TRƯỜNG ĐẠI HỌC KINH TẾ LUẬT

KHOA HỆ THỐNG THÔNG TIN

ĐỒ ÁN CUỐI KỲ

CHỦ ĐỀ: NGHIÊN CỨU ỨNG DỤNG DEEP LEARNING TRONG PHÂN TÍCH DỮ LIỆU CHUỖI THỜI GIAN

Môn học: Học sâu trong phân tích kinh doanh

Nhóm: Bamos (nhóm 3)

Giảng viên: Nguyễn Quang Phúc

Hồ Chí Minh, ngày 10 tháng 08 năm 2024

Thành viên

TT	Mã số sinh viên	Họ và tên	Hoàn thành
1	K214162155	Nguyễn Phúc Thịnh (Leader)	100%
2	K214160991	Nguyễn Thành Luân	100%
3	K214162153	Võ Minh Thành	100%
4	K214160990	Phạm Công Nguyễn Khôi	100%
5	K214161343	Lê Thành Tuân	100%

Lời cảm ơn

Trước hết, nhóm xin bày tỏ lòng biết ơn chân thành tới Trường Đại học Kinh tế - Luật vì đã đưa môn học Học sâu trong phân tích kinh doanh vào chương trình giảng dạy. Đặc biệt, nhóm muốn gửi lời cảm ơn sâu sắc tới giảng viên hướng dẫn là giảng viên của khóa học - thầy Nguyễn Quang Phúc, vì sự hướng dẫn tận tình và truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học vừa qua.

Trong quá trình tham gia môn học, cả nhóm đã tích lũy được nhiều kiến thức hữu ích, nâng cao khả năng phân tích và phát huy tinh thần học tập nghiêm túc, hiệu quả. Sự nhiệt tình và tận tâm của thầy đã giúp nhóm hoàn thành bài tiểu luận này. Chúng em rất mong nhận được những góp ý từ các thầy cô để bài tiểu luận được hoàn thiện hơn.

Chúc mọi người sức khỏe và nhiều thành công trong sự nghiệp giảng dạy.
Xin cảm ơn!
Nhóm Bamos

Lời cam kết

Chúng em xin đảm bảo rằng các kết quả trình bày dưới đây hoàn toàn phản ánh sự ứng dụng kiến thức dựa trên các bài giảng từ môn Học sâu trong phân tích kinh doanh, kết hợp với tài liệu tham khảo từ sách, báo và các phương tiện truyền thông khác. Chúng em cam kết rằng dự án này không sao chép hay đạo văn bất kỳ thông tin nào từ các nguồn bên ngoài. Tác giả cam kết rằng dự án sẽ được hoàn thành vào ngày 20 tháng 7 năm 2024 và sẽ được giám sát bởi giảng viên, thầy Nguyễn Quang Phúc.

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

CHAPTER 1. GIỚI THIỆU

1.1 Lý do chọn đề tài

Phân tích dữ liệu chuỗi thời gian là một lĩnh vực quan trọng trong nhiều ngành, đặc biệt là tài chính và kinh doanh, nơi dự báo chính xác có thể ảnh hưởng lớn đến quyết định đầu tư và chiến lược kinh doanh. Để hiểu rõ hơn, chúng ta cần làm quen với khái niệm cơ bản về dữ liệu chuỗi thời gian - tập hợp các quan sát được ghi lại theo trình tự thời gian, trong đó mỗi quan sát phụ thuộc vào các giá trị trước đó. Ví dụ phổ biến bao gồm giá cổ phiếu, tỷ giá hối đoái, nhiệt độ hàng ngày và doanh thu hàng tháng.

Dữ liệu chuỗi thời gian có nhiều đặc điểm nổi bật như tính tuần tự, xu hướng, mùa vụ, chu kỳ và nhiễu, giúp nó trở thành công cụ quan trọng trong phân tích và dự báo. Trong lĩnh vực tài chính, phân tích dữ liệu chuỗi thời gian giúp dự báo giá cổ phiếu, lãi suất và các chỉ số tài chính khác, từ đó hỗ trợ các nhà đầu tư và quản lý tài chính đưa ra quyết định chính xác và hiệu quả. Cụ thể, phân tích chuỗi thời gian giúp dự báo giá cổ phiếu, quản lý rủi ro, lập kế hoạch tài chính và tối ưu hóa danh mục đầu tư.

