ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
****

BÁO CÁO BÀI TẬP LỚN

Học phần: Nhập môn trí tuệ nhân tạo (IT3160)
Đề tài: *Bài toán dự đoán giá cổ phiếu sử dụng Linear Regression và LSTM*

Giảng viên hướng dẫn : TS.Trần Thế Hùng
Mã lớp : 147728
Nhóm : 16
Nhóm sinh viên thực hiện :

1. Nguyễn Tuấn Thành 20210800
2. Nguyễn Quang Trung 20215155
3. Hà Quỳnh Trang 20210852
4. Đỗ Văn Bình 20210103
5. Lại Thanh Xuân 20216173

MỤC LỤC

1. Lời mở đầu

1.1. Lý do chọn đề tài

Ngày nay, tất cả các quốc gia phát triển và hầu hết các nước đang phát triển đều có thị trường chứng khoán, một thị trường không thể thiếu với mọi nền kinh tế muốn phát triển vững mạnh. Ở Việt Nam, dù đã trải qua 20 năm hình thành và phát triển kể từ khi Trung tâm Giao dịch Chứng khoán TP Hồ Chí Minh (sau này được đổi tên thành Sở Giao dịch Chứng khoán TP Hồ Chí Minh - HOSE) có phiên giao dịch đầu tiên ngày 28-7-2000, đến nay ngành Chứng khoán đã đạt được những thành tựu nhất định cùng với những chuyển mình ngày càng lớn mạnh của nền kinh tế đất nước.

Chính vì lẽ đó dự đoán thị trường chứng khoán là một nhu cầu cấp thiết và có ý nghĩa thực tiễn. Chủ đề này đã được nhiều nhà nghiên cứu trong và ngoài nước quan tâm và đưa ra nhiều giải pháp. Mỗi giải pháp có những ưu nhược điểm khác nhau, tuy nhiên sử dụng học máy là giải pháp mang lại kết quả tốt. Do đó em đã lựa chọn đề tài "**Dự đoán giá cổ phiếu bằng phương pháp học máy Linear Regression và phương pháp học sâu LSTM**".

1.2. Mục tiêu nghiên cứu

Em tập trung nghiên cứu giải quyết bài toán dự đoán giá cổ phiếu trên thị trường cổ phiếu Việt Nam ở các sàn HNX, HOSE, UPCOM với các cổ phiếu như MSN, VCB, TCB, HPG… Trên cơ sở dữ liệu thu thập được từ thư viện vnstock, em tiền xử lý dữ liệu, lựa chọn các tham số sau đó áp dụng các mô hình học máy Linear Regression và mô hình học sâu LSTM để dự đoán giá cổ phiếu từ đó chọn được ra mô hình tối ưu.

1.3. Đối tượng, phạm vi nghiên cứu

- Đối tượng nghiên cứu của em là các dữ liệu cổ phiếu lấy được thông qua thư viện vnstock.
- Phạm vi nghiên cứu: : Các cổ phiếu có chuỗi ngày giao dịch trong phạm vi rộng lớn áp dụng các phương pháp học máy, học sâu cho bài toán dự đoán giá cổ phiếu.

1.4. Phương pháp nghiên cứu

- Phương pháp nghiên cứu lý thuyết: Tổng hợp, nghiên cứu các tài liệu về cổ phiếu chứng khoán; nghiên cứu các phương pháp, thuật toán sử dụng cho dự đoán giá cổ phiếu; nghiên cứu các phương pháp học sâu vào thị trường cổ phiếu. Tìm hiểu các kiến thức liên quan như thị trường chứng khoán, học máy, kỹ thuật lập trình trên máy tính.

- Phương pháp nghiên cứu thực nghiệm: Sau khi nghiên cứu lý thuyết, phát biểu bài toán, đề xuất mô hình; xây dựng và phát triển ứng dụng

dựa trên mô hình đề xuất; cài đặt thử nghiệm chương trình, đánh giá các kết quả đạt được.

- Phương pháp so sánh và đánh giá: phân tích đánh giá mô hình đề xuất với nhau.

2. Cơ sở lý thuyết

2.1. Học máy, học sâu

2.1.1. Học máy (Machine Learning)

Những năm gần đây, với sự bùng nổ của lĩnh vực Trí tuệ nhân tạo, học máy ngày càng được nhiều người quan tâm đến. Trước tiên học máy (Machine Learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Theo Simon (1983), học máy là "Một quá trình nhờ đó một hệ thống cải thiện hiệu suất (hiệu quả hoạt động) của nó". Các thuật toán học máy xây dựng một mô hình dựa trên dữ liệu mẫu, được gọi là dữ liệu huấn luyện, để đưa ra dự đoán hoặc quyết định mà không cần được lập trình chi tiết về việc đưa ra dự đoán hoặc quyết định này.

Học một ánh xạ (hàm):

$$f : x \rightarrow y$$

- x: quan sát (dữ liệu), kinh nghiệm
- y: phán đoán, tri thức mới, kinh nghiệm mới, …
- Hồi quy (regression): nếu y là một số thực
- Phân loại (classification): nếu y thuộc một tập rời rạc (tập nhãn lớp)

Học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán NP-khó, vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được.

Học máy đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kĩ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả.

Học máy có hiện nay được áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, chẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, phân loại các chuỗi DNA, nhận dạng tiếng nói và chữ viết, dịch tự động, chơi trò chơi và cử động rô-bốt (robot locomotion).

Học máy chia làm 4 loại:

- Học có giám sát
- Học không giám sát
- Học nửa giám sát
- Học tăng cường

2.1.2. Học sâu (Deep learning)

Học sâu (Deep learning) là một phần con của học máy (machine learning). Nó vẫn liên quan đến việc dạy máy tính học từ dữ liệu, nhưng đây lại là một bước tiến mới trong quá trình phát triển của trí tuệ nhân tạo.

Học sâu được phát triển dựa trên sự hiểu biết về mạng thần kinh nhân tạo (neural networks). Ý tưởng xây dựng AI bằng mạng thần kinh đã tồn tại từ những năm 1980, nhưng cho đến năm 2012, học sâu mới thực sự trở nên phổ biến. Học sâu sử dụng một tầng các lớp đơn vị xử lý phi tuyến để trích xuất hoặc chuyển đổi các tính năng (hoặc biểu diễn) của dữ liệu. Đầu ra của một lớp phục vụ như là đầu vào của lớp kế tiếp. Deep learning tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, dịch tự động (machine translation), xử lý ngôn ngữ tự nhiên,...

Tương tự học máy có khả năng hiện thực hóa nhờ vào lượng lớn dữ liệu con người tạo ra, học sâu cũng mang sức mạnh tính toán với chi phí rẻ hơn, được ứng dụng nhiều hơn nhờ tiến bộ trong thuật toán. Ở cùng một nhiệm vụ, học sâu có thể tạo ra kết quả vượt trội hơn so với máy học.

Công nghệ học sâu đã tạo nên sự đột phá trong quá trình nhận dạng đối tượng. Sự sáng tạo này đã nhanh chóng thúc đẩy trí tuệ nhân tạo phát triển trên nhiều khía cạnh, bao gồm cả hiểu ngôn ngữ tự nhiên (NLU).

