiếu lúc đóng cửa ACB được minh họa ở Hình 9. Giá trị cổ phiếu FPT nhìn chung từ năm 2015 đến nửa đầu năm 2020 ổn định. Tuy nhiên từ nửa sau năm 2020 đến tháng 4 năm 2022, giá trị cổ phiếu tăng mạnh từ xấp xỉ 45 lên xấp xỉ 120, tức gấp 2.5 lần. Sau đó đến nay, cổ phiếu FPT đang có xu hướng giảm.
Phân bố dữ liệu FPT được mô tả ở Hình 10 là phân bố 2 phía. Nhìn bằng mắt thường ta có thể thấy trước năm 2021 và sau năm 2021, cổ phiếu FPT có 2 phân bố phân biệt.

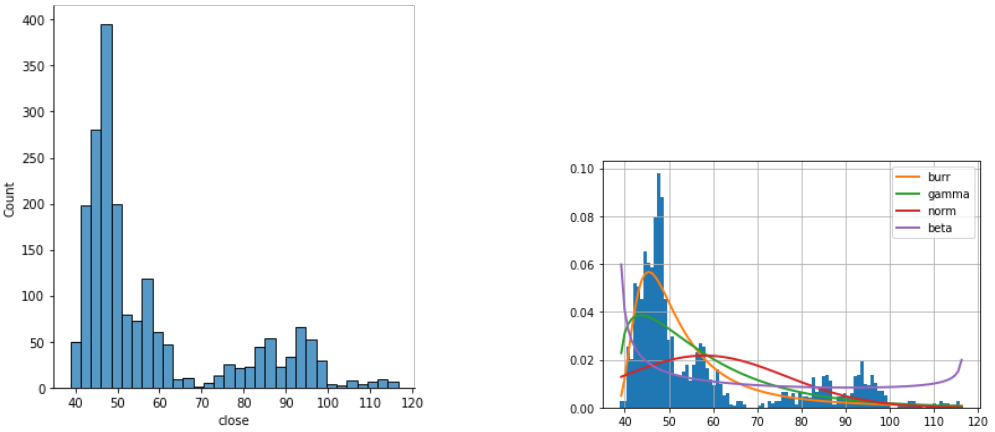


(a) *Biểu đồ Histogram của giá cổ phiếu lúc đóng cửa ACB*

(b) *Tìm phân bố của dữ liệu (phân bố burr là tốt nhất)*

Hình 8: *Phân bố dữ liệu ACB*

Hình 9: *Biểu đồ giá cổ phiếu lúc đóng FPT*



(a) *Biểu đồ Histogram của giá cổ phiếu lúc đóng cửa FPT*

(b) *Tìm phân bố của dữ liệu (phân bố burr là tốt nhất)*

Hình 10: *Phân bố dữ liệu FPT*

III. Chuẩn hóa dữ liệu

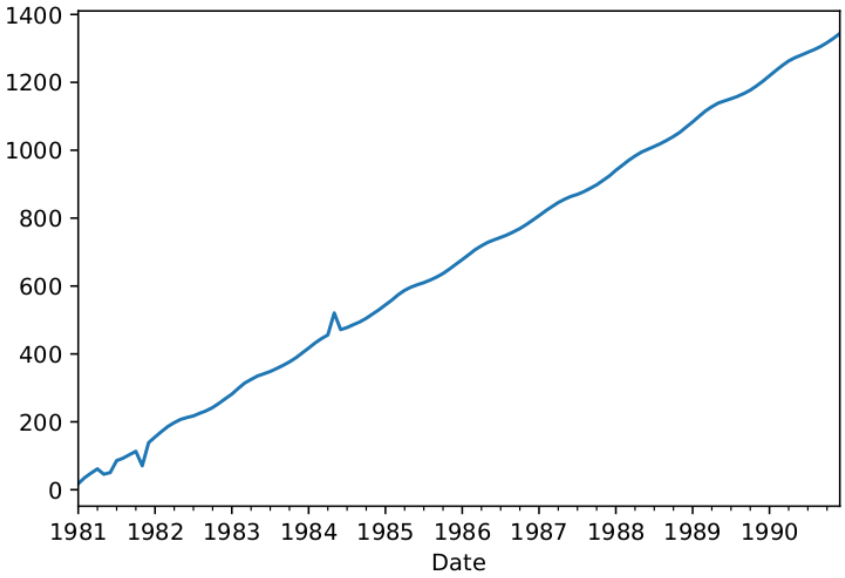
Ở đây, nhóm sử dụng Z-score để chuẩn hóa dữ liệu.

$$z = \frac{x - \tilde{x}}{\sigma}$$

trong đó, $\tilde{x}$: trung bình, σ: độ lệch chuẩn.

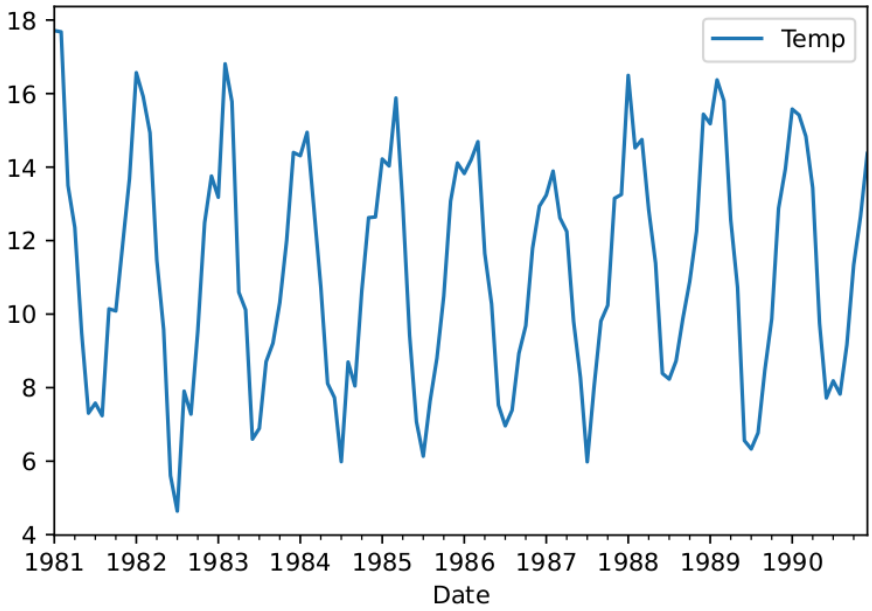
IV. Một số đặc điểm về time series

Time series có một vài đặc điểm để chỉ ra các hành vi và pattern của dữ liệu. Bằng cách xác định các đặc điểm này, ta có thể dự đoán tốt hơn.
Xu hướng (trend). Giá trị có khuynh hướng tăng hoặc giảm dần theo thời gian. Ví dụ ở Hình 11 cho ta thấy dữ liệu có khuynh hướng tăng dần theo thời gian.



Hình 11: *Ví dụ về khuynh hướng tăng của dữ liệu theo thời gian.*

Tính chu kỳ theo mùa (seasonal cycles). Dữ liệu có khuynh hướng lặp lại theo chu kỳ. Điều này là do dữ liệu bị ảnh hưởng bởi các yếu tố theo mùa (theo tháng hoặc theo năm, ...). Một ví dụ được mô tả ở Hình 12.
Tính ổn định (Stationary). Một giả thuyết thường xuyên xuất hiện trong các mô hình dự đoán như ARIMA là dữ liệu phải có tính ổn định. Một quá trình ổn định là một quá trình ngẫu nhiên trong đó tính chất thống kê không thay đổi theo thời gian. Nói cách khác, trung bình, phương sai, sự tương quan là hằng số. Vì thế, một time series ổn định sẽ không có xu hướng hay tính chu kỳ theo mùa.



Hình 12: *Ví dụ tính chu kỳ theo mùa của dữ liệu*

C. Các mô hình dự đoán

Tất cả source code của nhóm thực hiện ở link github sau: https://github.com/bdachung/stock-prediction

I. Mô hình One-Step Ahead thông qua việc trung bình giá

Cơ chế trung bình giá cho phép chung ta có thể dự đoán (thường là chỉ dự đoán trước một bước - one-step ahead) những biểu hiện của các cổ phiếu trong tương lai, dựa trên trung bình giá cổ phiếu đã quan sát được trong quá khứ.

Nhóm sẽ trình bày hai cơ chế trung bình: Trung bình giá cơ bản (Standard averaging) và Trung bình giá lũy thừa (Exponential moving average)

Đồng thời nhóm sẽ chọn kết quả của phương pháp trung bình giá lũy thừa làm baseline cho các phương pháp sau này.

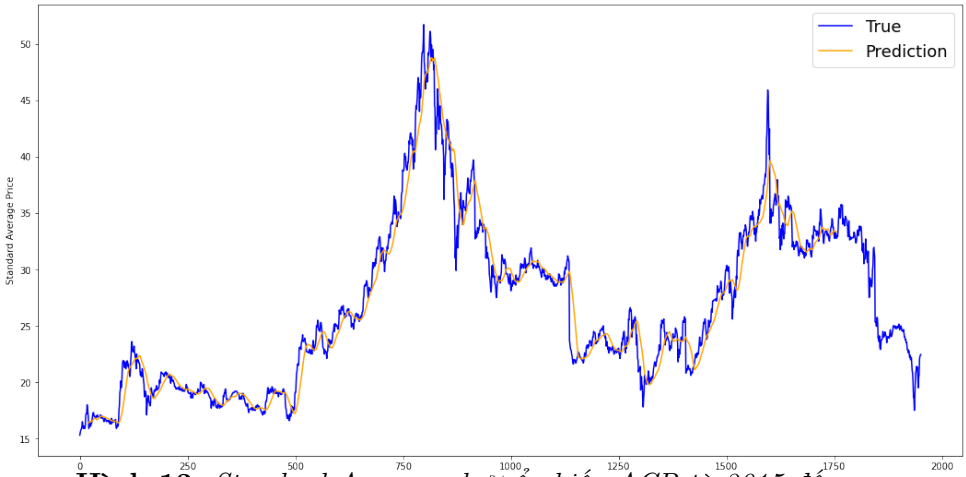
1. Trung bình giá cơ bản - Standard Average

Một cách dễ hiểu, giá cổ phiếu ở ngày thứ n trong tương lai sẽ bằng trung bình cộng giá của K ngày trong quá khứ.

K là một sô ta tự chọn (gọi là window size). Ví dụ chúng ta quy định: giá cổ phiếu ngày mai là trung bình cộng giá của $K = 30$ ngày trước đó.

$$x_n = \frac{1}{K} * \sum_{i=n-1-K}^{n} x_i$$

Liệu cách tính đơn giản như vậy có cho kết quả tốt đối với một thị trường phức tạp và khó dự đoán như chứng khoán?



Hình 13: *Standard Average cho cổ phiếu ACB từ 2015 đến nay*

Và kết quả khá bất ngờ, với K = 20, tức ta lấy trung bình cộng của 20 ngày trước làm kết quả cho ngày tiếp theo, sẽ được đồ thị như bên trên.

Có hai điểm lưu ý với mô hình này:

- Mô hình đơn giản nhưng cho kết quả tốt vì nó dự đoán một thời gian rất ngắn trong tương lai - chỉ 1 ngày tiếp theo. Và vì giá cổ phiếu sẽ không thường biến động cực lớn chỉ trong 1 ngày (ví dụ từ 100 về 0, hoặc ngược lại), nên giá của ngày tiếp theo được tính như trên là hợp lí
- Nó rất khó để dự đoán nhiều hơn một ngày

2. Trung bình giá lũy thừa - Exponential Moving Average

Bây giờ chúng ta sẽ áp dụng một phương pháp phức tạp hơn công thức trung bình cộng đơn giản kia.