Tuy nhiên, phân tích dữ liệu chuỗi thời gian cũng đối mặt với nhiều thách thức phức tạp như tính phi tuyến, không dừng, hiệu ứng mùa vụ, chu kỳ, và sự biến động cao trong dữ liệu. Những yếu tố này tạo ra trở ngại lớn cho việc dự báo chính xác. Phân tích dữ liệu chuỗi thời gian gặp nhiều thách thức phức tạp như tính phi tuyến, không dừng, hiệu ứng mùa vụ, chu kỳ, và sự biến động cao. Ví dụ, biến động giá cổ phiếu trong năm 2020 do đại dịch COVID-19 đã làm cho việc dự báo trở nên khó khăn khi các mô hình tuyến tính như ARIMA không thể dự đoán chính xác. Thêm vào đó, các chu kỳ kinh tế và sự biến động mạnh trong các thị trường tài chính, như giá dầu và tỷ giá hối đoái, đã tạo ra những trở ngại lớn cho dự báo. Những yếu tố này đòi hỏi các phương pháp phân tích tiên tiến hơn như Deep Learning để nâng cao độ chính xác và hiệu quả trong dự báo.

Deep Learning vượt trội trong phân tích chuỗi thời gian nhờ khả năng xử lý các mối quan hệ phi tuyến và dài hạn trong dữ liệu, điều mà các mô hình truyền thống khó thực hiện. Các mô hình như RNN, LSTM, và GRU có khả năng ghi nhớ thông tin qua nhiều bước thời gian, giúp nắm bắt các mẫu quan trọng trong chuỗi dữ liệu, đặc biệt là trong các tình huống phức tạp như biến động giá cổ phiếu. LSTM và GRU được thiết kế để giải quyết vấn đề mất mát gradient, cho phép chúng học và dự báo chính xác ngay cả khi có các yếu tố ảnh hưởng từ quá khứ xa. Khả năng tự động học hỏi từ dữ liệu mà không cần các giả định cụ thể giúp Deep Learning trở nên linh hoạt và mạnh mẽ, cải thiện đáng kể độ chính xác của dự báo và tối ưu hóa chiến lược đầu tư và quản lý rủi ro trong các lĩnh vực như tài chính và kinh doanh.

Do đó, nhóm quyết định chọn đề tài "Ứng Dụng Deep Learning Trong Phân Tích Dữ Liệu Chuỗi Thời Gian" để nghiên cứu và khám phá tiềm năng của công nghệ này trong việc giải quyết các vấn đề thực tiễn, nâng cao hiệu quả kinh doanh và ứng dụng trong nhiều ngành công nghiệp khác.

1.2 Mục tiêu nghiên cứu

Mục tiêu của nghiên cứu này là xây dựng và ứng dụng các phương pháp Deep Learning nhằm cải thiện việc phân tích và dự báo dữ liệu chuỗi thời gian, với trọng tâm là dự báo giá chứng khoán của ba công ty thuộc ba lĩnh vực khác nhau: ngân hàng (ACB), bán lẻ (MWG), và công nghệ (CMC - CMG). Nghiên cứu cũng sẽ so sánh hiệu quả của các mô hình Deep Learning với các mô hình học máy truyền thống như ARIMA và SARIMA, nhằm xác định phương pháp dự báo tốt nhất cho từng lĩnh vực cụ thể.

Trong nghiên cứu này, chúng tôi sẽ sử dụng các mô hình Deep Learning tiên tiến như RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory), và GRU (Gated Recurrent Unit) để xử lý và dự báo các chuỗi thời gian phức tạp và phi tuyến tính. Các mô hình học máy truyền thống như ARIMA và SARIMA cũng sẽ được triển khai để so sánh và đánh giá hiệu quả. Việc này sẽ giúp khám phá tiềm năng của Deep Learning trong việc giải quyết các thách thức liên quan đến phân tích dữ liệu chuỗi thời gian.

Nghiên cứu sẽ đánh giá hiệu suất của các mô hình dựa trên các chỉ số quan trọng như MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), và MAPE (Mean Absolute Percentage Error). Mục tiêu cuối cùng là tìm ra mô hình dự báo tối ưu nhất cho từng mã chứng khoán, từ đó hỗ trợ các nhà đầu tư và quản lý tài chính đưa ra các quyết định đầu tư chính xác và tối ưu hóa chiến lược kinh doanh trong từng lĩnh vực cụ thể.