Học sâu cũng như Học máy có thể chia thành 4 nhóm chính:

- Học sâu không giám sát
- Học sâu có giám sát
- Học sâu bán giám sát
- Học sâu tăng cường

2.2.Học có giám sát và Học không giám sát

2.2.1. Học có giám sát

Học có giám sát là một kĩ thuật của ngành học máy để xây dựng một hàm (function) từ dữ liệu huấn luyện. Dữ liệu huấn luyện bao gồm các cặp gồm đối tượng đầu vào (thường dạng vec-tơ), và đầu ra mong muốn. Đầu ra của một hàm có thể là một giá trị liên tục (gọi là hồi quy), hay có thể là dự đoán một nhãn phân loại cho một đối tượng đầu vào (gọi là phân loại). Nhiệm vụ của chương trình học có giám sát là dự đoán giá trị của hàm cho một đối tượng bất kì là đầu vào hợp lệ, sau khi đã xem xét một số ví dụ huấn luyện (nghĩa là, các cặp đầu vào và đầu ra tương ứng).

Nó được gọi là việc học có giám sát bởi vì quá trình của thuật toán học từ tập dữ liệu đầu vào có thể được coi là một "giáo viên" giám sát quá trình học tập. Chúng ta biết câu trả lời đúng, thuật toán sẽ lặp đi lặp lại làm cho việc dự đoán về dữ liệu đầu vào liên tục được "giáo viên" hoàn thiện. Việc học dừng lại khi thuật toán đạt được mức hiệu suất ở mức chấp nhận được.

Một số ví dụ phổ biến của thuật toán học máy được giám sát là:

- Hồi quy tuyến tính cho các vấn đề hồi quy.
- Nguyên lý "Khu rừng ngẫu nhiên" cho việc phân loại và hồi quy.
- Hỗ trợ các hệ máy vector cho các vấn đề về phân loại.

2.2.2. Học không giám sát

Học không có giám sát là một phương pháp của ngành học máy nhằm tìm ra một mô hình mà phù hợp với các quan sát. Nó khác biệt với học có giám sát ở chỗ là đầu ra đúng tương ứng cho mỗi đầu vào là không biết trước. Trong học không có giám sát, một tập dữ liệu đầu vào được thu thập. Học không có giám sát thường đối xử với các đối tượng đầu vào như là một tập các biến ngẫu nhiên. Sau đó, một mô hình mật độ kết hợp sẽ được xây dựng cho tập dữ liệu đó.

Mục tiêu của việc học không giám sát là để mô hình hóa cấu trúc nền tảng hoặc sự phân bố trong dữ liệu để hiểu rõ hơn về nó.

Đây được gọi là học tập không giám sát vì không giống như việc học có giám sát ở trên, không có câu trả lời đúng và không có vị "giáo viên" nào cả. Các thuật toán được tạo ra chỉ để khám phá và thể hiện các cấu trúc hữu ích bên trong dữ liệu.

Các vấn đề học tập không giám sát có thể được phân ra thành hai việc chia nhóm và kết hợp.

- Chia nhóm: Vấn đề về chia nhóm là nơi bạn muốn khám phá các nhóm vốn có bên trong dữ liệu, chẳng hạn như phân nhóm khách hàng theo hành vi mua hàng.
- Kết hợp: Vấn đề về học tập quy tắc kết hợp là nơi bạn muốn khám phá các quy tắc mô tả dữ liệu của bạn, chẳng hạn như những người mua X cũng có khuynh hướng mua Y

Học không có giám sát có thể được dùng kết hợp với suy luận Bayes để cho ra xác suất có điều kiện (nghĩa là học có giám sát) cho bất kỳ biến ngẫu nhiên nào khi biết trước các biến khác.

Học không có giám sát cũng hữu ích cho việc nén dữ liệu: về cơ bản, mọi giải thuật nén dữ liệu hoặc là dựa vào một phân bố xác suất trên một tập đầu vào một cách tường minh hay không tường minh.

Một số ví dụ phổ biến của thuật toán học không giám sát là:

- Xây dựng tham số "k-mean" cho vấn đề chia nhóm.
- Thuật toán Apriori cho các vấn đề liên quan đến việc học tập quy tắc.

2.3. Cơ sở lý thuyết Linear Regression

Linear Regression là một phương pháp thống kê được sử dụng để mô hình hóa mối quan hệ giữa một biến phụ thuộc (Y) và một hoặc nhiều biến độc lập (X). Mô hình cho rằng mối quan hệ giữa Y và X là tuyến tính, có nghĩa là

biểu diễn thành một đường thẳng.

Trong mô hình Linear Regression, Biến phụ thuộc (Y) là đại lượng mà chúng ta muốn dự đoán, còn biến độc lập (X) là đại lượng được sử dụng để dự đoán (Y). Mối quan hệ tuyến tính giữa Y và X được biểu thị bằng công thức:

$$Y = \beta 0 + \beta 1 X + \epsilon$$

Trong đó:

Y là biến phụ thuộc

X là biến độc lập

β0 là hằng số

β1 là hệ số hồi quy

ϵ là lỗi

Trong trường hợp hồi quy tuyến tính đa biến (multiple linear regression), phương trình tổng quát có dạng:

$$y = \beta 0 + \beta 1 x1 + \beta 2 x2 + \ldots + \beta nxn + \epsilon$$

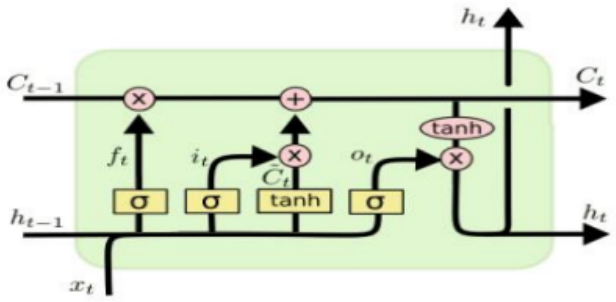
Phương pháp phổ biến nhất để ước lượng tham số β0 và β1 là phương pháp bình phương tối thiểu (Ordinary Least Squares - OLS). Phương pháp nay tìm các giá trị của các hệ số hồi quy sao cho tổng bình phương sai số giữa giá trị thực tế và giá trị dự đoán nhỏ nhất.

2.4. Cơ sở lý thuyết LSTM (Long-short term memory)

LSTM(Long-short Term Memory) là một loại mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) được thiết kế để xử lý và phân tích các chuỗi dữ liệu có sự phụ thuộc dài hạn. LSTM được giới thiệu lần đầu bởi Hochreiter và Schmidhuber vào năm 1997. Kiến trúc này đã được phổ biến và sử dụng rộng rãi cho tới ngày nay.

LSTM đã tỏ ra khắc phục được rất nhiều những hạn chế của RNN trước đây về triệt tiêu đạo hàm. Tuy nhiên cấu trúc của chúng có phần phức tạp hơn mặc dù vẫn dữ được tư tưởng chính của RNN là sự sao chép các kiến trúc theo dạng chuỗi.

Thay vì chỉ có một tầng đơn như RNN thì LSTM có tới 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác với nhau theo một cấu trúc đặc biệt.