Nhưng trước khi đi vào phương pháp này, nhóm muốn nhấn mạnh hai điều:

- Chứng khoán là một mảng phức tạp và rất khó để dự đoán chính xác
- Có một số phương pháp dường như dự đoán cực kì chính xác từng biến động của thị trường chứng khoán, nhưng lại không hiệu quả khi giao dịch thật sự

Đây chính là một phương pháp như vậy

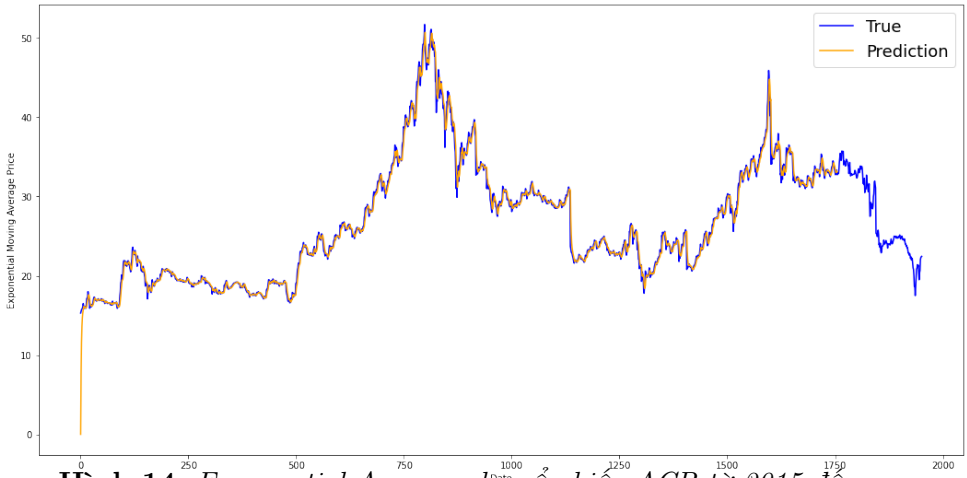
Giá cổ phiếu ngày thứ n trong tương lai được tính như sau:

$$x_n = \mathbf{EMA}_{n-1} = \gamma * \mathbf{EMA}_{n-2} + (1-\gamma)x_{n-1}$$

$$\mathbf{EMA}_0 = 0$$

Đúng với tên gọi của nó, mô hình lũy thừa vì có công thức đệ quy với trung bình giá **EMA**. Bản chất của mô hình này cho phép chúng ta quan sát và lưu trữ sâu hơn về quá khứ.

Kết quả của mô hình tốt đến ngạc nhiên, đồ thị dự đoan dường như trùng với phân phối thực sự của cổ phiếu này



Hình 14: *Exponential Average chờ cổ phiếu ACB từ 2015 đến nay*

Hình 15: *Exponential Average chờ cổ phiếu ACB từ 2015 đến nay*

Phương pháp này rất nổi tiếng trong giới trader vì nó vừa đơn giản mà kết quả lại tốt đến ngạc nhiên.

Vậy nếu nó đã tốt đến như thế, vì sao chúng ta lại cần những phương pháp phức tạp khác?

Vì mô hình này chỉ dự đoán tốt với đúng 1 ngày tiếp theo, càng cố dự đoán xa hơn, kết quả sẽ càng tệ. Do đó nó không phù hợp khi ta cần một chiến lược đầu tư dài hạn.

Dù vậy, vì đây là một kết quả rất tốt nên nhóm sẽ sử dụng chúng làm baseline cho những phương pháp tiếp theo.

Dựa trên 3 độ đo chính gồm: **MSE** (Mean squared error), **MAPE** (Mean absolute percentage error), **MAE** (Mean Absolute Error), ta có bảng baseline sau:

	Baseline		
Stocks	MSE	MAPE	MAE
ACB	0.6615	0.0191	0.4985
BID	1.1850	0.0219	0.8046
BVH	2.7125	0.0211	1.1681
CTG	0.6313	0.0225	0.5996
FPT	9.3772	0.0186	1.7228
	2.9135	**0.0206**	**0.9587**

II. Mô hình Autoregressive Integrated Moving Average (ARIMA)

Mô hình ARIMA là một trong các mô hình thống kê cho việc dự đoán dữ liệu time series bằng cách sử dụng các giá trị quá khứ. ARIMA gồm có 3 thành phần chính: Autoregressive (AR), Moving Average (MA), and Integrated (I).

1. Autoregressive (AR)

Mô hình autoregressive (AR) là một mô hình hồi quy trong đó giá trị quan sát tại một thời điểm là một tổ hợp tuyến tính các giá trị quan sát trong quá khứ. Tuy nhiên, mô hình AR không sử dụng hết tất cả giá trị quan sát trong quá khứ để dự đoán hiện tại, mà sẽ định nghĩa một tham số **p** là số giá trị quan sát trong quá khứ được sử dụng.
Mô hình AR được định nghĩa như sau:

$$AR(p) = a_0 + a_1 x_{t-1} + a_2 x_{t-2} + \ldots + a_p x_{t-p}$$

trong đó, $a_i, 0 \leq i \leq p$ là hệ số trong mô hình hồi quy, $x_i, t-p \leq i \leq t-1$ là giá trị quan sát.

2. Moving Average (MA)

Mô hình MA đi phân tích sai số của các thời điểm trước để giảm nhiễu với mục đích dự đoán tốt hơn ở thời điểm hiện tại. Mô hình này cũng sử dụng một tham số q như là một cửa sổ trượt.
Mô hình MA được định nghĩa như sau:

$$MA(q) = \mu + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}$$

trong đó, μ là trung bình của quá trình, $\theta_i, 1 \leq i \leq q$ là hệ số và $\epsilon_i, t-q \leq i \leq t-1$ là sai số ở các thời điểm trước.
Như đã đề cập, điều kiện sử dụng ARIMA là dữ liệu phải ổn định (stationary). Do quá trình MA là white noise, vì thể nó thỏa điều kiện stationary.

$$E(\epsilon_t) = 0,$$
$$\sigma(\epsilon_t) = \alpha = const,$$
$$corr(\epsilon_t, \epsilon_{t-j}) = 0, 0 < j < t$$

Phương trình đầu tiên có nghĩa là trung bình về chuỗi thời gian không thay đổi theo thời gian. Điều thứ hai có nghĩa là phương sai của chuỗi thời gian không đổi. Cái cuối cùng có nghĩa là sai số độc lập với nhau.

3. Integrated (I)

Yêu cầu sử dụng ARIMA là dữ liệu phải ổn định (stationary). Tuy nhiên, trong thực tế thì dữ liệu không phải lúc nào cũng ổn định (stationary). Một cách thường được dùng để chuyển đổi dữ liệu sao cho ổn định là lấy hiệu giữa 2 giá trị quan sát liên tiếp thay vì sử dụng dữ liệu ban đầu. Ta gọi tham số d là số lần dữ liệu được lấy hiệu, ví dụ:

$$\begin{aligned} d = 1: &\quad \Delta x_t = x_t - x_{t-1}, \\ d = 2: &\quad \Delta^2 x_t = \Delta x_t - \Delta x_{t-1}. \end{aligned}$$

trong đó, $\Delta x_i, t - q \leq i \leq i$ là giá trị sau khi lấy hiệu d lần, $\epsilon_i, t - q \leq i \leq i \leq 1$ là white noise.
Ngoài ra còn có 1 số biến thể của ARIMA như là Season ARIMA dùng để xử lý tính chu kỳ theo mùa trong dữ liệu time series.
Tuy nhiên, ARIMA có một số hạn chế như sau:

- Quan hệ tuyến tính: ARIMA không thể biểu diễn các quan hệ phi tuyến tính.
- Dữ liệu 1 biến: trong thực tế, dữ liệu có rất nhiều biến, tuy nhiên 1 mô hình ARIMA chỉ xử lý được 1 biến.

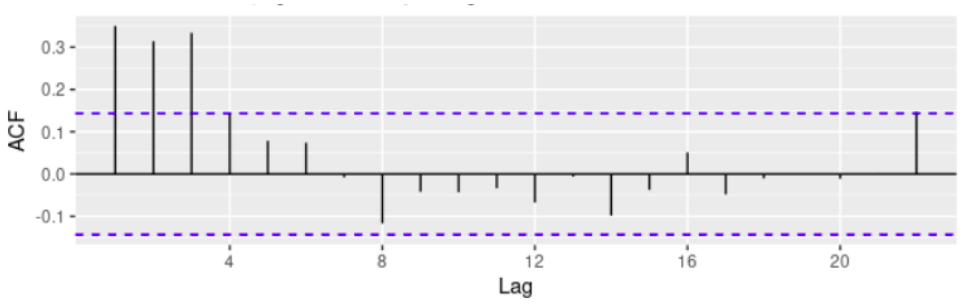
4. Một số cách để tìm giá trị tham số cho mô hình ARIMA

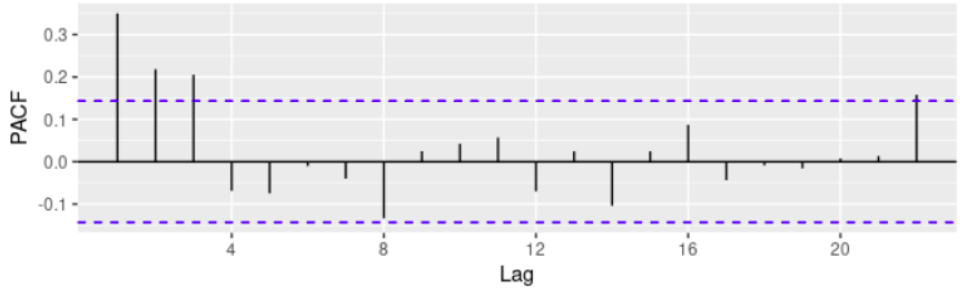
a. Kiểm định Augmented Dickey Fuller Test (ADF Test) cho tham số d

Một cách trực quan, Augmented Dickey Fuller Test (ADF Test) dùng để kiểm tra dữ liệu time series có non-stationary hay không với giả thuyết null là dữ liệu non-stationary. Ta mong muốn bác bỏ giả thuyết này.
Trong bài tập lớn này, ta tìm **d** sao cho p-value < 0.05 tức độ tin cây là 95% và dùng **d** này làm để làm tham số cho mô hình ARIMA(p,**d**,q)

b. Vẽ PACF và ACF cho tham số p và q

ACF cho ta biết autocorrelation giữa y_t và y_{t-k} với k khác nhau. Nếu y_t và y_{t-1} tương quan, y_{t-1} và y_{t-2} tương quan thì y_t và t_{t-2} tương quan. Để giải quyết vấn đề này, người ta dùng PACF. PACF sẽ loại bỏ ảnh hưởng của các giá trị trung gian $y_{t-1}, y_{t-2}, ..., y_{t-k-1}$ khi đo sự tương quan giữa y_t và t_{t-k}.
Ví dụ:





Ví dụ về ACF và PACF

Dữ liệu tuân theo ARIMA(p,d,0) nếu ACF và PACF thỏa :

- ACF giảm theo cấp số nhân hoặc có hình sin
- Có 1 significant spike tại lag p trong PACF, nhưng sau đó thì không.

Dữ liệu tuân theo ARIMA(0,d,q) nếu ACF và PACF thỏa :

- PACF giảm theo cấp số nhân hoặc có hình sin
- Có 1 significant spike tại lag q trong ACF, nhưng sau đó thì không

Ví dụ ở 2 hình trên, p=3 được coi là tối ưu vì ACF có thể được coi là giảm theo cấp số nhân và ở PACF có liên tiếp significant spike ở lag 1,2,3 nhưng sau đó thì không còn; q=3 được coi là tối ưu cũng theo lý luận như trên.

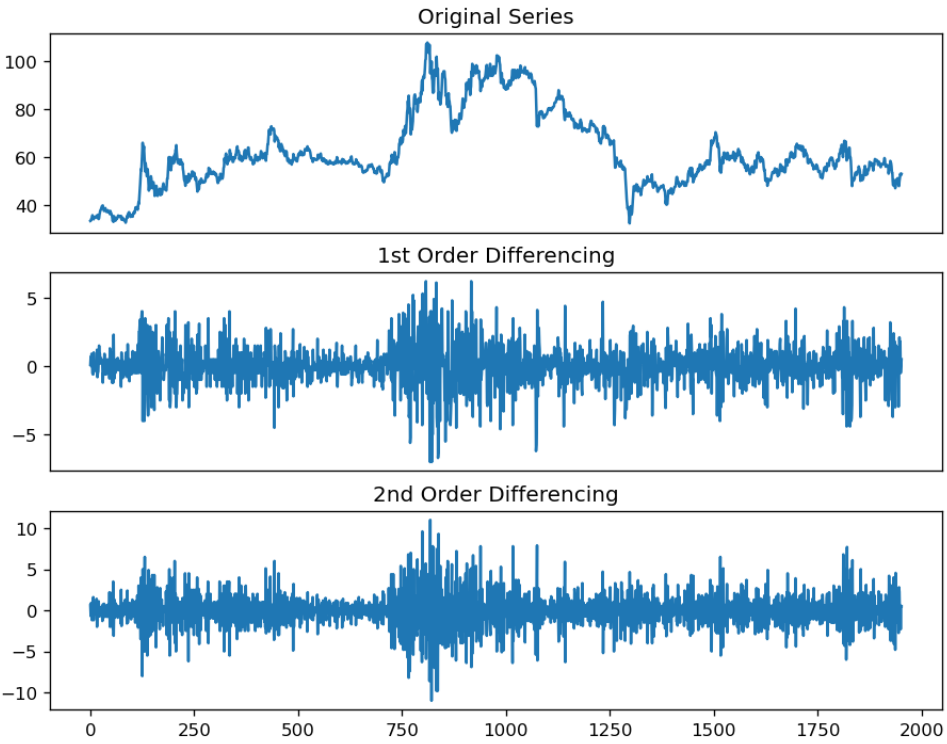
5. Áp dụng ARIMA cho từng loại cổ phiếu

a. ARIMA cho BVH

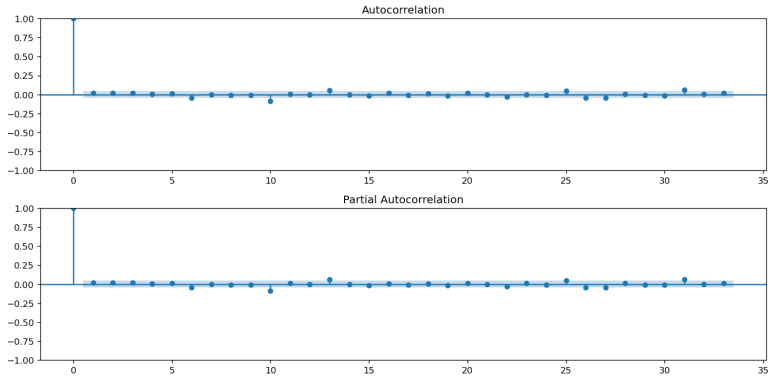
Tìm tham số **d** dựa vào ADF test. Hình 16 mô tả dữ liệu ở các **d** khác nhau.