1.3 Phạm vi nghiên cứu

- **Không gian:** Trường Đại học Kinh tế - Luật
- **Thời gian:** 03/07/2024 - 10/08/2024

1.4 Cấu trúc dự án

Dự án này được cấu trúc thành sáu chương, mỗi chương bao gồm các nội dung chính như sau:

Chương 1: Giới thiệu

Trình bày lý do chọn đề tài, mục tiêu nghiên cứu, phạm vi nghiên cứu và cấu trúc của dự án. Chương này đặt nền móng cho toàn bộ nghiên cứu, giúp người đọc hiểu rõ bối cảnh và mục tiêu của nghiên cứu.

Chương 2: Tổng quan lý thuyết

Giới thiệu về dữ liệu chuỗi thời gian, các đặc điểm và tầm quan trọng của nó trong phân tích tài chính.

Trình bày các mô hình học máy truyền thống trong phân tích dữ liệu chuỗi thời gian như ARIMA và SARIMA, cùng với các ưu nhược điểm của chúng.

Giới thiệu về các mô hình Deep Learning trong phân tích dữ liệu chuỗi thời gian, bao gồm RNN, LSTM và GRU.

Chương 3: Tiền xử lý dữ liệu

Mô tả dữ liệu giá chứng khoán được sử dụng trong nghiên cứu, bao gồm nguồn gốc, thời gian thu thập và các đặc điểm chính của dữ liệu.

Quy trình tiền xử lý dữ liệu và phân tích khám phá dữ liệu (EDA) để làm sạch và chuẩn bị dữ liệu cho các mô hình dự báo.

Chương 4: Xây dựng và huấn luyện mô hình

Xây dựng các mô hình ARIMA và SARIMA, bao gồm các bước tiền xử lý để đưa dữ liệu về chuỗi dừng

Xây dựng các mô hình Deep Learning như RNN, LSTM và GRU và áp dụng kỹ thuật sliding windows.

Tổng hợp kết quả thử nghiệm và chuẩn bị cho việc đánh giá hiệu suất trong chương tiếp theo.

Chương 5: Đánh giá và so sánh hiệu năng các mô hình

Giới thiệu các chỉ số đánh giá hiệu suất của các mô hình khi phân tích dữ liệu chuỗi thời gian, như RMSE, MAE, MAPE và R2.

Tổng hợp kết quả từ chương 4 và phân tích so sánh hiệu năng của các mô hình.

Đánh giá ưu nhược điểm của từng mô hình và đưa ra nhận xét tổng quan về hiệu quả của các phương pháp dự báo.

CHAPTER 2. TỔNG QUAN LÝ THUYẾT

2.1 Giới thiệu về dữ liệu chuỗi thời gian

Khái niệm Dữ liệu chuỗi thời gian

Trước hết, chúng ta cần tìm hiểu khái niệm cơ bản của dữ liệu chuỗi thời gian để hiểu rõ hơn về tầm quan trọng và ứng dụng của phân tích loại dữ liệu này. Dữ liệu chuỗi thời gian (Time Series Data) là một loại dữ liệu được thu thập và ghi nhận theo thứ tự thời gian. Điều này có nghĩa là mỗi điểm dữ liệu trong chuỗi không chỉ đại diện cho giá trị của một biến số tại một thời điểm cụ thể mà còn phụ thuộc vào các giá trị trước đó trong chuỗi. Dữ liệu chuỗi thời gian được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ kinh tế, tài chính, đến khí tượng học và y tế.