Hình : Mô hình của một tế bào LSTM

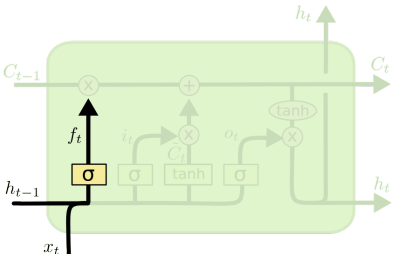
Một tế bào LSTM gồm 4 tầng khác nhau.
Xét tại thời điểm t,

h_t thể hiện kết quả
X_t ẩn, thể hiện tín hiệu vào của dữ liệu
C_t là đầu ra của mạng LSTM

Ý tưởng chính của mạng LSTM đó là : Với mỗi thời điểm t , ta sẽ có một trạng thái của ô LSTM (1 cell) là tương ứng. Thể hiện trên hình đó σlà đường thẳng chạy ngang từ C_{t-1} tới C_t , ứng với việc ta sẽ truyền kết quả từ trạng thái trước đến trạng thái sau. Tuy nhiên điều đó không có nghĩa là toàn bộ thông tin đều đi mà không bị gì cả. Tương tác với các giá trị C_{t-1}, ta sẽ có các cổng (như hình có các hàm kích hoạt sigmoid với kí hiệu σ, và hàm tanh) và các phép toán trên ma trận (×,+).

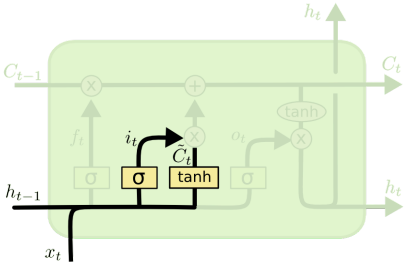
Các giai đoạn bên trong một tế bào LSTM :

Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là "tầng cổng quên" (forget gate layer). Nó sẽ lấy đầu vào là h_{t-1}và *xt* rồi đưa ra kết quả là một số trong khoảng [0,1] cho mỗi số trong trạng thái tế bào C_{t-1}.Đầu ra là 1thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

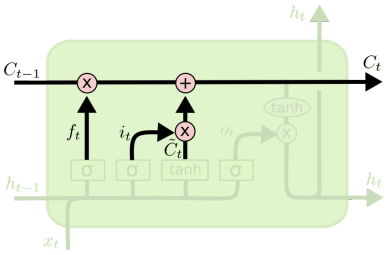
Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là "tầng cổng vào" (input gate layer) để quyết định giá trị nào ta sẽ cập nhập. Tiếp theo là một tầng *tanh* tạo ra một véc-tơ cho giá trị mới *Ct~* nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

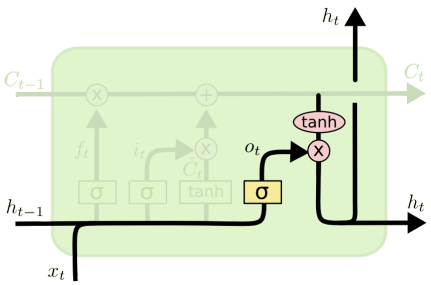
Giờ là lúc cập nhập trạng thái tế bào cũ C_{t-1} thành trạng thái mới C_t .Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện là xong.

Ta sẽ nhân trạng thái cũ với f_t để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm i_t*Ct. Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao



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

Cuối cùng, ta cần quyết định xem ta muốn đầu ra là gì. Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm *tanh* để có giá trị nó về khoảng [−1,1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

3. Bài toán dự đoán giá cổ phiếu

3.1 Các tham số sử dụng

3.1.1. Tham số các chỉ báo kỹ thuật về động lượng và xu hướng

Chỉ báo kỹ thuật SMA (Simple Moving Average) là đường trung bình động đơn giản, là đường nối tất cả mức giá đóng cửa trung bình trong N chu kỳ của một tài sản, với N được chọn trước. Đường trung bình động đơn giản (SMA) là một cách giúp lọc nhiễu và làm dịu những biến động giá phức tạp trở nên mượt hơn giúp thầy quan sát xu hướng thị trường tốt hơn, chỉ báo chậm theo xu hướng bởi vì nó dựa trên giá cả trong quá khứ. Bằng cách nhìn vào độ dốc của đường trung bình, thầy có thể xác định tốt hơn hướng đi tiềm năng của giá cả thị trường. Công thức tính SMA:

$$SMA = \frac{A1 + A_2 + A_3 + \dots + A_n}{n}$$

Trong đó:
An = giá đóng cửa của tài sản tại thời điểm giao dịch n
n = số phiên giao dịch

Chỉ báo kỹ thuật RSI (Relative Strength Index) là chỉ báo giúp đo lường mức độ thay đổi giá. Từ đó đánh giá các điều kiện mua vượt mức hoặc bán vượt mức của giá cổ phiếu hoặc tài sản khác. Chỉ số RSI đo lường sức mạnh tương đối của giá chứng khoán với các mức giá trong lịch sử của chính mã chứng khoán đó. Công thức tính RSI:

$$RSI = 100 - \left[\frac{100}{1 + \frac{Mức\ tăng\ trung\ bình}{Mức\ giảm\ trung\ bình}}\right]$$

Trong đó:
RSI thường được tính dựa vào giá đóng cửa 14 ngày gần nhất.
Mức tăng trung bình hay mức giảm trung bình là phần trăm lãi lỗ trung bình trong một khoảng thời gian.

Chỉ báo kỹ thuật MACD (Moving Average Convergence Divergence) là chỉ báo giúp cung cấp các biến động của thị trường, hỗ trợ người dùng xác định tín hiệu mua bán của thị trường. Để xác định đường MACD, dùng cần dựa vào độ chênh lệch của hai đường trung bình động (EMA) 12 ngày và 26 ngày. Chỉ báo MACD được cấu tạo từ bốn thành phần chính là đường MACD, đường tín hiệu, biểu đồ và đường zero. Mỗi thành phần lại mang đặc điểm và ý nghĩa khác nhau.

- Đường MACD có vai trò xác định xu hướng giá của thị trường, giá trị của nó được tính bằng hiệu số của hai đường trung bình hàm mũ EMA (12) và EMA (26).
- Đường tín hiệu Signal cũng chính là đường EMA (9) của đường MACD. Khi hai đường này phối hợp cùng nhau là lúc chúng dự báo một xu hướng đảo chiều sắp diễn ra và các người dùng nên tận dụng thời điểm này để thực hiện giao dịch một cách có lợi nhất.
- Biểu đồ Histogram là biểu đồ thể hiện sự phân kỳ và hội tụ nhờ xác định độ chênh lệch giữa đường MACD và đường tín hiệu.
- Đường Zero đóng vai trò là đường tham chiếu giúp người dùng đánh giá xu hướng thị trường mạnh hay yếu.

$$MACD = EMA(12) - EMA(26)$$

3.1.2. Tham số là các chỉ báo kỹ thuật về động lượng và stochastic

Chỉ báo kỹ thuật Stochastic là chỉ báo đánh giá động lượng của giá tài sản cũng như sức mạnh tổng thể của xu hướng phổ biến. Chỉ báo Stochastic Oscillator sẽ cho chúng ta thấy thông tin về động lượng và cường độ xu hướng. Trong các thị trường có xu hướng, Chỉ báo Stochastic có thể cảnh báo về khả năng thoái lui hoặc thậm chí là đảo chiều; và trong nhiều thị trường khác nhau, chỉ báo có thể cho biết khi nào sức mạnh của xu hướng cơ bản đang giảm dần. Chỉ báo Stochastic được biểu thị bằng 2 dòng được cấu tạo từ đường dao động %K và % D.