	d=0	d=1	d=2
p-value	0.136	0.000	0.000

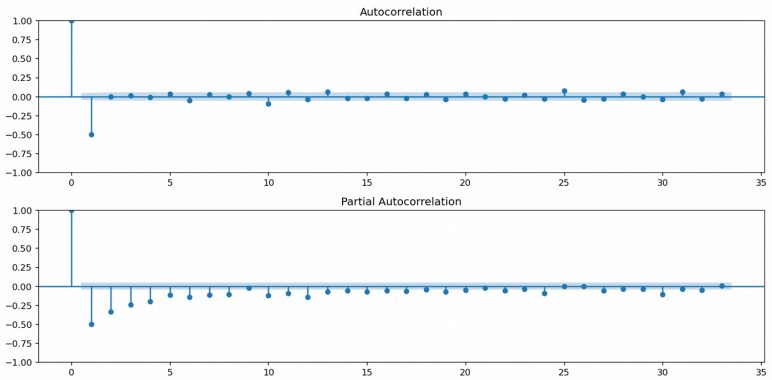
Do p-value của d=1 và d=2 < 0.05, chọn d=1;2 để đi tìm kiếm tham số tối ưu.
Tìm tham số **p**, **q** dựa vào ACF và PACF ở Hình 17. Khi d=1, ta chọn $p \in \{0, 1\}, q \in \{0, 1\}$. Khi d=2, chọn $p \in \{7, 8, 9\}, q \in \{0, 1, 2\}$



Hình 16: *Dữ liệu gốc, dữ liệu khi lấy hiệu 1 lần và dữ liệu khi lấy hiệu 2 lần*



(a) *ACF, PACF khi d=1*

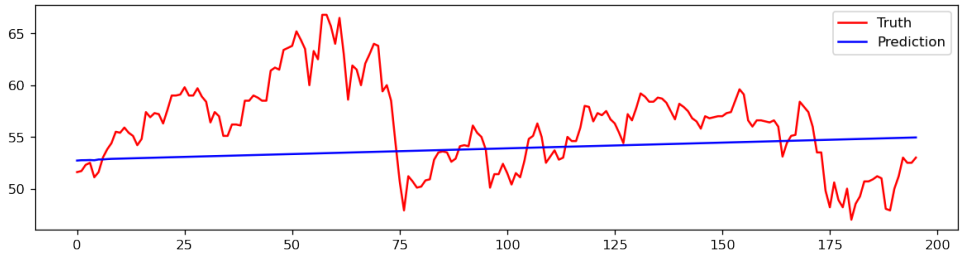


(b) *ACF, PACF khi d=2*

Hình 17: *ACF và PACF để tìm kiếm p và q*

	(p,d,q)	RMSE	MAPE	MAE
0	(0,1,0)	5.262215	0.072899	4.240561
1	(0,1,1)	5.280393	0.073162	4.256554
2	(1,1,0)	5.281626	0.07318	4.257638
3	(1,1,1)	5.322205	0.073773	4.293571
4	(7,2,0)	57.980216	0.932725	51.523622
5	(7,2,1)	4.899506	0.06786	3.90448
6	(7,2,2)	4.918215	0.068117	3.920903
7	(8,2,0)	57.532239	0.925677	51.137017
8	(8,2,1)	4.868898	0.067433	3.877071
9	(8,2,2)	4.888682	0.067716	3.896207
10	(9,2,0)	56.829339	0.914494	50.523093
11	(9,2,1)	**4.85365**	**0.067222**	**3.863489**
12	(9,2,2)	4.904047	0.067919	3.908226

Kết quả ARIMA với các tham số khác nhau dự đoán cho BVH. Trong đó bộ tham số (p,d,q)=(9,2,1) cho kết quả tốt nhất.



Hình 18: *Trực quan hóa kết quả dự đoán của mô hình ARIMA (9,2,1) so với thực tế. Kết quả dự đoán của hình không được tốt vì nó chỉ cho ra 1 đường thẳng, không bắt được những lần lên xuống của dữ liệu.*

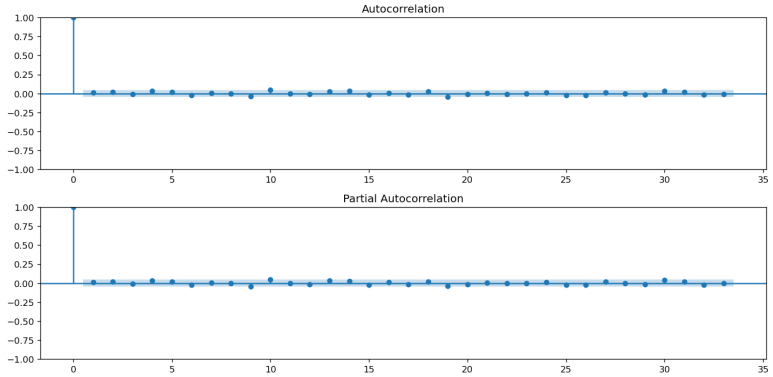
b. ARIMA cho BID

Tìm tham số **d** dựa vào ADF test. Hình 19 mô tả dữ liệu ở các **d** khác nhau.

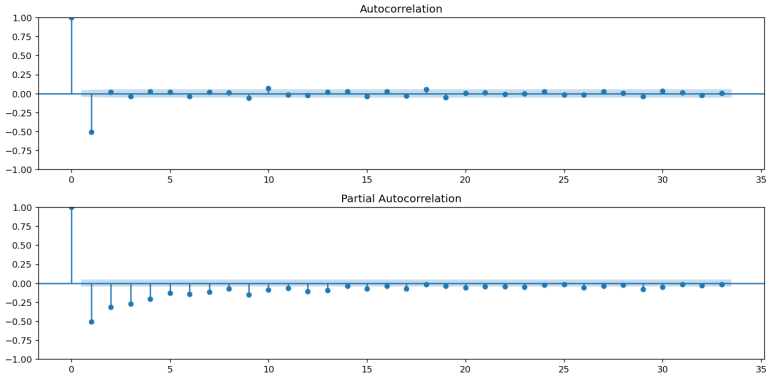
	d=0	d=1	d=2
p-value	0.234	0.000	0.000

Do p-value của d=1 và d=2 < 0.05, chọn d=1;2 để đi tìm kiếm tham số tối ưu.
Tìm tham số **p**, **q** dựa vào ACF và PACF ở Hình 20. Khi d=1, ta chọn $p \in \{0, 1\}, q \in \{0, 1\}$. Khi d=2, chọn $p \in \{6, 7, 8\}, q \in \{0, 1, 2\}$

Hình 19: *Dữ liệu gốc, dữ liệu khi lấy hiệu 1 lần và dữ liệu khi lấy hiệu 2 lần*



(a) *ACF, PACF khi d=1*

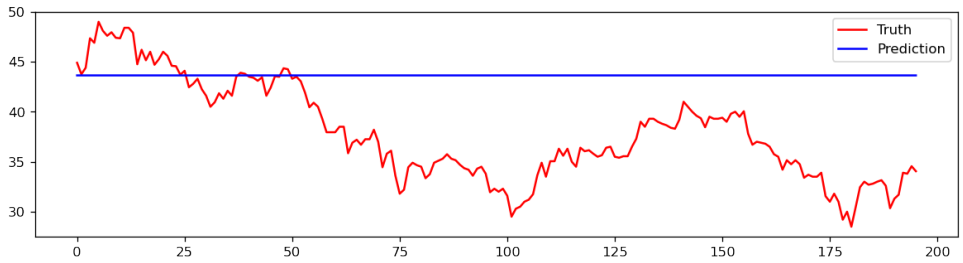


(b) *ACF, PACF khi d=2*

Hình 20: *ACF và PACF để tìm kiếm p và q*

	(p,d,q)	RMSE	MAPE	MAE
0	(0,1,0)	7.521763	0.185108	6.448469
1	(0,1,1)	7.511632	0.184841	6.439297
2	(1,1,0)	**7.510835**	**0.18482**	**6.438573**
3	(1,1,1)	7.578013	0.186597	6.500007
4	(6,2,0)	87.108687	2.134787	75.501239
5	(6,2,1)	9.360026	0.232257	8.104903
6	(6,2,2)	9.502401	0.236096	8.24
7	(7,2,0)	89.444968	2.191693	77.518946
8	(7,2,1)	9.461033	0.234982	8.20089
9	(7,2,2)	9.34083	0.231773	8.087797
10	(8,2,0)	88.685852	2.172791	76.845384
11	(8,2,1)	9.329917	0.231436	8.075888
12	(8,2,2)	9.502341	0.236132	8.241322

Kết quả ARIMA với các tham số khác nhau dự đoán cho BID. Trong đó bộ tham số (p,d,q)=(1,1,0) cho kết quả tốt nhất.



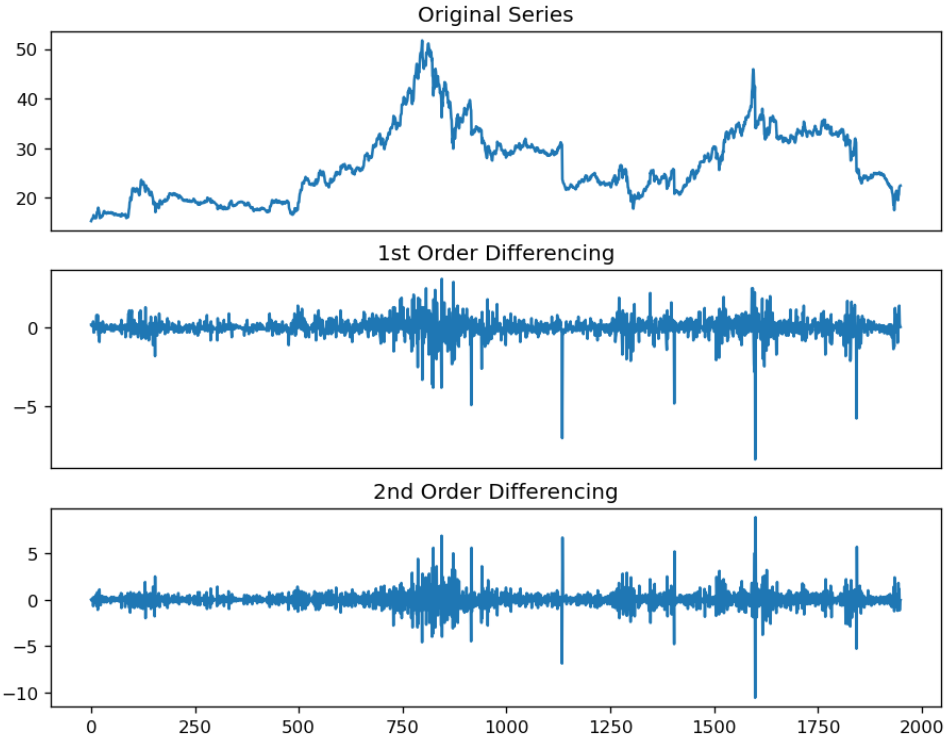
Hình 21: *Trực quan hóa kết quả dự đoán của mô hình ARIMA (1,1,0) so với thực tế. Kết quả dự đoán của hình không được tốt vì nó chỉ cho ra 1 đường thẳng, không bắt được những lần lên xuống của dữ liệu.*

c. ARIMA cho ACB

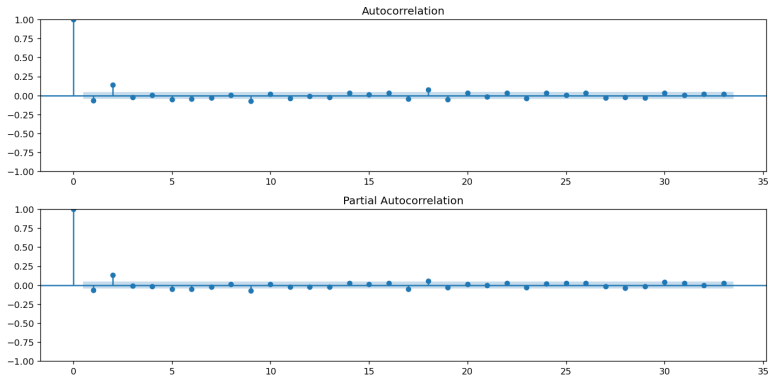
Tìm tham số **d** dựa vào ADF test. Hình 22 mô tả dữ liệu ở các **d** khác nhau.