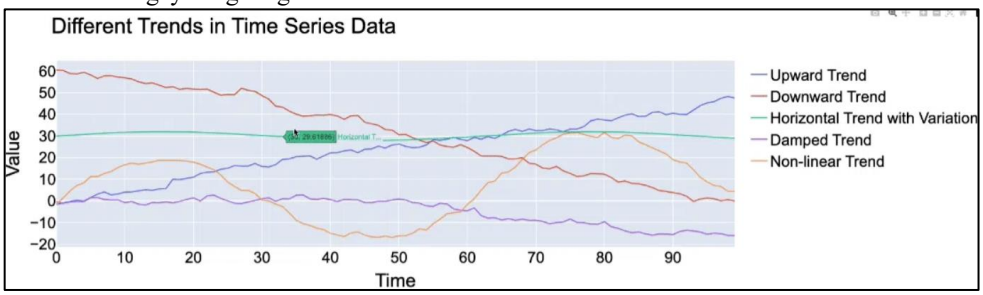
Một chuỗi thời gian có thể được biểu diễn dưới dạng một dãy các điểm dữ liệu, thường được thu thập ở các khoảng thời gian đều đặn như hàng giây, hàng phút, hàng giờ, hàng ngày, hàng tuần, hoặc hàng năm. Đơn giản hơn, ta có thể hiểu chuỗi thời gian như một tập hợp các quan sát mà thứ tự của các quan sát đó là quan trọng và không thể thay đổi. Ví dụ, giá cổ phiếu của một công ty được ghi nhận hàng ngày là một chuỗi thời gian, hoặc nhiệt độ trung bình hàng ngày của một thành phố trong suốt một năm cũng là một chuỗi thời gian. Trong cả hai trường hợp, các giá trị không thể được sắp xếp lại mà không làm mất đi ý nghĩa của dữ liệu.

Đặc điểm của Dữ liệu chuỗi thời gian

Dữ liệu chuỗi thời gian có nhiều đặc điểm nổi bật như tính tuần tự, xu hướng, mùa vụ, chu kỳ và nhiễu, giúp nó trở thành công cụ quan trọng trong phân tích và dự báo. Dữ liệu chuỗi thời gian có một số đặc điểm quan trọng và phức tạp, làm cho việc phân tích và dự báo trở nên khác biệt so với các dạng dữ liệu khác.

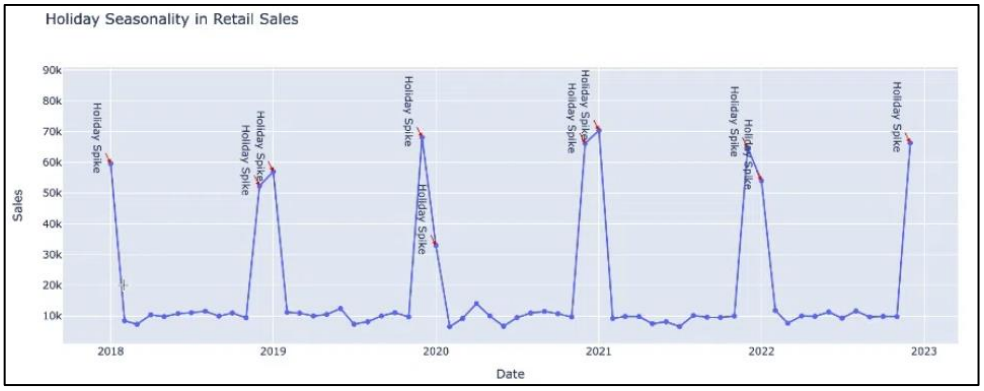
Tính tuần tự (Sequential Nature): Điểm nổi bật nhất của dữ liệu chuỗi thời gian là tính tuần tự của nó. Mỗi quan sát trong chuỗi thời gian không tồn tại độc lập mà có mối liên hệ với các quan sát trước đó. Điều này có nghĩa là các mô hình phân tích dữ liệu chuỗi thời gian phải tính đến mối quan hệ giữa các thời điểm khác nhau.

Xu hướng (Trend): Xu hướng là sự thay đổi dài hạn của dữ liệu theo thời gian. Một chuỗi thời gian có thể có xu hướng tăng lên, giảm xuống, hoặc không có xu hướng rõ ràng. Việc nhận diện và mô hình hóa xu hướng là quan trọng trong việc dự báo dữ liệu chuỗi thời gian. Ví dụ, xu hướng tăng dài hạn của giá nhà đất trong một thành phố có thể phản ánh sự phát triển kinh tế và dân số ngày càng tăng.