- **Đường %K**(màu xanh) là đường dao động chính được Lane đặt tên Stochastics vì khá gần với phạm vi giá đang xét.
- **Đường %D**(màu cam) là đường trung bình động được tính toán theo SMA3 của đường %K. Do vậy, đường %D sẽ có độ trễ đáng kể so với đường %K.
- **Đường biên:**Các đường biên mặc định là 20 (đường biên phía dưới) và 80 (đường biên phía trên).

$$\%K = \left[\frac{C - L14}{H14 - L14}\right] \times 100$$

$$\%D = \frac{\%K \text{ hiện tại} + \%K \text{ kỳ trước} + \%K \text{ 2 kỳ trước}}{3}$$

Trong đó:

- C = Giá đóng cửa hiện tại
- L14 = Giá thấp nhất của tài sản trong 14 kỳ gần đây
- H14 = Giá cao nhất trong cùng 14 kỳ

Chỉ báo Williams %R là một chỉ báo thống kê cho các nhà đầu tư biết liệu một cổ phiếu có bị bán quá mức hay mua quá mức hay không. Williams %R còn được là một chỉ báo động lượng nghịch đảo của chỉ báo Stochastic. Williams %R phản ánh mức giá đóng cửa tương ứng với mức cao nhất cao nhất trong một chu kì mặc định. Ngược lại, Chỉ báo Stochastic phản ánh mức độ đóng cửa so với mức giá thấp nhất.

$$\%R = \left[\frac{H14 - C}{H14 - L14}\right] \times (-100)$$

Trong đó:

- C = Giá đóng cửa hiện tại
- L14 = Giá thấp nhất của tài sản trong 14 kỳ gần đây
- H14 = Giá cao nhất trong cùng 14 kỳ

Chỉ báo StochRSI là một chỉ báo động lượng đo lường mức RSI so với mức cao-thấp của nó trong một khoảng thời gian. Nó sử dụng công thức Stochastics vào những giá trị RSI, và điều đó làm chỉ báo Stochastic RSI là một chỉ báo đặc biệt vì nó là chỉ báo của chỉ báo.

$$StochRSI = \frac{RSI\ hiện\ tại - RSI\ thấp\ nhất}{RSI\ cao\ nhất - RSI\ thấp\ nhất}$$

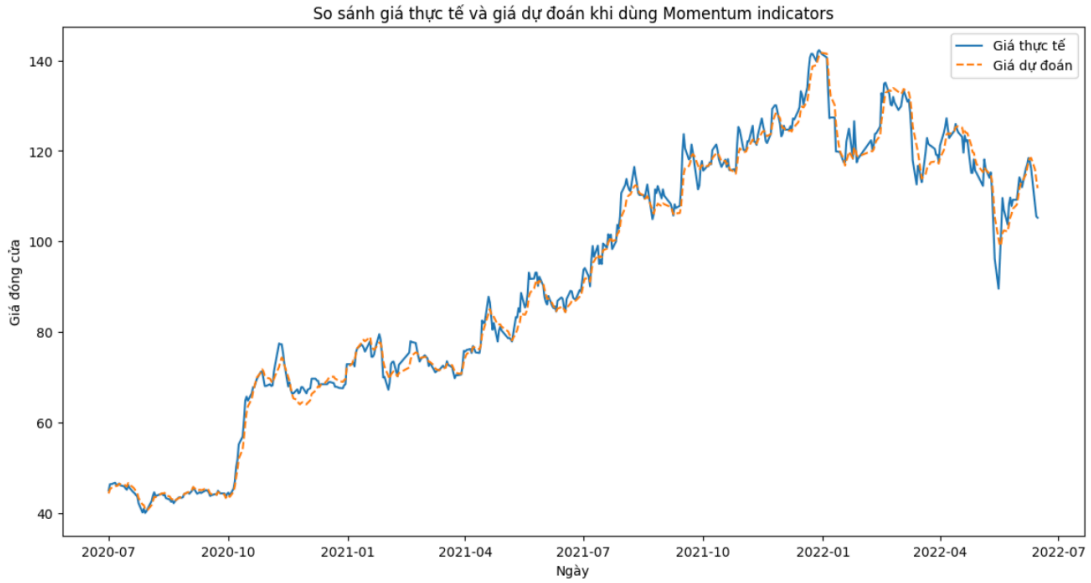
Trong đó:

- RSI hiện tại là giá trị RSI thời điểm hiện tại, chỉ báo RSI đang hiện hành.
- RSI thấp nhất là giá trị RSI thấp nhất trong một khoảng thời gian đã chọn nào đó (thường thì lấy 14 phiên giao dịch trong thời gian gần đây nhất).
- RSI cao nhất là lấy giá trị RSI cao nhất trong một khoảng thời gian đã chọn nào đó (thường thì lấy 14 phiên giao dịch gần đây nhất)

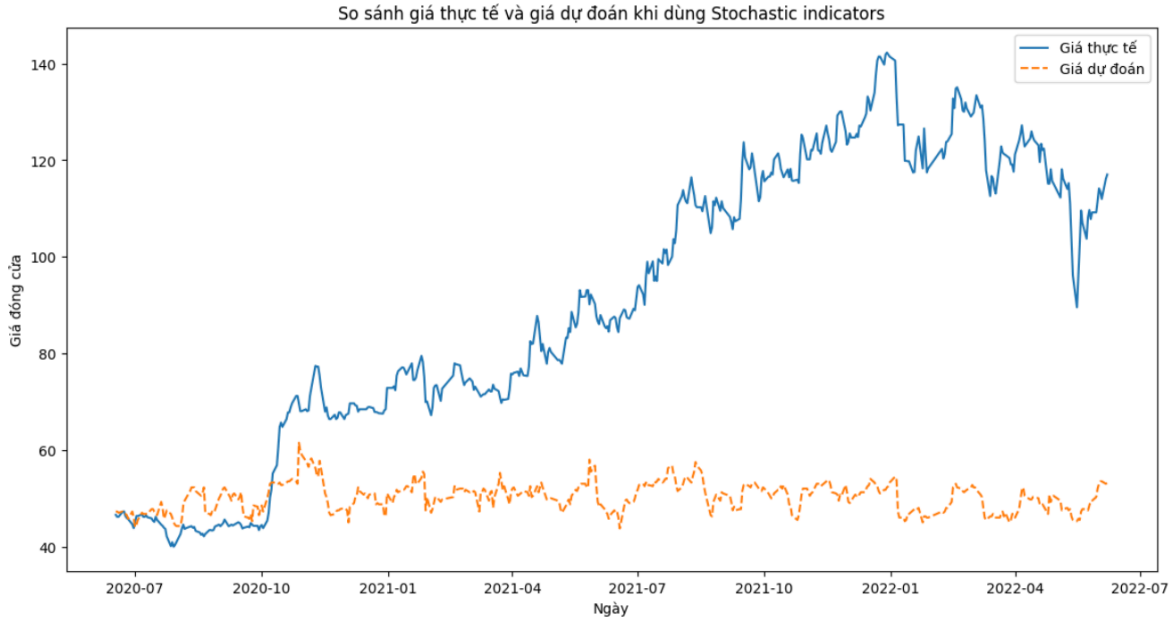
3.2. So sánh kết quả dự đoán hai mô hình với hai kiểu tham số