	d=0	d=1	d=2
p-value	0.286	0.000	0.000

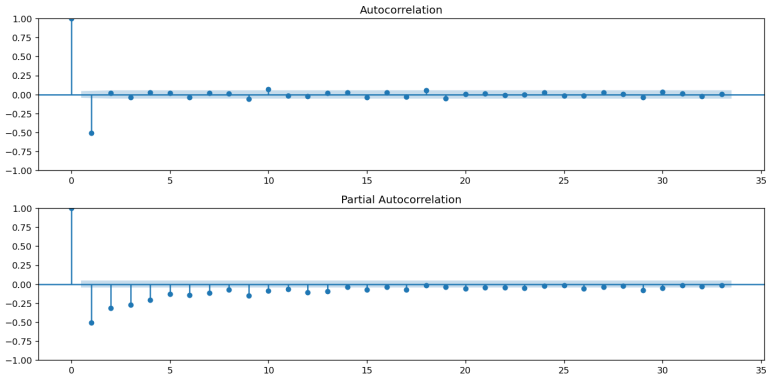
Do p-value của d=1 và d=2 < 0.05, chọn d=1;2 để đi tìm kiếm tham số tối ưu.
Tìm tham số **p**, **q** dựa vào ACF và PACF ở Hình 23. Khi d=1, ta chọn $p \in \{1, 2, 3\}, q \in \{1, 2, 3\}$. Khi d=2, chọn $p \in \{6, 7, 8\}, q \in \{1, 2, 3\}$



Hình 22: *Dữ liệu gốc, dữ liệu khi lấy hiệu 1 lần và dữ liệu khi lấy hiệu 2 lần (ACB)*



(a) *ACF, PACF khi d=1 (ACB)*

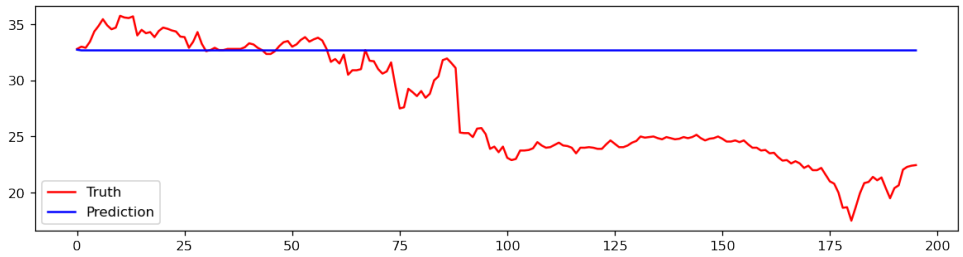


(b) *ACF, PACF khi d=2 (ACB)*

Hình 23: *ACF và PACF để tìm kiếm p và q (ACB)*

	(p,d,q)	RMSE	MAPE	MAE
0	(1,1,1)	7.134289	0.242638	5.735193
1	(1,1,2)	**7.069808**	**0.240708**	**5.693627**
2	(1,1,3)	7.173631	0.243983	5.766696
3	(2,1,1)	7.073409	0.240809	5.69572
4	(2,1,2)	7.073051	0.240799	5.69551
5	(2,1,3)	7.18874	0.244434	5.77612
6	(3,1,1)	7.073422	0.24081	5.695728
7	(3,1,2)	7.200384	0.244792	5.78386
8	(3,1,3)	7.212915	0.245179	5.792258
9	(6,2,1)	8.190086	0.278553	6.579653
10	(6,2,2)	8.156625	0.277383	6.552151
11	(6,2,3)	8.106188	0.275645	6.511406
12	(7,2,1)	8.210316	0.279255	6.596116
13	(7,2,2)	8.169373	0.277848	6.562888
14	(7,2,3)	8.001255	0.27206	6.426159
15	(8,2,1)	8.196325	0.278767	6.584642
16	(8,2,2)	8.152478	0.277264	6.54914
17	(8,2,3)	8.204426	0.279054	6.591402

Kết quả ARIMA với các tham số khác nhau dự đoán cho ACB. Trong đó bộ tham số (p,d,q)=(1,1,2) cho kết quả tốt nhất.



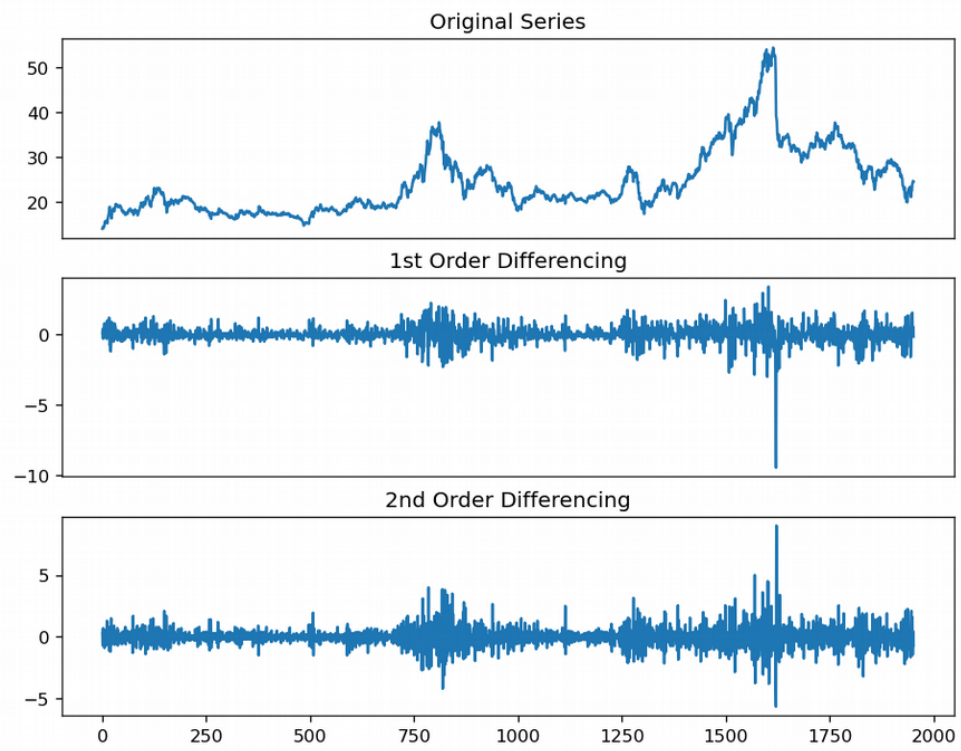
Hình 24: *Trực quan hóa kết quả dự đoán của mô hình ARIMA (1,1,2) so với thực tế. Kết quả dự đoán của hình không được tốt vì nó chỉ cho ra 1 đường thẳng, không bắt được những lần lên xuống của dữ liệu.*

d. ARIMA cho CTG

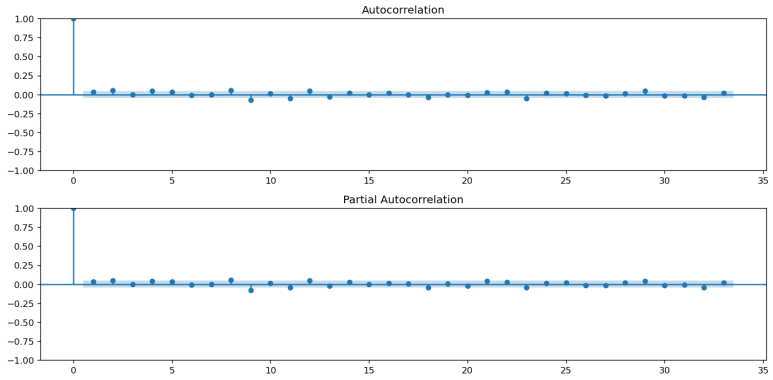
Tìm tham số **d** dựa vào ADF test. Hình 25 mô tả dữ liệu ở các **d** khác nhau.

	d=0	d=1	d=2
p-value	0.146	0.000	0.000

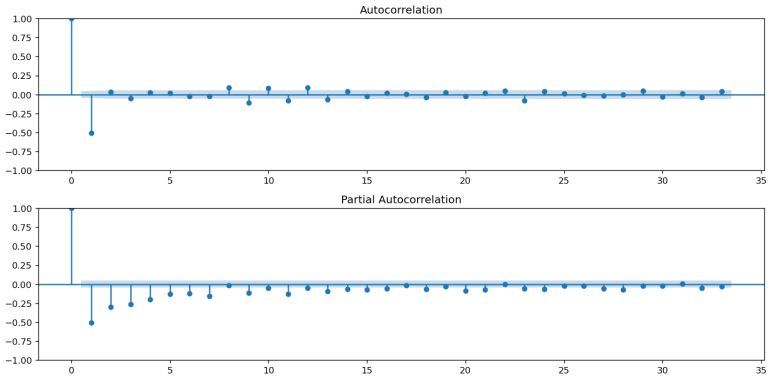
Do p-value của d=1 và d=2 < 0.05, chọn d=1;2 để đi tìm kiếm tham số tối ưu.
Tìm tham số **p**, **q** dựa vào ACF và PACF ở Hình 26. Khi d=1, ta chọn $p \in \{0, 1\}, q \in \{0, 1\}$. Khi d=2, chọn $p \in \{6, 7, 8\}, q \in \{0, 1, 2\}$



Hình 25: *Dữ liệu gốc, dữ liệu khi lấy hiệu 1 lần và dữ liệu khi lấy hiệu 2 lần (CTG)*



(a) *ACF, PACF khi d=1 (CTG)*

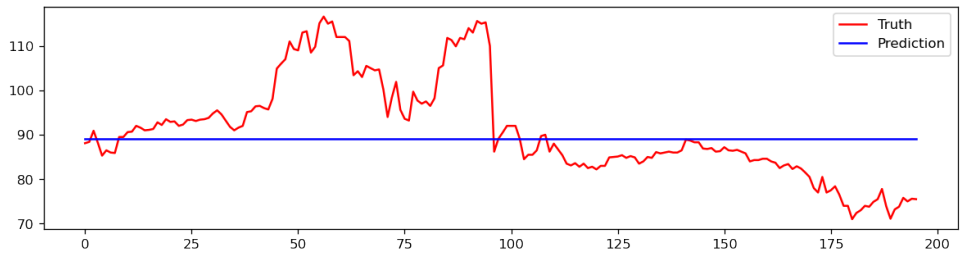


(b) *ACF, PACF khi d=2 (CTG)*

Hình 26: *ACF và PACF để tìm kiếm p và q (CTG)*

	(p,d,q)	RMSE	MAPE	MAE
0	(0,1,0)	7.829185	0.267076	6.877806
1	(0,1,1)	7.815075	0.266532	6.863165
2	(1,1,0)	**7.812765**	**0.266443**	**6.860772**
3	(1,1,1)	7.901216	0.26984	6.952127
4	(6,2,0)	34.022687	1.145809	29.675016
5	(6,2,1)	9.22433	0.315331	8.130971
6	(6,2,2)	9.268056	0.316965	8.174634
7	(7,2,0)	27.262629	0.919063	23.777251
8	(7,2,1)	9.229822	0.315534	8.13638
9	(7,2,2)	9.179527	0.313666	8.08654
10	(8,2,0)	26.980023	0.909593	23.531094
11	(8,2,1)	9.233437	0.315656	8.139537
12	(8,2,2)	9.238236	0.315835	8.144332

Kết quả ARIMA với các tham số khác nhau dự đoán cho CTG. Trong đó bộ tham số (p,d,q)=(1,1,0) cho kết quả tốt nhất.