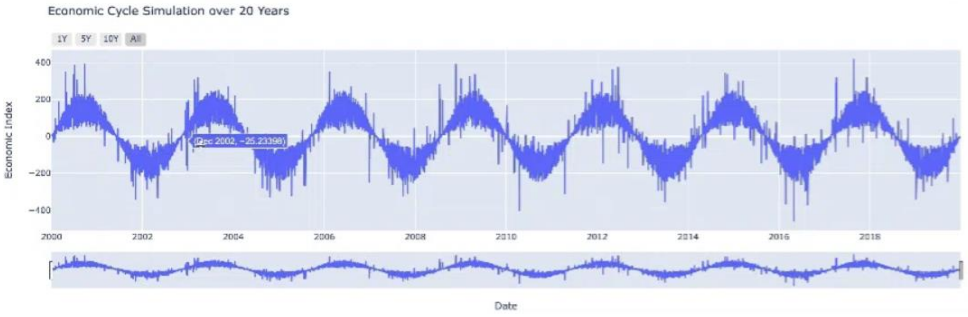
Hình 2.1 Xu hướng của time series data

Mùa vụ (Seasonality): Mùa vụ là các biến động lặp lại theo chu kỳ trong dữ liệu, thường xảy ra theo các khoảng thời gian cố định như hàng năm, hàng quý, hoặc hàng tháng. Mùa vụ có thể là kết quả của các yếu tố tự nhiên, kinh tế, hoặc văn hóa. Ví dụ, doanh thu bán lẻ thường tăng mạnh vào cuối năm do mùa mua sắm lễ hội.



Hình 2.2 Holiday Seasonality

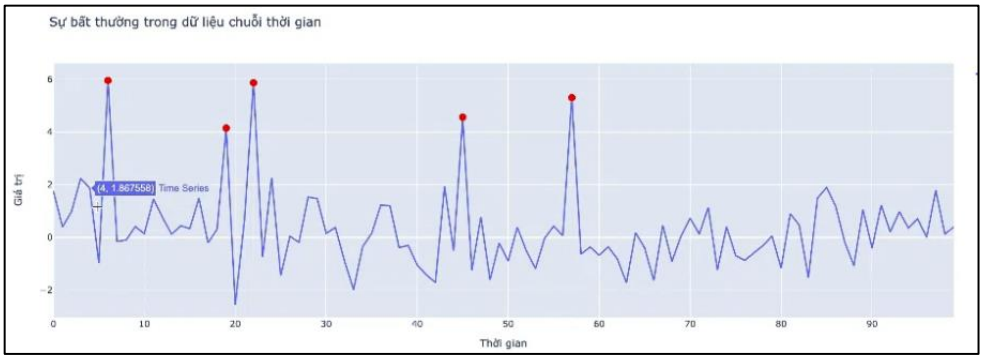
Chu kỳ (Cycle): Chu kỳ là các biến động dài hạn không đều đặn, thường kéo dài qua nhiều năm. Chu kỳ kinh tế là một ví dụ điển hình, với các giai đoạn bùng nổ (expansion) và suy thoái (recession) luân phiên nhau. Chu kỳ khác với mùa vụ ở chỗ nó không xảy ra theo một khoảng thời gian cố định và có thể bị ảnh hưởng bởi nhiều yếu tố phức tạp.



HÌnh 2.3 Đồ thị mô tả chu kỳ kinh tế từ năm 2000 đến năm 2020

Nhiễu (Noise): Nhiễu là các biến động ngẫu nhiên không thể dự đoán trước trong dữ liệu. Nhiễu có thể do nhiều nguyên nhân khác nhau, như lỗi ghi nhận dữ liệu, các sự kiện bất thường hoặc các yếu tố ngẫu nhiên khác. Việc lọc nhiễu là một bước quan trọng trong phân tích chuỗi thời gian để đảm bảo rằng các dự báo không bị ảnh hưởng bởi các biến động không có ý nghĩa.

Tính không dừng (Non-stationarity): Một chuỗi thời gian được gọi là không dừng khi các đặc điểm thống kê của nó, như trung bình, phương sai và tự tương quan, thay đổi theo thời gian. Điều này làm cho việc phân tích và dự báo trở nên khó khăn hơn, vì nhiều mô hình thống kê giả định rằng dữ liệu là dừng (stationary). Chuỗi không dừng thường cần được biến đổi (ví dụ bằng cách lấy sai phân) để trở thành dừng trước khi có thể áp dụng các mô hình phân tích.
Tính tự tương quan (Autocorrelation): Tự tương quan là mối quan hệ giữa các giá trị trong chuỗi thời gian với các giá trị trước đó của chính nó. Tự tương quan là một đặc điểm quan trọng của chuỗi thời gian, vì nó cho thấy mức độ mà giá trị của chuỗi tại một thời điểm nhất định có thể được dự đoán dựa trên các giá trị trước đó. Tự tương quan có thể giúp phát hiện các mô hình hoặc xu hướng trong dữ liệu.
Tính dị biệt (Outliers): Tính dị biệt xuất hiện khi có các giá trị bất thường hoặc cực đoan trong chuỗi thời gian. Những giá trị này có thể là kết quả của các sự kiện bất thường hoặc lỗi ghi nhận dữ liệu. Nếu không được xử lý đúng cách, các giá trị dị biệt có thể làm sai lệch kết quả phân tích và dự báo.