3.2.1. Mô hình Linear Regression

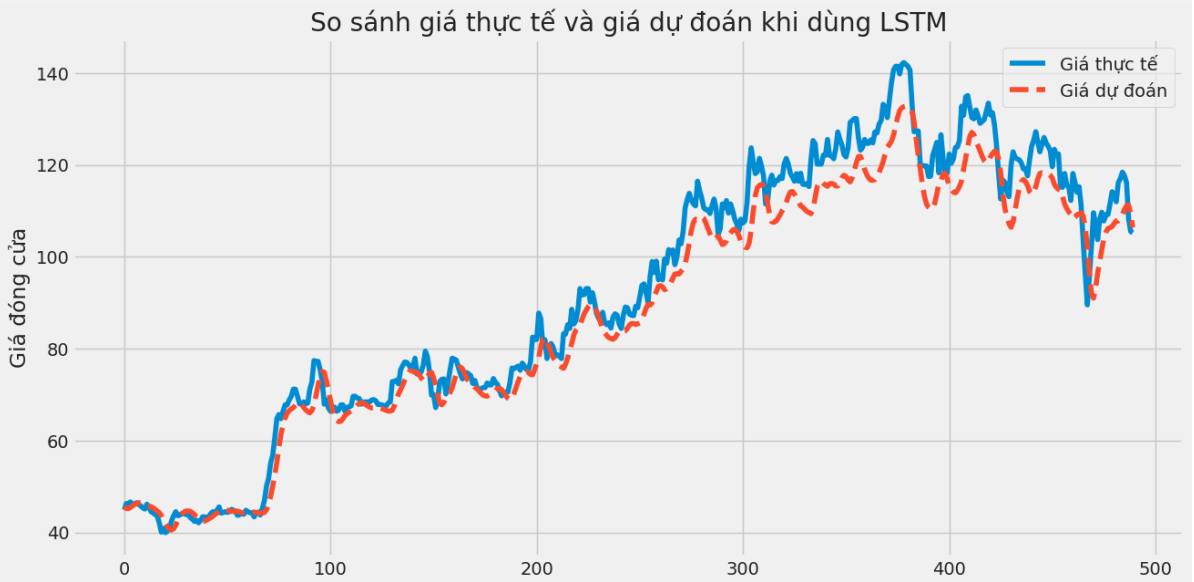
Với tham số là các chỉ báo momentum và xu hướng, ta có kết quả dự đoán giá cổ phiếu MSN trong giai đoạn từ 31/05/2014 đến 31/05/2024.



Với tham số là các chỉ báo momentum và stochastic, ta có kết quả dự đoán giá cổ phiếu MSN trong giai đoạn từ 31/05/2014 đến 31/05/2024.



3.2.2. Mô hình LSTM



4. Thu thập và phân tích dữ liệu

4.1. Thu thập dữ liệu

Thu thập dữ liệu cổ phiếu MSN bằng thư viện vnstock:

- Tải thư viện vnstock

```
!pip install -U vnstock
```

- Tải dữ liệu cổ phiếu MSN

```
data = stock_historical_data(symbol="MSN", start_date="2014-05-31", end_date="2024-05-31", resolution="1D", type="stock", beautify=True, decor=False, source='DNSE')
data
```

	time	open	high	low	close	volume	ticker
0	2014-06-02	49780	51070	49780	50820	229020	MSN
1	2014-06-03	50560	51330	50560	51330	184340	MSN
2	2014-06-04	51590	51590	50560	51070	51080	MSN
3	2014-06-05	50820	50820	50040	50040	76150	MSN
4	2014-06-06	50820	50820	49780	50300	22360	MSN
...	...	...	...	...	...	...	...
2494	2024-05-27	73500	74000	72700	73500	3881500	MSN
2495	2024-05-28	74200	75500	73800	75500	5100400	MSN
2496	2024-05-29	75700	76900	75000	75000	8003700	MSN
2497	2024-05-30	74300	77400	74100	77200	11161400	MSN
2498	2024-05-31	78000	78400	76600	76600	5592200	MSN

2499 rows × 7 columns

4.2. Phân tích dữ liệu

- Kiểm tra loại dữ liệu và kiểm tra tập dữ liệu có giá trị null hay không.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2499 entries, 0 to 2498
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   time    2499 non-null   object
 1   open    2499 non-null   int64 
 2   high    2499 non-null   int64 
 3   low     2499 non-null   int64 
 4   close   2499 non-null   int64 
 5   volume  2499 non-null   int64 
 6   ticker  2499 non-null   object
dtypes: int64(5), object(2)
memory usage: 136.8+ KB
```

- Các thông số cơ bản của dữ liệu

```
data.describe()
```

	open	high	low	close	volume
count	2499.000000	2499.000000	2499.000000	2499.000000	2.499000e+03
mean	64124.485394	65068.852341	63194.632653	64142.679872	1.069136e+06
std	26133.111754	26665.111799	25634.399318	26120.396063	1.305262e+06
min	30440.000000	30950.000000	30180.000000	30690.000000	1.450000e+03
25%	41530.000000	42050.000000	41020.000000	41530.000000	3.235200e+05
50%	63490.000000	64379.000000	62700.000000	63600.000000	6.775400e+05
75%	76930.000000	77900.000000	75595.000000	76820.000000	1.308100e+06
max	142280.000000	144760.000000	140220.000000	142280.000000	1.299960e+07

- Biểu đồ giá đóng cửa của cổ phiếu MSN

- Chèn thêm các giá trị tham số vào trong dữ liệu

```
data1 = data.copy()
data1 = data1.set_index('time')
# Tính toán các chỉ báo kỹ thuật
data1['SMA_50'] = data1['close'].rolling(window=50).mean()  # Simple Moving Average (50 ngày)
data1['RSI'] = ta.momentum.RSIIndicator(data1['close'], window=14).rsi()  # Relative Strength Index (14 ngày)
macd = ta.trend.MACD(data1['close'])  # MACD
data1['MACD'] = macd.macd()
data1['MACD_Signal'] = macd.macd_signal()

# Xóa các hàng chứa giá trị NaN
data1.dropna(inplace=True)

# Hiển thị dữ liệu
data1
```

time	open	high	low	close	volume	ticker	SMA_50	RSI	MACD	MACD_Signal
2014-08-08	45150	45920	44890	45660	207960	MSN	47895.8	40.330608	-707.268189	-655.459539
2014-08-11	45150	45660	44630	44630	416620	MSN	47772.0	33.873816	-760.341170	-676.435865
2014-08-12	44630	44890	43600	44370	522720	MSN	47632.8	32.461065	-813.998450	-703.948382
2014-08-13	44370	44630	43860	44120	198470	MSN	47493.8	31.117207	-866.704314	-736.499568
2014-08-14	44630	44630	44120	44370	317820	MSN	47380.4	34.057177	-878.178089	-764.835273
...	...	...	...	...	...	...	...	...	...	...

```
# Tính toán các chỉ báo kỹ thuật
data2 = data.copy()
data2 = data2.set_index('time')
stoch = ta.momentum.StochasticOscillator(data2['high'], data2['low'], data2['close'], window=14)
data2['Stoch_%K'] = stoch.stoch()
data2['Stoch_%D'] = stoch.stoch_signal()

data2['Williams_%R'] = ta.momentum.WilliamsRIndicator(data2['high'], data2['low'], data2['close'], lbp=14).williams_r()
data2['StochRSI'] = ta.momentum.StochRSIIndicator(data2['close'], window=14).stochrsi()

data2.dropna(inplace=True)
data2
```

time	open	high	low	close	volume	ticker	Stoch_%K	Stoch_%D	Williams_%R	StochRSI
2014-07-08	47990	47990	47470	47990	119560	MSN	53.196347	40.216496	-46.803653	1.000000
2014-07-09	47990	48500	47730	48240	245550	MSN	66.666667	53.326431	-33.333333	1.000000
2014-07-10	48240	48240	47210	47730	206000	MSN	62.621359	60.828124	-37.378641	0.752611
2014-07-11	47730	47730	46950	47210	175810	MSN	28.333333	52.540453	-71.666667	0.522392
2014-07-14	47210	47470	46950	46950	134220	MSN	13.888889	34.947860	-86.111111	0.413021
...	...	...	...	...	...	...	...	...	...	...
2024-05-27	73500	74000	72700	73500	3881500	MSN	47.222222	57.870370	-52.777778	0.193401
2024-05-28	74200	75500	73800	75500	5100400	MSN	75.000000	56.481481	-25.000000	0.510855

5. Triển khai mô hình Linear Regression

5.1. Huấn luyện mô hình

- Chia dữ liệu thành các phần train, test, prediction

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
train_size = int(len(data1) * 0.6)
test_size = int(len(data1) * 0.2)
train_data = data1[:train_size]
test_data = data1[train_size:train_size + test_size]
prediction_data = data1[train_size + test_size:]
```