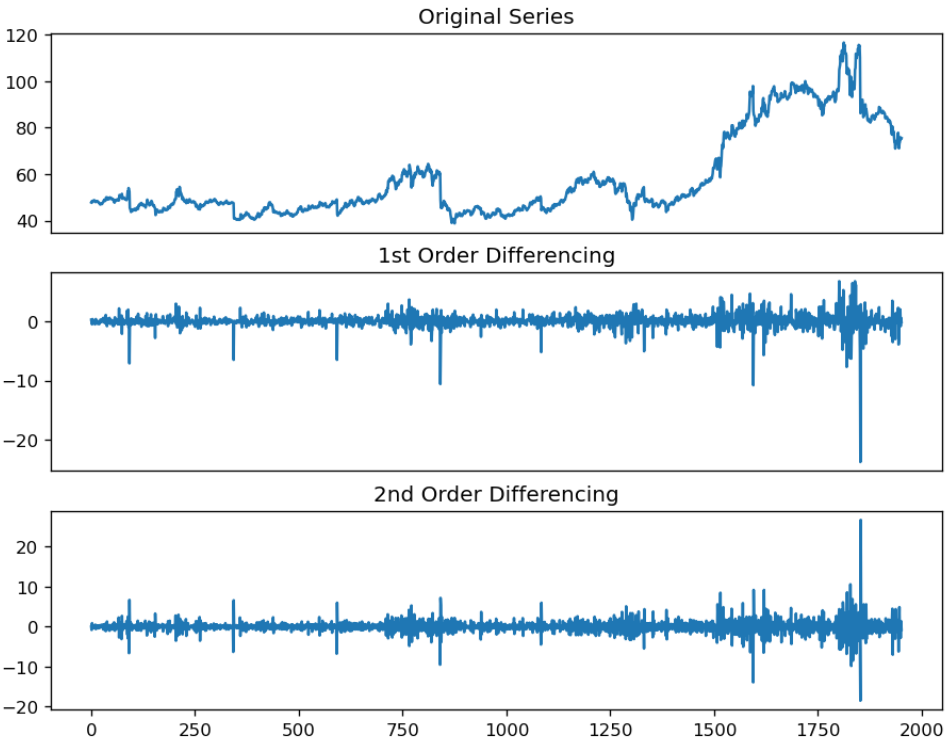
Hình 27: *Trực quan hóa kết quả dự đoán của mô hình ARIMA (1,1,0) so với thực tế. Kết quả dự đoán của hình không được tốt vì nó chỉ cho ra 1 đường thẳng, không bắt được những lần lên xuống của dữ liệu.*

e. ARIMA cho FPT

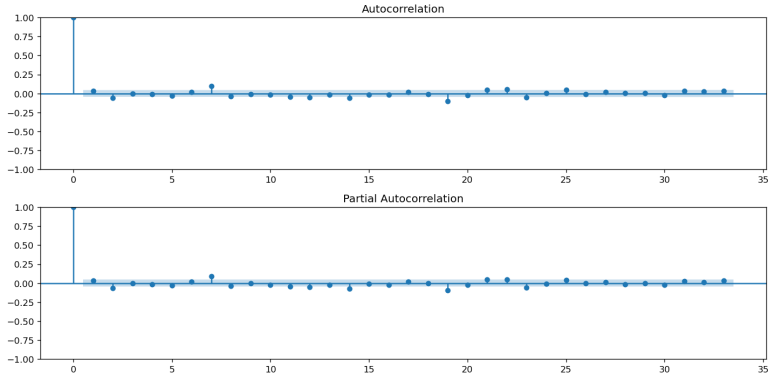
Tìm tham số **d** dựa vào ADF test. Hình 28 mô tả dữ liệu ở các **d** khác nhau.

	d=0	d=1	d=2
p-value	0.764	0.000	0.000

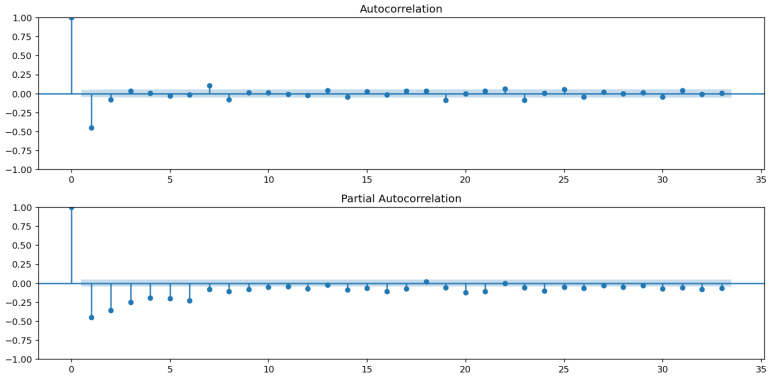
Do p-value của d=1 và d=2 < 0.05, chọn d=1;2 để đi tìm kiếm tham số tối ưu.
Tìm tham số **p**, **q** dựa vào ACF và PACF ở Hình 29. Khi d=1, ta chọn $p \in \{0, 1\}, q \in \{0, 1\}$. Khi d=2, chọn $p \in \{6, 7, 8\}, q \in \{2, 3, 4\}$



Hình 28: *Dữ liệu gốc, dữ liệu khi lấy hiệu 1 lần và dữ liệu khi lấy hiệu 2 lần (FPT)*



(a) *ACF, PACF khi d=1 (FPT)*

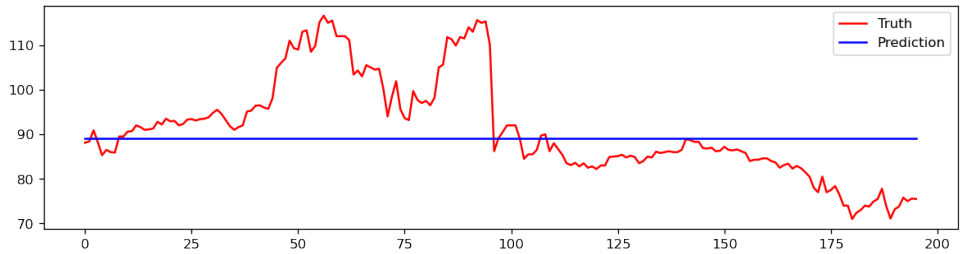


(b) *ACF, PACF khi d=2 (FPT)*

Hình 29: *ACF và PACF để tìm kiếm p và q (FPT)*

	(p,d,q)	RMSE	MAPE	MAE
0	(0,1,0)	**11.389534**	**0.092189**	**8.677041**
1	(0,1,1)	11.397727	0.092142	8.676622
2	(1,1,0)	11.397504	0.092143	8.676637
3	(1,1,1)	11.390452	0.092184	8.677027
4	(6,2,2)	12.666346	0.11697	10.488661
5	(6,2,3)	13.047548	0.121702	10.846072
6	(6,2,4)	13.259635	0.124181	11.038761
7	(7,2,2)	12.351685	0.112423	10.165105
8	(7,2,3)	12.461475	0.114033	10.285021
9	(7,2,4)	12.625945	0.116348	10.453148
10	(8,2,2)	14.02507	0.132151	11.579898
11	(8,2,3)	12.719234	0.117568	10.548046
12	(8,2,4)	12.331524	0.112104	10.141997

Kết quả ARIMA với các tham số khác nhau dự đoán cho FPT. Trong đó bộ tham số (p,d,q)=(0,1,0) cho kết quả tốt nhất.



Hình 30: *Trực quan hóa kết quả dự đoán của mô hình ARIMA (0,1,0) so với thực tế. Kết quả dự đoán của hình không được tốt vì nó chỉ cho ra 1 đường thẳng, không bắt được những lần lên xuống của dữ liệu.*

III. Mô hình Support Vector Regression (SVR)

1. Giới thiệu mô hình

Mô hình **Support Vector Regression - SVR** được tạo ra dựa trên ý tưởng từ mô hình Support Vector Machine - SVM. Mục tiêu của SVR là tìm được một siêu phẳng chứa được tối đa các training observation trong giới hạn margin ε (tolerance level).

2. Bài toán tối ưu trong SVR

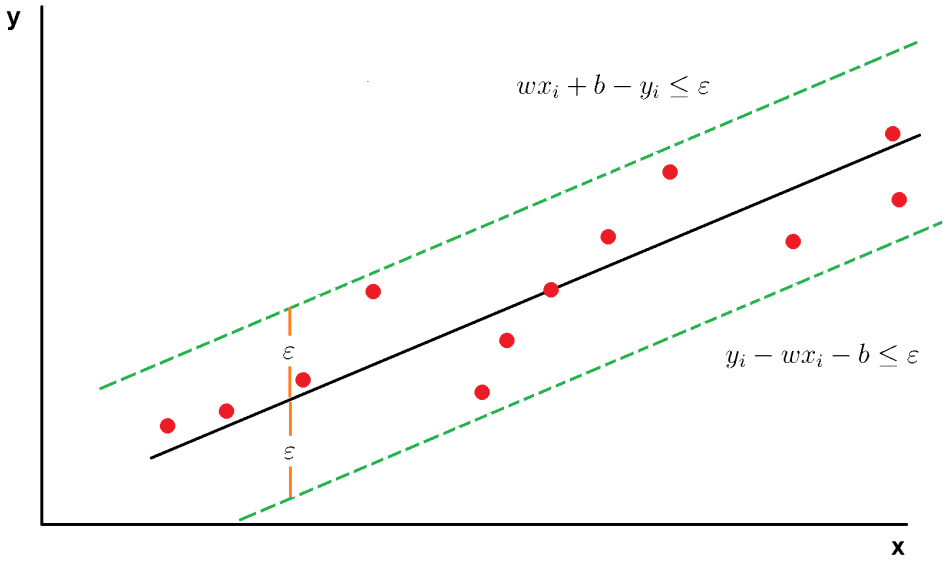
Từ mục tiêu của mô hình SVR, dễ thấy rằng SVR là một bài toán tối đa hóa margin ε và sẽ tương đương với bài toán tối ưu như sau:

- **Minimize:**

$$\min \frac{1}{2}\|w\|^2$$

- **Subject to:**

$$y_i - wx_i - b \leq \varepsilon$$
$$wx_i + b - y_i \leq \varepsilon$$

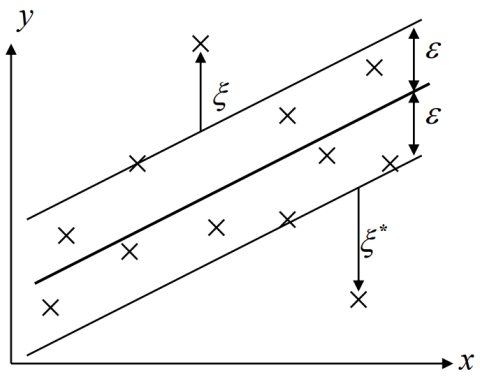


Support Vector Regression - SVR

Gọi y_i là giá trị thực thứ i và $\hat{y}_i$ là giá trị dự đoán thứ i với $\hat{y}_i$ có dạng $\hat{y}_i = wx_i + b$, từ các ràng buộc trong bài toán tối ưu trên, ta suy ra: $|y - \hat{y}| \leq \varepsilon$.

3. Soft margin

Từ các ràng buộc của bài toán tối ưu trên, dễ thấy rằng tất cả mọi quan sát không được nằm ngoài margin, và nó tương tự với trường hợp hard margin trong bài toán phân loại của SVM. Bài toán tối ưu này đôi khi sẽ không thể tìm ra được lời giải, vì vậy ý tưởng sẽ là sử dụng soft margin để thay thế.



Soft margin

Hình trên thể hiện ý tưởng cơ bản của mô hình dự đoán SVR với soft margin. Cho tập dữ liệu huấn luyện $\{(x_1, y_1),\ldots, (x_N, y_N) \in X \times \mathbb{R}\}$ với X là không gian các mẫu input. Trong trường hợp này, bài toán tối ưu của SVR sẽ trở thành:

- **Minimize:**

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

- **Subject to:**

$$\begin{gathered} y_i - wx_i - b \leq \varepsilon + \xi_i \\ wx_i + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{gathered}$$

Trong đó ξ_i và ξ_i^* là các biến slack để giải quyết các ràng buộc không khả thi của bài toán tối ưu với mỗi mẫu thứ i. Hằng số C > 0 xác định sự đánh đổi giữa độ phẳng của hàm f(x) và độ lệch lớn hơn ε. Nếu hằng số C càng lớn thì các biến slack sẽ càng nhỏ để để thỏa bài toán tối ưu, điều này dẫn đến mô hình sẽ linh hoạt hơn vì độ lệnh nhỏ nhưng phương sai của mô hình sẽ lớn hơn. Nếu ta giảm hằng số C tức là giảm cost của việc có các quan sát bên ngoài margin thì về cơ bản ta sẽ có được nhiều quan sát bên ngoài margin hơn, mô hình sẽ trở nên ít linh hoạt hơn và các phương sai của mô hình sẽ giảm. Vì vậy ta cần chọn giá trị hợp lý cho C để điều chỉnh cho bài toán hồi quy trở nên tốt hơn.