Hình 2.4 Iregular trong time series data

Tầm quan trọng của phân tích Dữ liệu chuỗi thời gian trong lĩnh vực Tài chính
Phân tích dữ liệu chuỗi thời gian là một công cụ không thể thiếu trong lĩnh vực tài chính, nơi mà việc dự báo và ra quyết định đầu tư dựa trên sự biến động của các chỉ số tài chính là vô cùng quan trọng. Dưới đây là một số ứng dụng chính của phân tích chuỗi thời gian trong tài chính:
- **Dự báo giá cổ phiếu:** Giá cổ phiếu là một trong những ví dụ phổ biến nhất của dữ liệu chuỗi thời gian trong tài chính. Việc dự báo giá cổ phiếu dựa trên dữ liệu lịch sử là một nhiệm vụ phức tạp nhưng rất quan trọng đối với các nhà đầu tư và quản lý quỹ. Phân tích chuỗi thời gian giúp phát hiện các xu hướng, chu kỳ và biến động ngắn hạn, từ đó đưa ra các dự đoán về giá cổ phiếu trong tương lai.
- **Quản lý rủi ro:** Trong tài chính, quản lý rủi ro là việc nhận diện, đánh giá và đưa ra các biện pháp để giảm thiểu rủi ro tài chính. Phân tích chuỗi thời gian giúp các nhà quản lý rủi ro đánh giá mức độ biến động của các tài sản tài chính, phát hiện các giai đoạn rủi ro cao và đưa ra các chiến lược phòng ngừa hợp lý.

- **Phân tích lãi suất:** Lãi suất là một yếu tố quan trọng ảnh hưởng đến nhiều lĩnh vực trong tài chính, từ đầu tư đến vay vốn. Việc phân tích và dự báo lãi suất dựa trên dữ liệu chuỗi thời gian giúp các nhà quản lý tài chính và ngân hàng đưa ra các quyết định chiến lược về đầu tư, huy động vốn và quản lý nợ.
- **Dự báo tỷ giá hối đoái:** Tỷ giá hối đoái là một yếu tố quan trọng trong giao dịch quốc tế và đầu tư xuyên biên giới. Phân tích chuỗi thời gian của tỷ giá hối đoái giúp các nhà đầu tư và doanh nghiệp dự đoán sự biến động của tỷ giá, từ đó đưa ra các quyết định về đầu tư và quản lý rủi ro ngoại hối.
- **Tối ưu hóa danh mục đầu tư:** Dữ liệu chuỗi thời gian được sử dụng để phân tích và tối ưu hóa danh mục đầu tư. Bằng cách dự đoán biến động của các tài sản khác nhau trong danh mục, các nhà đầu tư có thể điều chỉnh tỷ trọng của các tài sản để đạt được lợi nhuận cao nhất và giảm thiểu rủi ro.

- **Phân tích chu kỳ kinh tế:** Phân tích chu kỳ kinh tế dựa trên dữ liệu chuỗi thời gian giúp các nhà kinh tế và nhà hoạch định chính sách nhận diện các giai đoạn của chu kỳ kinh tế, từ đó đưa ra các quyết định về chính sách tiền tệ và tài khóa.