- Sử dụng phương pháp Walk Forward Validation để dự đoán giá cổ phiếu, dùng các giá trị của ngày hôm trước để dự đoán cho ngày hôm sau và tiếp tục như thế cho ngày kia

```
# Walk-Forward Validation
def walk_forward_validation(train, test, features, target, prediction_data):
    predictions = []
    history = train.copy()

    for i in range(len(test)):
        model = LinearRegression()
        model.fit(history[features].dropna(), history[target].dropna())
        yhat = model.predict(test[features].iloc[i].values.reshape(1, -1))
        predictions.append(yhat[0])
        history = pd.concat([history, test.iloc[i]], ignore_index=True)

    future_predictions = []
    for i in range(len(prediction_data)):
      model = LinearRegression()
      model.fit(history[features].dropna(), history[target].dropna())
      yhat = model.predict([prediction_data[features].iloc[i]])
      future_predictions.append(yhat[0])
      new_row = prediction_data.iloc[i].copy()
      new_row['close'] = yhat[0]
      history = pd.concat([history, new_row], ignore_index=True)

    return predictions, future_predictions

# Dự đoán bằng Walk-Forward Validation
predictions, future_predictions = walk_forward_validation(train_data, test_data, features, target, prediction_data)
predictions
```

5.2. Đánh giá mô hình và kết quả dự đoán

- Sử dụng MSE để đánh giá dự đoán
  Với tham số là chỉ báo xu hướng và động lượng

```
# Đánh giá mô hình
mse = mean_squared_error(test_data[target], predictions)
print(f'Mean Squared Error: {mse}')

# Vẽ biểu đồ so sánh giá thực tế và giá dự đoán
plt.figure(figsize=(14, 7))
plt.plot(test_data.index, test_data[target], label='Giá thực tế')
plt.plot(test_data.index, predictions, label='Giá dự đoán', linestyle='--')
plt.title('So sánh giá thực tế và giá dự đoán khi dùng Momentum indicators')
plt.xlabel('Ngày')
plt.ylabel('Giá đóng cửa')
plt.legend()
plt.show()
```

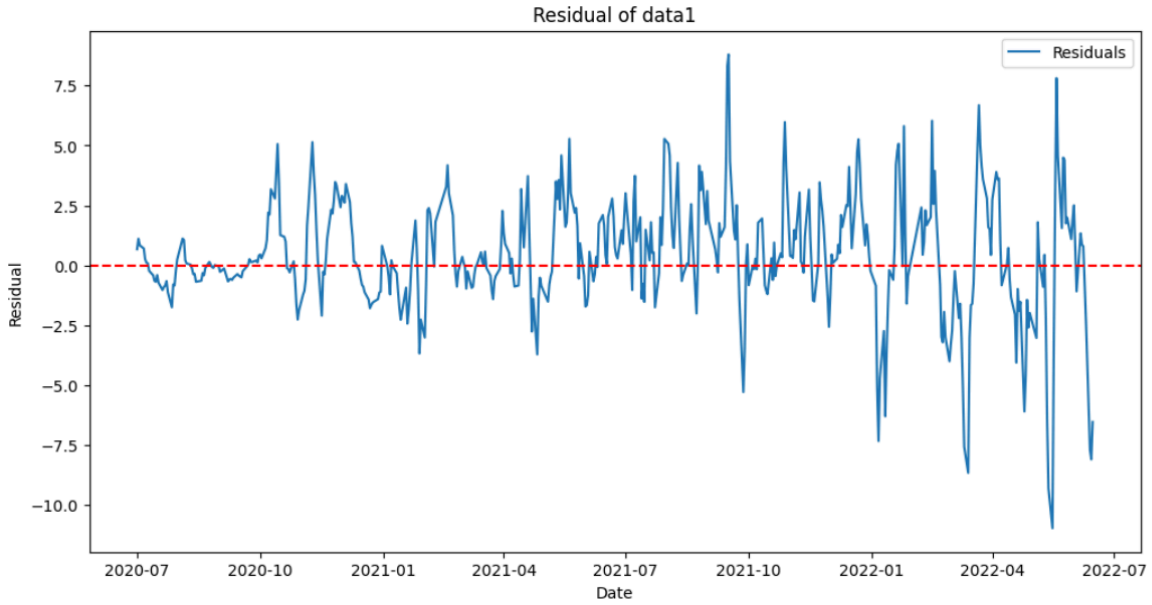
```
Mean Squared Error: 6.024720852843264
```

Với chỉ báo động lượng và stochastic

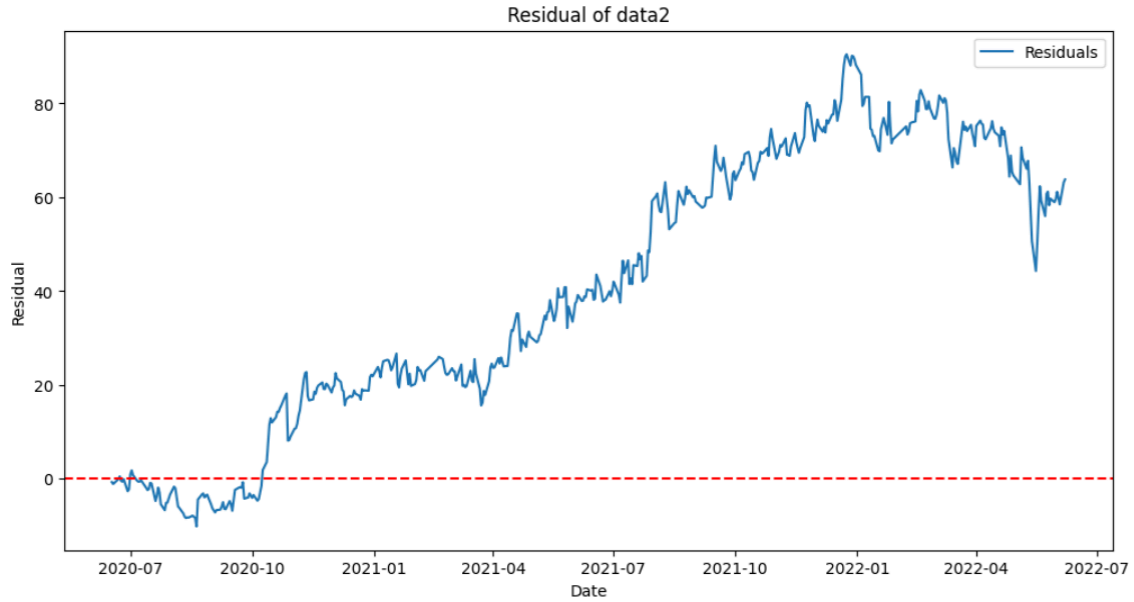
```
mse2 = mean_squared_error(test_data2[target], predictions2)
print(f'Mean Squared Error: {mse}')

# Vẽ biểu đồ so sánh giá thực tế và giá dự đoán
plt.figure(figsize=(14, 7))
plt.plot(test_data2.index, test_data2[target], label='Giá thực tế')
plt.plot(test_data2.index, predictions2, label='Giá dự đoán', linestyle='--')
plt.title('So sánh giá thực tế và giá dự đoán khi dùng Stochastic indicators')
plt.xlabel('Ngày')
plt.ylabel('Giá đóng cửa')
plt.legend()
plt.show()
```

```
Mean Squared Error: 6024757.033545478
```