4. Các loại hàm kernel

Sự linh hoạt của mô hình SVR sẽ liên quan đến các hàm kernel khác nhau của nó.

a. Linear kernel

Đây là trường hợp đơn giản, với kernel là tích vô hướng của hai vector:

$$k(x, y) = x^T y$$

Trong thư viện sklearn, để dùng hàm nhân đa thức cho SVR, ta thiết lập kernel = 'linear'. Hàm nhân tuyến tính nhanh về mặt tính toán và ít bị overfitting.

b. Polynomial kernel

Công thức của hàm nhân đa thức như sau:

$$k(x, y) = (r + \gamma x^T y)^d$$

Trong đó, d > 0, d $\in \mathbb{R}$ chỉ bậc của đa thức. Ở đây, d không cần phải là một số tự nhiên vì mục đích chính của ta là cách tính kernel chứ không phải bậc đa thức. Trong thư viện sklearn, để dùng hàm nhân đa thức cho SVR, ta thiết lập kernel = 'poly' và các hệ số d(degree), γ(gamma), r(coef0).
Hàm nhân đa thức phù hợp để giải quyết các bài toán phi tuyến tính.

c. Radial Basic Function kernel (RBF kernel)

Công thức của hàm nhân đa thức như sau:

$$k(x, y) = \exp(-\gamma \|x - y\|^2), \gamma > 0$$

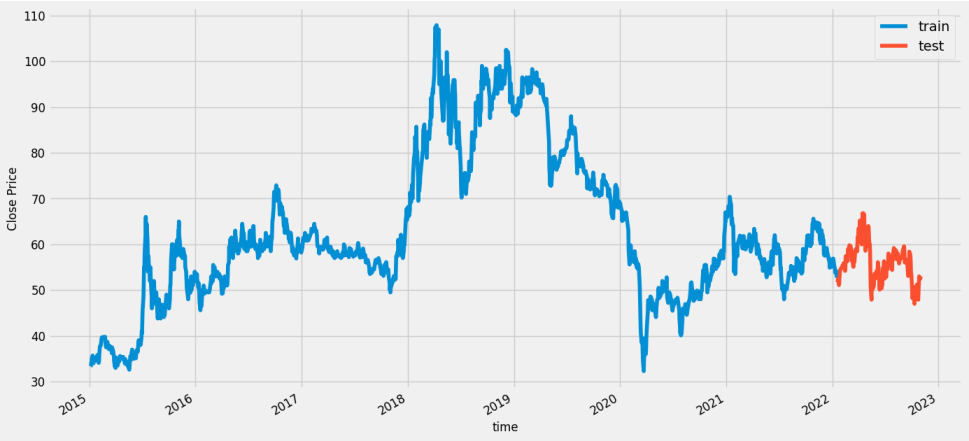
Trong thư viện sklearn, để dùng hàm nhân RBF cho SVR, ta thiết lập kernel = 'rbf' và hệ số γ(gamma).
Hàm nhân RBF không chỉ hiệu quả về mặt tính toán mà còn phù hợp để giải quyết các bài toán phi tuyến tính. Trong thực tế, RBF kernel được dùng nhiều nhất.

5. Áp dụng mô hình SVR vào dự đoán giá chứng khoán

Để áp dụng mô hình SVR vào dự đoán giá chứng khoán, nhóm có một số thiết lập ban đầu như sau:

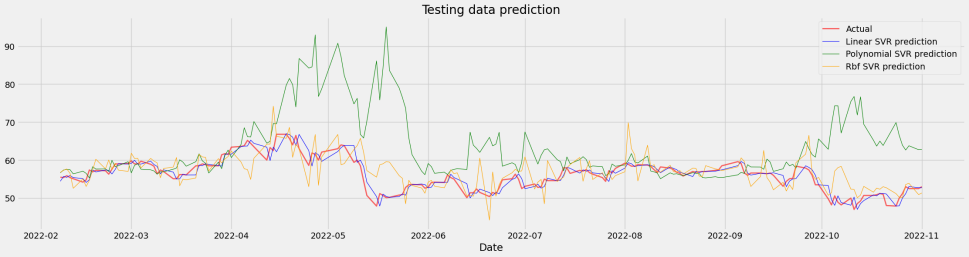
- Dùng 90% tập dữ liệu của từng công ty để train và 10% còn lại để test.
- Chuẩn hóa dữ liệu bằng StandardScaler
- Dự đoán dựa trên thuộc tính close của tập dữ liệu
- Dự đoán ngày hiện tại dựa trên dữ liệu của 9 ngày trước đó.
- Sử dụng mô hình SRV có hằng số C = 1000, với 3 loại hàm nhân: linear kernel; polynomial kernel với d = 2, $r = 0$; RBF kernel với $\gamma = 0.5$.

a. Dự đoán cho BVH



Trực quan hóa dữ liệu tập train và tập test của công ty BVH

Áp dụng mô hình SVR trên tập dữ liệu của công ty BVH, ta có được kết quả dự đoán so với giá trị thực trên tập test được thể hiện trong hình sau:



Trực quan hóa kết quả dự đoán so với giá trị thực trên tập test của công ty BVH

Bảng sau là kết quả một số độ đo error của mô hình trên tập test:

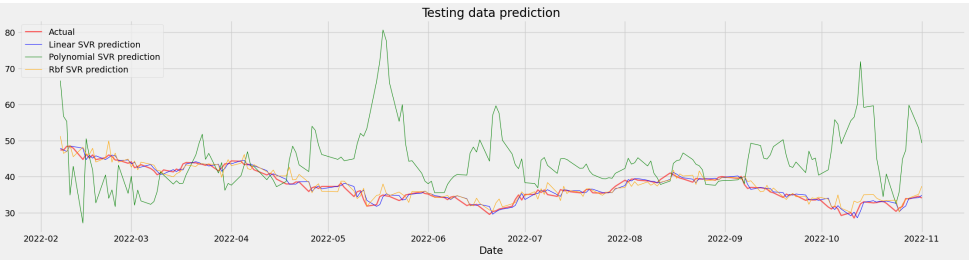
Kernel	MSE	MAPE	MAE
Linear	2.985	0.0189	1.0569
Polynomial	138.749	0.1429	7.6794
RBF	13.3278	0.0483	2.6793

Qua các kết quả trên, ta thấy được mô hình SVR với linear kernel cho ra kết quả dự đoán gần với thực tế nhất.

b. Dự đoán cho BID

Trực quan hóa dữ liệu tập train và tập test của công ty BID

Áp dụng mô hình SVR trên tập dữ liệu của công ty BID, ta có được kết quả dự đoán so với giá trị thực trên tập test được thể hiện trong hình sau:



Trực quan hóa kết quả dự đoán so với giá trị thực trên tập test của công ty BID

Bảng sau là kết quả một số độ đo error của mô hình trên tập test:

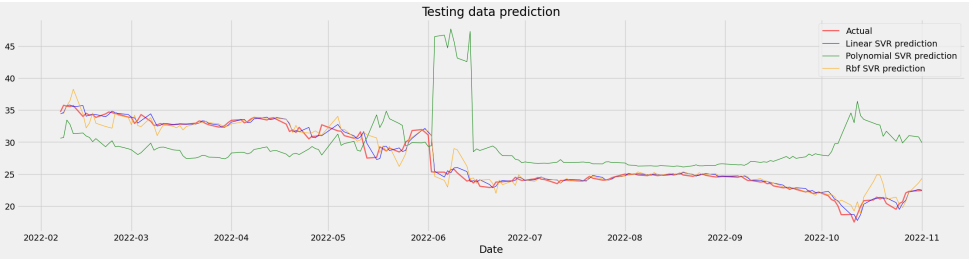
Kernel	MSE	MAPE	MAE
Linear	0.969	0.0206	0.7497
Polynomial	165.21	0.2717	9.5154
RBF	2.2836	0.0318	1.1728

Qua các kết quả trên, ta thấy được mô hình SVR với linear kernel cho ra kết quả dự đoán gần với thực tế nhất.

c. Dự đoán cho ACB

Trực quan hóa dữ liệu tập train và tập test của công ty ACB

Áp dụng mô hình SVR trên tập dữ liệu của công ty ACB, ta có được kết quả dự đoán so với giá trị thực trên tập test được thể hiện trong hình sau:



Trực quan hóa kết quả dự đoán so với giá trị thực trên tập test của công ty ACB

Bảng sau là kết quả một số độ đo error của mô hình trên tập test:

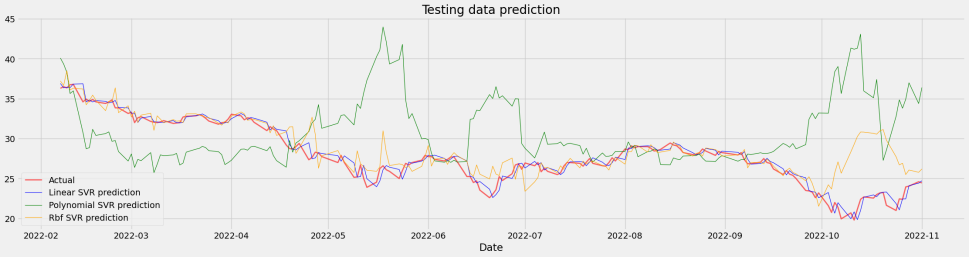
Kernel	MSE	MAPE	MAE
Linear	0.5135	0.0165	0.4347
Polynomial	45.0151	0.2014	5.0468
RBF	1.3568	0.0285	0.7462

Qua các kết quả trên, ta thấy được mô hình SVR với linear kernel cho ra kết quả dự đoán gần với thực tế nhất.

d. Dự đoán cho CTG

Trực quan hóa dữ liệu tập train và tập test của công ty CTG

Áp dụng mô hình SVR trên tập dữ liệu của công ty CTG, ta có được kết quả dự đoán so với giá trị thực trên tập test được thể hiện trong hình sau:



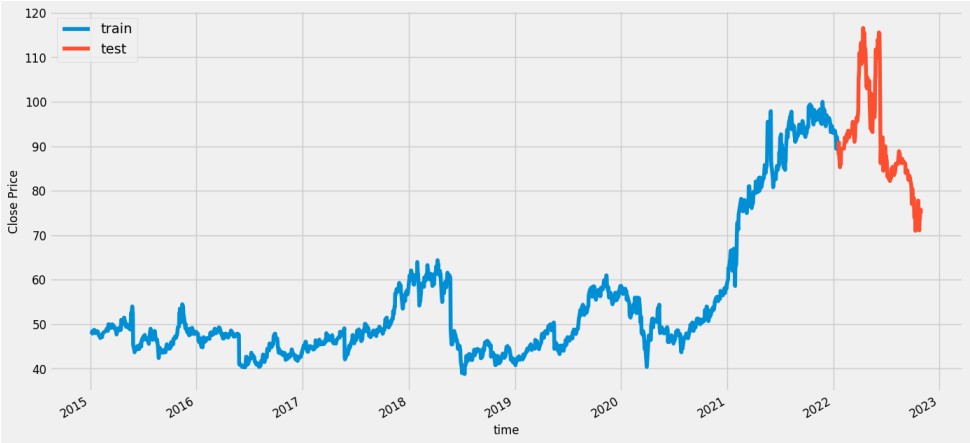
Trực quan hóa kết quả dự đoán so với giá trị thực trên tập test của công ty CTG

Bảng sau là kết quả một số độ đo error của mô hình trên tập test:

Kernel	MSE	MAPE	MAE
Linear	0.4982	0.01996	0.5247
Polynomial	52.4081	0.20798	5.2892
RBF	5.9562	0.0573	1.4004

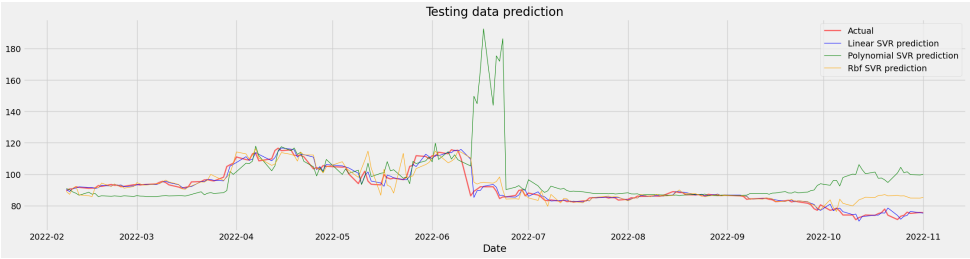
Qua các kết quả trên, ta thấy được mô hình SVR với linear kernel cho ra kết quả dự đoán gần với thực tế nhất.

e. Dự đoán cho FPT



Trực quan hóa dữ liệu tập train và tập test của công ty FPT

Áp dụng mô hình SVR trên tập dữ liệu của công ty FPT, ta có được kết quả dự đoán so với giá trị thực trên tập test được thể hiện trong hình sau:



Trực quan hóa kết quả dự đoán so với giá trị thực trên tập test của công ty FPT

Bảng sau là kết quả một số độ đo error của mô hình trên tập test:

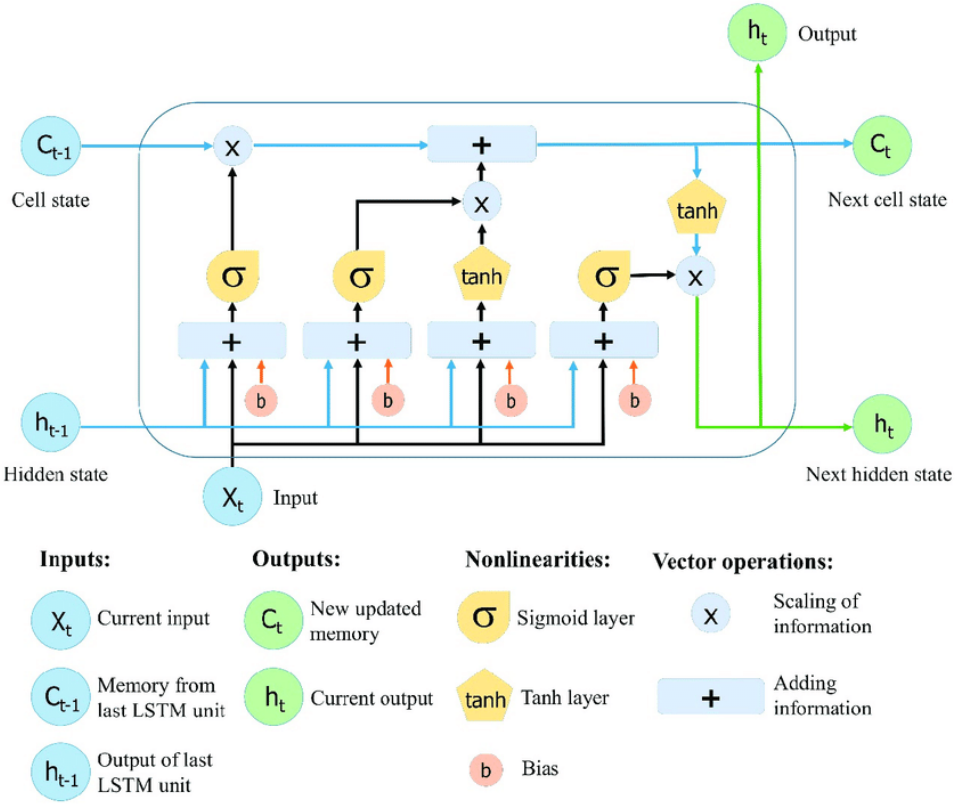
Kernel	MSE	MAPE	MAE
Linear	7.1543	0.0163	1.5021
Polynomial	369.4572	0.1186	10.134
RBF	27.9378	0.0386	3.3757

Qua các kết quả trên, ta thấy được mô hình SVR với linear kernel cho ra kết quả dự đoán gần với thực tế nhất.

IV. Mô hình Long short-term memory - LSTM

1. Giới thiệu mô hình

Đặc trưng của dữ liệu chứng khoán là một dạng dữ liệu liên tục theo thời gian (time series) và giá trị trong tương lai phụ thuộc vào những biến động trong quá khứ.
Mô hình LSTM cực kỳ mạnh trong việc xử lý các dữ liệu chuỗi thời gian, đồng thời nó cũng có thể tìm ra những dữ liệu ẩn (hidden patterns) trong dữ liệu ở quá khứ để đưa ra dự đoán xa hơn vào tương lai

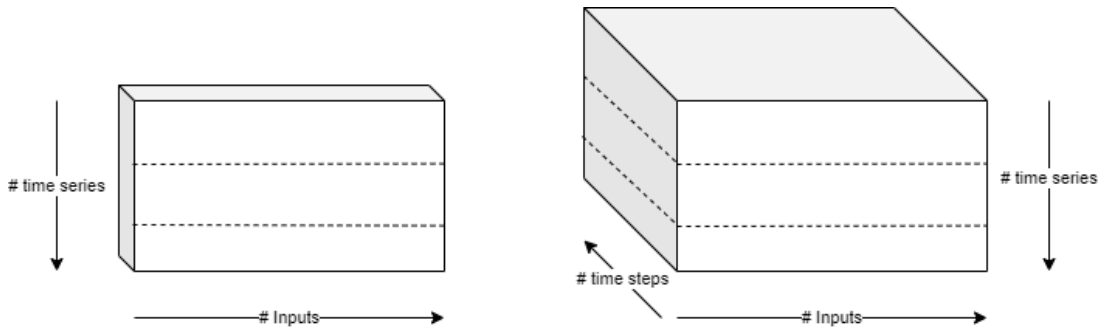


Một cell của mô hình LSTM

Nhóm sẽ không đi quá sâu vào cách hoạt động của mô hình, chúng ta chỉ chú ý đến bản chất của mô hình này. Rằng nó sẽ mang được những thông tin của chuỗi cổ phiếu trong quá khứ, truyền đến hiện tại để học và dự đoán ra xu hướng biến động của cổ phiếu này trong tương lai.

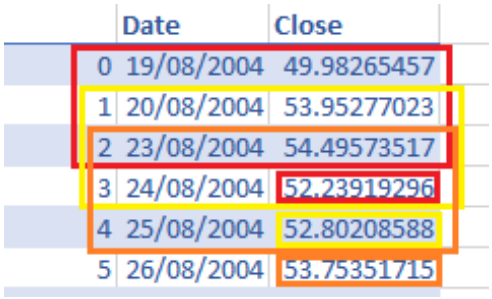
2. Xây dựng mô hình

Việc đầu tiên, chúng ta sẽ thêm một biến là timestep để chuyển đổi ma trận của ta từ hai chiều thành ba chiều



2D to 3D

Sử dụng kỹ thuật này sẽ giúp mô hình chúng ta nhìn về timestep lần trong quá khứ để lấy đó làm dữ liệu học ra kết quả của ngày tiếp theo



	Date	Close
0	19/08/2004	49.98265457
1	20/08/2004	53.95277023
2	23/08/2004	54.49573517
3	24/08/2004	52.23919296
4	25/08/2004	52.80208588
5	26/08/2004	53.75351715

Timestep

Hình trên là một ví dụ cho việc ta lấy giá Close của 3 ngày gần nhất để dự đoán ra giá của ngày thứ 4, và cứ thế.

Nhóm sẽ sử dụng thư viện Tensorflow để dễ dàng hơn trong việc xây dựng mô hình LSTM này, thay vì code hoàn toàn từ đầu.

Mô hình sau khi xây dựng sẽ có dạng như sau

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 3, 100)            40800
_________________________________________________________________
dropout_4 (Dropout)          (None, 3, 100)            0
_________________________________________________________________
lstm_5 (LSTM)                (None, 3, 50)             30200
_________________________________________________________________
dropout_5 (Dropout)          (None, 3, 50)             0
_________________________________________________________________
lstm_6 (LSTM)                (None, 3, 50)             20200
_________________________________________________________________
dropout_6 (Dropout)          (None, 3, 50)             0
_________________________________________________________________
lstm_7 (LSTM)                (None, 50)                20200
_________________________________________________________________
dropout_7 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 51
=================================================================
Total params: 111,451
Trainable params: 111,451
Non-trainable params: 0
```

Một số tham số khác:

- Optimizer: Adam
- Loss: MSE - Mean Squared Error

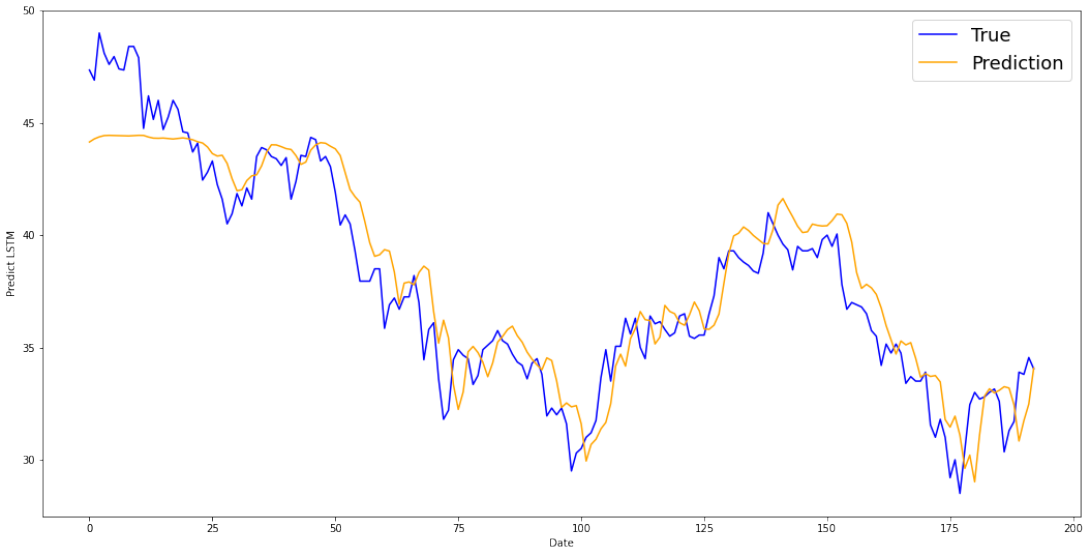
- Epochs: 100 (có Early Stopping)
- Batch size: 32

Sau khi nhóm huấn luyện hết tập train của 5 loại cổ phiếu, kết quả dự đoán trên tập test như sau:

a. Kết quả của BID với mô hình LSTM

```
============== BID ==============
MSE: 3.0530356480359426
MAPE: 0.037541819887086054
MAE: 1.398886881220526

Epoch 51: early stopping
```

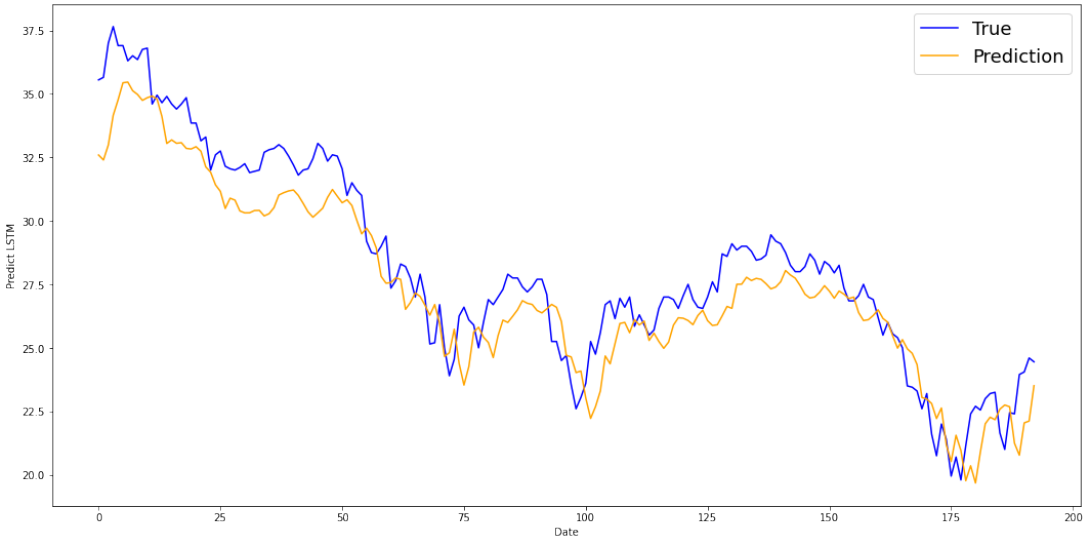


Kết quả của BID với mô hình LSTM

b. Kết quả của CTG với mô hình LSTM

```
============== CTG ==============
MSE: 2.0949027734947903
MAPE: 0.0435026613720281
MAE: 1.2302284576732259

Epoch 30: early stopping
```

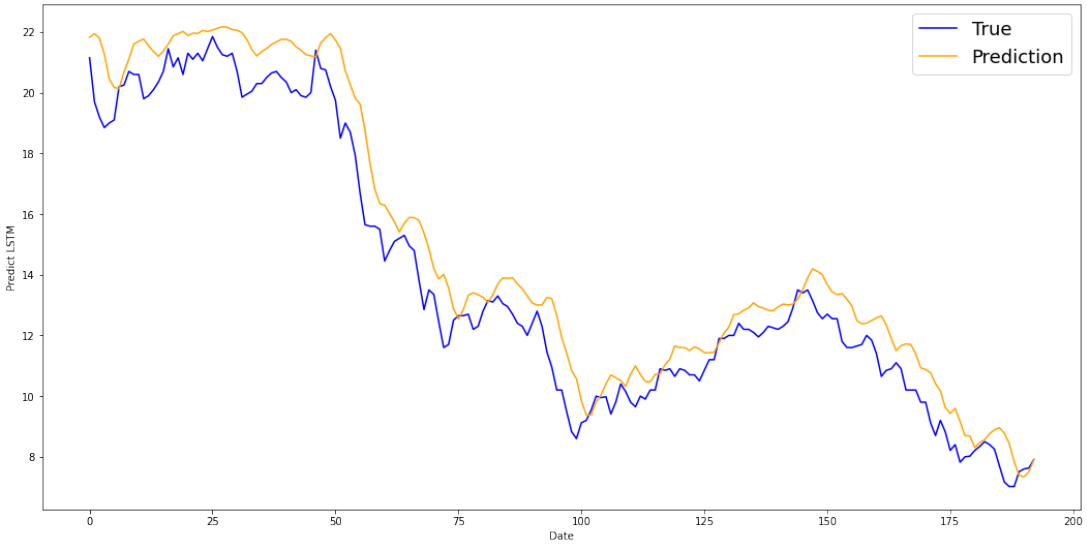


Kết quả của CTG với mô hình LSTM

c. Kết quả của AGR với mô hình LSTM