- Đánh giá residual của giá dự đoán
  Với tham số là chỉ báo xu hướng và động lượng

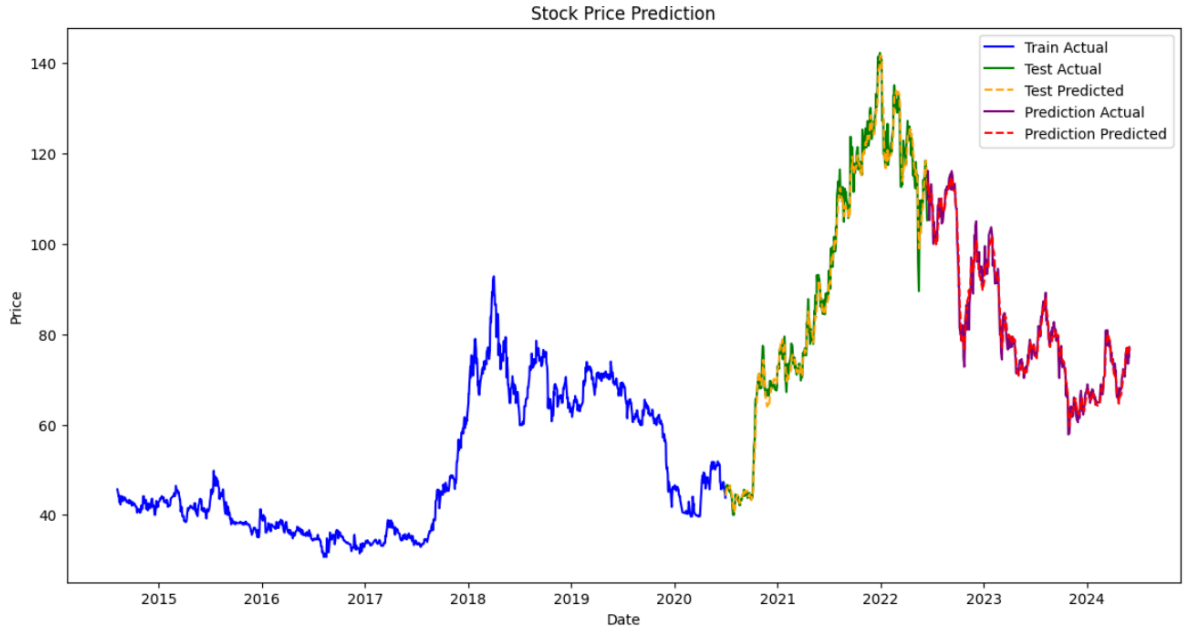


Với tham số của chỉ báo động lượng và stochastic

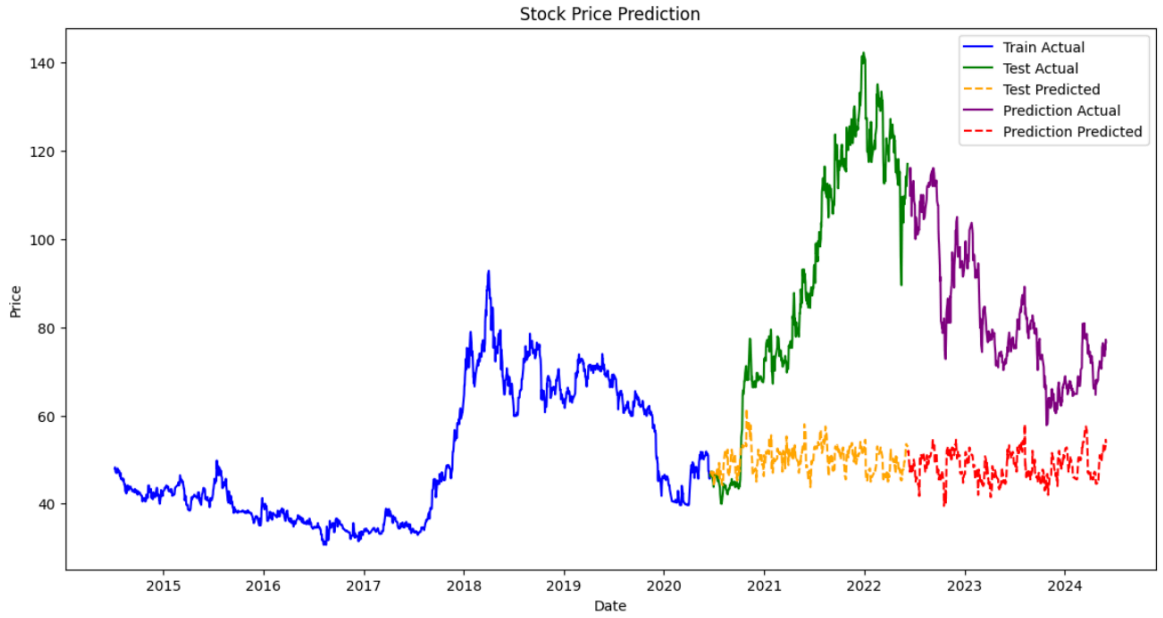


- Kết quả dự đoán

Với tham số là xu hướng và động lượng



Với tham số là động lượng và stochastic



6. Triển khai mạng Neuron Network & LSTM

6.1. Tiền xử lý

- Sử dụng chung nguồn dữ liệu đồng nhất với mô hình Linear Regression
- Tuy nhiên, các giá trị trong giá đóng qua các năm có những chênh lệch khá lớn, để tránh hiện tượng phương sai chênh lệch quá nhiều và giảm nguy cơ overfitting, cần scale lại dữ liệu về dải giá trị từ 0 đến 1.
- Sau đó, chia tỉ lệ giữa các tập train/test đồng nhất với Linear Regression để dễ dàng đánh giá

6.2. Kiến trúc của mạng neuron

Kiến trúc của mô hình sử dụng mạng neuron hồi quy với bộ nhớ dài-ngắn hạn kết hợp với các fully-connected layer, chi tiết của mạng được trình bày như sau:

- Mạng neuron được sử dụng trong bài toán có những thông số và kiến trúc như sau, được tổng kết qua thực thi mã nguồn sử dụng Tensorflow:

```
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape= (x_train.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
```

Kiến trúc bao gồm 4 layer:

- Layer thứ nhất: 128 cell LSTM, các đơn vị này giúp học các phụ thuộc dài hạn trong dữ liệu chuỗi thời gian
- Layer thứ hai: 64 cell LSTM, giảm số lượng cell so với lớp đầu tiên để giảm độ phức tạp và trích xuất các đặc trưng cao cấp hơn.
- Layer thứ ba: 25 unit fully-connected layer, lớp này giúp tổng hợp và trích xuất các đặc trưng từ đầu ra của lớp LSTM cuối cùng.
- Layer thứ tư: 1 unit, cho ra output cuối cùng