```
============== AGR ==============
MSE: 1.4020901128544727
MAPE: 0.07598287522446184
MAE: 0.9905833262112475

Epoch 16: early stopping
```

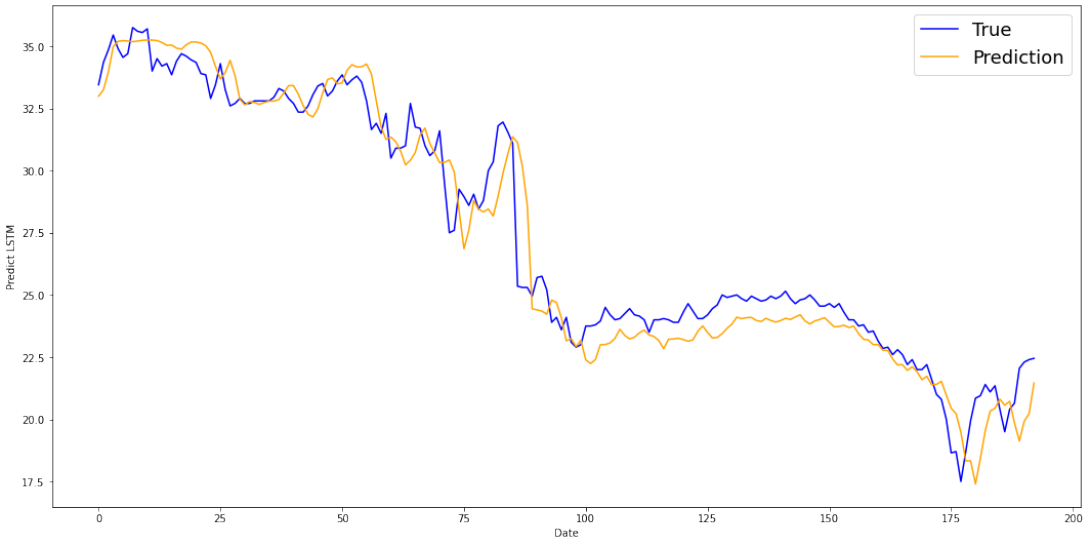


Kết quả của AGR với mô hình LSTM

d. Kết quả của ACB với mô hình LSTM

```
============== ACB ==============
MSE: 1.420404601069918
MAPE: 0.034315722893661575
MAE: 0.8928547043874473
```

```
Epoch 27: early stopping
```

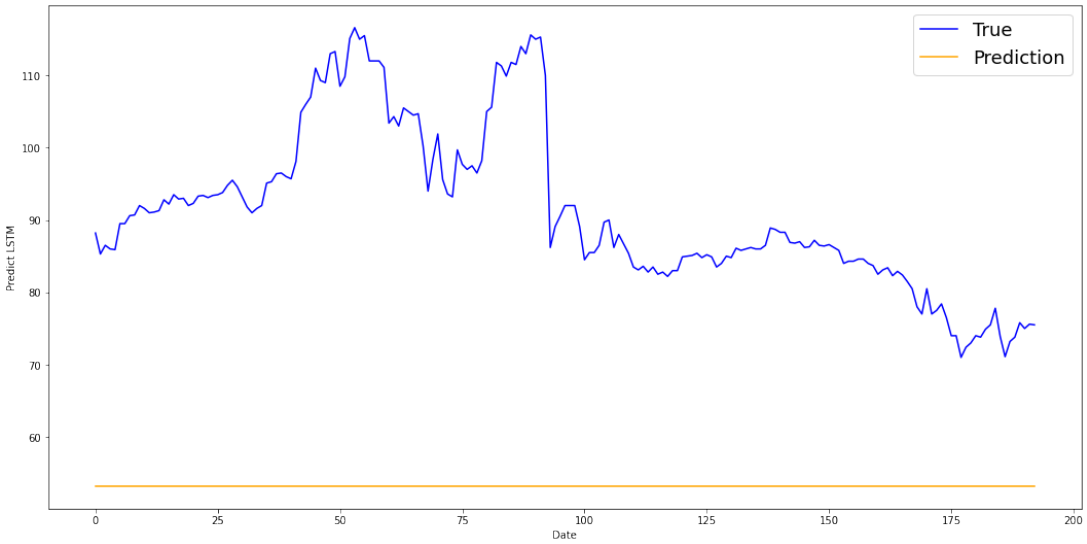


Kết quả của ACB với mô hình LSTM

e. Kết quả của FPT với mô hình LSTM

```
============== FPT ==============
MSE: 1573.2481776398047
MAPE: 0.40811842525899117
MAE: 38.03634159700858

Epoch 50: early stopping
```



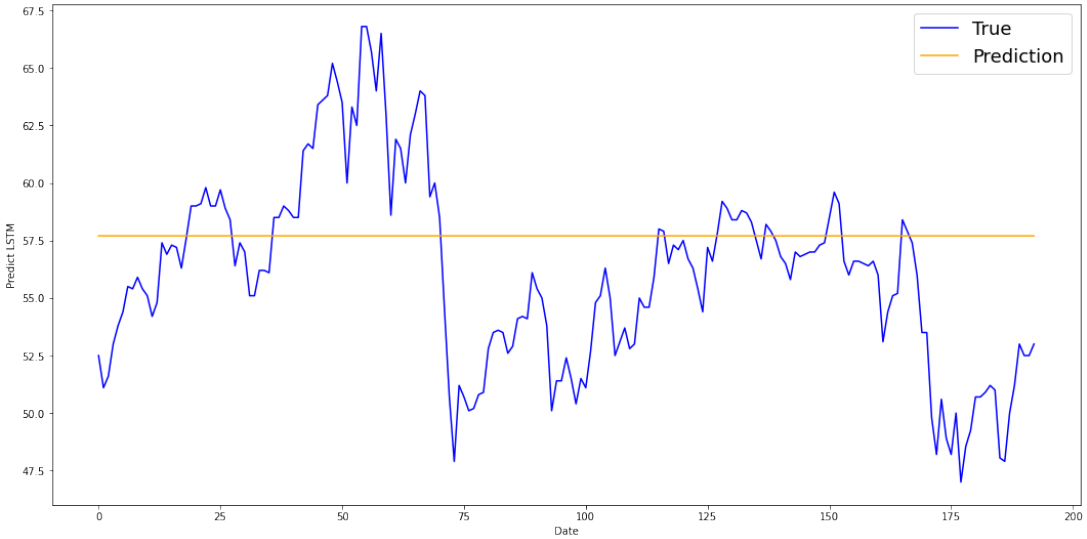
Kết quả của FPT với mô hình LSTM

f. Kết quả của BVH với mô hình LSTM

```
============== BVH ==============
```

```
MSE: 19.60270957091074
MAPE: 0.06429147143589788
MAE: 3.492934714203671

Epoch 30: early stopping
```



Kết quả của BVH với mô hình LSTM

3. Kết quả của mô hình

	LSTM		
Stocks	MSE	MAPE	MAE
ACB	1.4204	0.0343	0.8929
BID	3.0530	0.0375	1.3989
BVH	19.6027	0.0643	3.4929
CTG	2.0949	0.0435	1.2302
FPT	1573.2482	0.4081	38.0363
	319.8838	**0.1175**	**9.0102**

4. Nhận xét

- Đồ thị dự đoán rất tốt trên những cổ phiếu có giá thấp
- Đồ thị dự đoán rất tệ trên những cổ phiếu có giá cao, điều này khả năng cao là do việc chưa chuẩn hóa dữ liệu. Vì vậy việc chuẩn hóa dữ liệu rất quan trọng trong data mining, nó có thể làm một mô hình tốt cho ra kết quả cực kỳ sai lệch

D. Tổng hợp kết quả

Ở đây, nhóm tổng hợp các kết quả tốt nhất của từng mô hình đối với từng loại cổ phiếu. Bảng thống kê kết quả trung bình được trình bày ở dưới đây.

Mô hình	MSE	MAPE	MAE
EMA(baseline)	2.9135	0.0206	0.9587
ARIMA	58.9363	0.117	6.307
SVR	2.424	0.0184442	0.85362
LSTM	319.8838	0.1175	9.0102

Bảng kết quả trung bình của 5 loại cổ phiếu.

Mô hình	MSE	MAPE	MAE
EMA(baseline)	2.7125	0.0211	1.1681
ARIMA	23.5579	0.067222	3.863489
SVR	2.985	0.018941	1.0569
LSTM	19.6027	0.0643	3.4929

Bảng kết quả của BVH.

Mô hình	RMSE	MAPE	MAE
EMA(baseline)	1.1850	0.0219	0.8046
ARIMA	56.4126	0.18482	6.438573
SVR	0.969	0.020564	0.7497
LSTM	3.0530	0.0375	1.3989

Bảng kết quả của BID.

Mô hình	MSE	MAPE	MAE
EMA(baseline)	0.6615	0.0191	0.4985
ARIMA	49.9821	0.240708	5.693627
SVR	0.5135	0.016484	0.4347
LSTM	1.4204	0.0343	0.8929

Bảng kết quả trung bình của ACB.

Mô hình	RMSE	MAPE	MAE
EMA(baseline)	0.6313	0.0225	0.5996
ARIMA	61.0393	0.266443	6.860772
SVR	0.4982	0.019964	0.5247
LSTM	2.0949	0.0435	1.2302

Bảng kết quả trung bình của CTG.

Mô hình	MSE	MAPE	MAE
EMA(baseline)	9.3772	0.0186	1.7228
ARIMA	129.7215	0.092189	8.677041
SVR	7.1543	0.016268	1.5021
LSTM	1573.2482	0.4081	38.0363

Bảng kết quả của FPT.

Dựa vào bảng kết quả, ta thấy mô hình EMA (baseline) tuy đơn giản nhưng lại có sai số thấp hơn so với các mô hình phức tạp hơn như ARIMA và LSTM. Mô hình SVR tuy có tốt hơn nhưng lại tốt hơn không nhiều. Việc này có thể là do nhóm thiết lập thí nghiệm chỉ dự đoán 1 ngày kế tiếp nên không thể thấy rõ được sức mạnh của các mô hình machine learning và deep learning. Hơn nữa, ở LSTM, nhóm chưa thực hiện chuẩn hóa dữ liệu nên việc này có thể là một trong những nguyên nhân khiến mô hình không học tốt được.

E. Kết luận

Như vậy, trong bài tập lớn này, nhóm đã nghiên cứu về 4 mô hình cho việc dự đoán time series gồm Exponential Moving Average (EMA) (baseline), ARIMA, SVR và LSTM. Trong đó, tuy EMA là một kỹ thuật cổ điển nhưng lại là 1 baseline rất tốt ít nhất trong bài tập lớn này. ARIMA, tuy có phức tạp hơn, nhưng do yêu cầu về dữ liệu phải ổn định (stationary) nên có thể là 1 nguyên nhân dẫn đến mô hình không tốt. Về SVR, tuy là 1 kỹ thuật machine learning, nhưng lại hiệu quả hơn so với LSTM, một mô hình deep learning được nhóm kỳ vọng sẽ đạt kết quả tốt nhất.
Trong tương lai, nhóm sẽ thực hiện chuẩn hóa và cho các mô hình deep learning như LSTM nhiều dữ liệu hơn thay vì chỉ sử dụng 1 mô hình cho 1 loại cổ phiếu. Đồng thời, nhóm sẽ tìm hiểu và áp dụng các mô hình, kỹ thuật khác trong bài toán dự đoán trong time series như Transformer, 1D Dilated Causal Convolution Network,...

Tài liệu tham khảo

Tài liệu

[1] Machine learning cơ bản, *Kernel Support Vector Machine*
https://machinelearningcoban.com/2017/04/22/kernelsmv/

[2] Paul Paisitkriangkrai, *Linear Regression and Support Vector Regression*
https://cs.adelaide.edu.au/~chhshen/teaching/ML_SVR.pdf

[3] Hyndman, R.J., Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on 2022-08-11.

[4] Thushan Ganegedara, *LStock Market Predictions with LSTM in Python*
https://www.datacamp.com/tutorial/lstm-python-stock-market

[5] Jordi Corbilla , *Stock prediction using deep neural learning*
https://jordicorbilla.github.io/stock-prediction-deep-neural-learning/