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 60, 128)           66560

 lstm_3 (LSTM)               (None, 64)                49408

 dense_2 (Dense)             (None, 25)                1625

 dense_3 (Dense)             (None, 1)                 26

=================================================================
Total params: 117619 (459.45 KB)
Trainable params: 117619 (459.45 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Tổng số tham số của mạng neuron: 117,619 tham số

6.3. Tối ưu Loss function với Gradient Descent

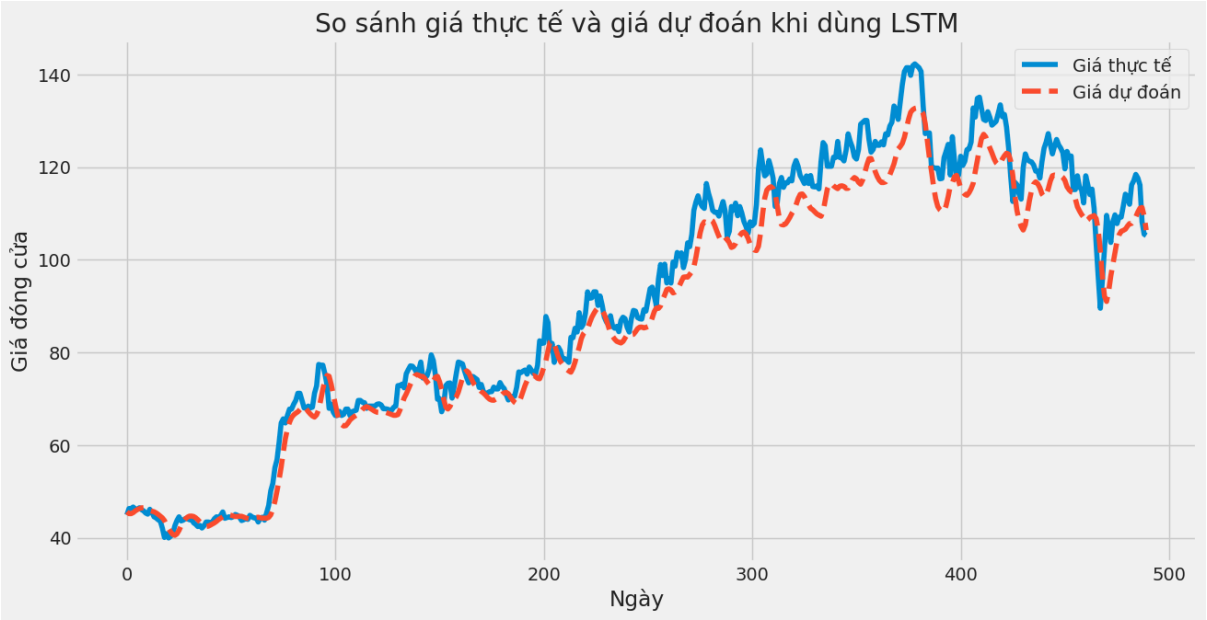
- Hàm mất mát (Loss function) của mạng sử dụng Mean Squared Error, đây là hàm mất mát phổ biến cho các bài toán hồi quy, đo lường sự khác biệt trung bình giữa các giá trị dự đoán và các giá trị thực tế đặc biệt là với bài toán dự đoán giá cổ phiếu
- Để tối ưu hàm mất mát này, sử dụng thuật toán tối ưu phổ biến Adam, sử dụng các giá trị trung bình di động của các gradient bậc nhất và bậc hai để điều chỉnh tốc độ học (learning rate)
- Mã nguồn thực thi:

```
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model.fit(x_train, y_train, batch_size=1, epochs=1)
```

```
1410/1410 [==============================] - 84s 56ms/step - loss: 8.5904e-04
<keras.src.callbacks.History at 0x7c6d13699150>
```

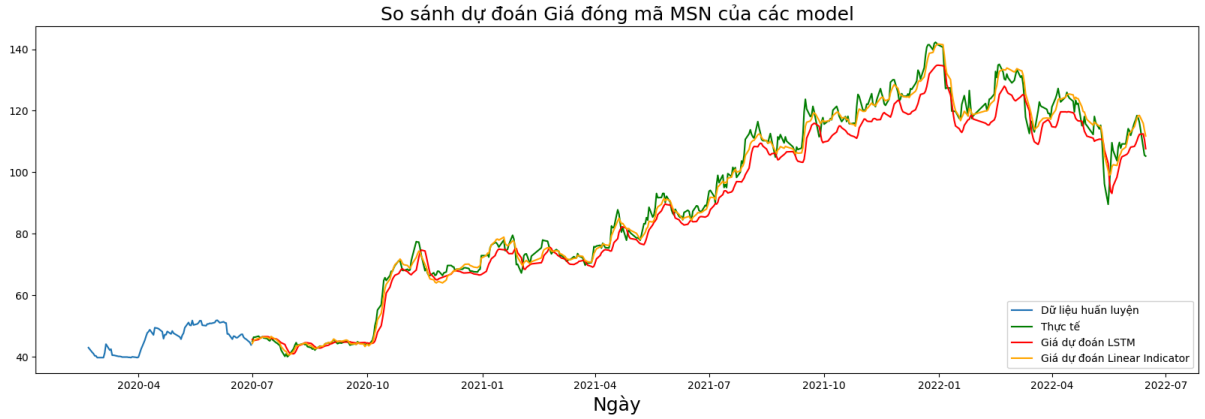
Kết quả của mô hình

Kết quả đầu ra của mô hình có thể dự đoán khá đúng xu hướng biến động của giá đóng trong phạm vi bài toán


So sánh giá thực tế và giá dự đoán khi dùng LSTM
Giá thực tế
Giá dự đoán
Giá đóng cửa
Ngày
140
120
100
80
60
40
0
100
200
300
400
500

7. So sánh kết quả hai mô hình

Kết quả của hai mô hình được biểu diễn ở trong cùng một biểu đồ như hình dưới đây:



Phân tích kết quả với lỗi trung bình, ta được kết quả như sau:

```
# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean(((predictions_lstm - y_test) ** 2)))
rmse_linear = np.sqrt(np.mean(((predictions - y_test) ** 2)))
print("RMSE of LSTM model: ", rmse)
print("RMSE of Linear model: ", rmse_linear)
```

```
RMSE of LSTM model:  5.103025756842733
RMSE of Linear model:  40.85791716030041
```

Ta có thể thấy đối với lỗi trung bình, mô hình mạng hồi quy sử dụng LSTM có lỗi nhỏ hơn rất nhiều so với linear, điều này chứng tỏ rằng mạng neuron hồi quy dự đoán khá tốt giá cổ phiếu trong phạm vi bài toán.

8. Đánh giá công việc

STT	Họ và tên	MSSV	Phân chia công việc	Phần trăm đóng góp
1	Nguyễn Tuấn Thành	20210800	- Xây dựng mạng Neuron network với LSTM	20%
2	Nguyễn Quang Trung	20215155	- Thu thập dữ liệu sử dụng vnstock - Xây dựng mô hình LR: Momentum Indicator	20%
3	Hà Quỳnh Trang	20210852	- Xây dựng mô hình LR: Stochastic Indicator	20%
4	Đỗ Văn Bình	20210103	- Phân tích dữ liêu: xác định và đánh dấu xu hướng	20%
5	Lại Thanh Xuân	20216173	- Phân tích dữ liêu: Tính toán và thêm các chỉ báo chứng khoán khác cho dữ liệu cổ phiếu	20%