Những thách thức khi phân tích Dữ liệu chuỗi thời gian

Tuy nhiên, phân tích dữ liệu chuỗi thời gian cũng đối mặt với nhiều thách thức như tính phi tuyến, không dừng, hiệu ứng mùa vụ, chu kỳ và sự biến động cao trong dữ liệu. Những yếu tố này tạo ra trở ngại lớn cho việc dự báo chính xác.
- **Tính phi tuyến (Nonlinearity):** Các mối quan hệ giữa các điểm dữ liệu trong chuỗi thời gian thường không tuyến tính, gây khó khăn cho việc dự báo chính xác. Điều này đặc biệt rõ ràng trong dữ liệu tài chính, nơi phản ứng của thị trường có thể thay đổi một cách khó lường. Ví dụ: Giá cổ phiếu có thể tăng vọt sau một báo cáo thu nhập quý tích cực hoặc giảm mạnh do một vụ bê bối tài chính. Mối quan hệ giữa các sự kiện và phản ứng của thị trường không phải lúc nào cũng theo một đường thẳng dự đoán được.
- **Tính không dừng (Non-stationarity):** Đa số dữ liệu chuỗi thời gian trong thực tế có các thuộc tính thống kê không ổn định theo thời gian, như trung bình và phương sai thay đổi. Điều này gây khó khăn cho các mô hình truyền thống yêu cầu dữ liệu có tính chất dừng. Ví dụ: Dữ liệu về GDP của một quốc gia có thể thay đổi qua các giai đoạn khác nhau của chu kỳ kinh tế, từ suy thoái đến phục hồi và tăng trưởng. Trung bình và phương sai của GDP không ổn định qua thời gian.
- **Sự biến động cao và nhiễu (High Volatility and Noise):** Dữ liệu tài chính thường có biến động cao và nhiễu, làm cho việc dự báo trở nên khó khăn. Điều này đòi hỏi mô hình phải phân biệt được giữa nhiễu và thông tin hữu ích. Ví dụ: Tỷ giá hối đoái có thể bị ảnh hưởng bởi các sự kiện như tin tức về chính trị hoặc kinh tế, gây ra những biến động không dự đoán trước và khó phân tích.
- **Tính tự tương quan (Autocorrelation):** Dữ liệu chuỗi thời gian thường chứa các mối quan hệ tự tương quan, nơi giá trị trước đó ảnh hưởng đến giá trị sau. Mô hình phân tích cần xác định và tích hợp thông tin này để dự báo chính xác. Ví dụ: Trong dữ liệu về lượng mưa, lượng mưa một ngày có thể có quan hệ mật thiết với lượng mưa của những ngày gần đó do các điều kiện thời tiết kéo dài.

- **Tính dị biệt (Outliers):** Các giá trị dị biệt có thể làm méo lệch mô hình và kết quả dự báo. Việc phát hiện và xử lý các giá trị này là một phần quan trọng của quá trình phân tích. Ví dụ: Một bản báo cáo tài chính phát hiện ra sai sót kế toán có thể dẫn đến một sự biến động lớn không theo quy luật trong giá cổ phiếu mà không phản ánh xu hướng thực sự của thị trường.

Để giải quyết những thách thức này, nhóm nghiên cứu quyết định chọn đề tài "Ứng Dụng Deep Learning Trong Phân Tích Dữ Liệu Chuỗi Thời Gian." Bên cạnh đó,nhóm cũng sử dụng các mô hình truyền thống như ARIMA để so sánh và đánh giá hiệu quả. Deep Learning đã chứng tỏ khả năng vượt trội trong việc xử lý và dự báo các chuỗi thời gian phức tạp và phi tuyến tính, với các mô hình như RNN, LSTM và GRU có khả năng học và tự động hóa, giúp cải thiện độ chính xác của dự báo và tối ưu hóa chiến lược đầu tư và quản lý rủi ro. Mục tiêu của đề tài là nâng cao hiệu quả kinh doanh và ứng dụng trong nhiều ngành công nghiệp khác nhau, từ đó đề xuất những giải pháp dự báo và quản lý dữ liệu chuỗi thời gian một cách hiệu quả.

2.2 Các mô hình máy học truyền thống trong phân tích dữ liệu chuỗi thời gian

Commented [1]: kiểm tra lại hết, chatgpt nó xưng bạn kìa đcmm @luannt21416c@st.uel.edu.vn _Được giao cho luannt21416c@st.uel.edu.vn_

2.2.1 ARIMA: Mô Hình Dự Báo Chuỗi Thời Gian

ARIMA là viết tắt của AutoRegressive Integrated Moving Average, một mô hình dự báo chuỗi thời gian rất phổ biến trong thống kê và học máy. Mô hình này kết hợp ba thành phần chính: AR (AutoRegressive), I (Integrated), và MA (Moving Average). Để hiểu rõ hơn về ARIMA, chúng ta sẽ đi sâu vào từng thành phần.

- **Thành phần AR (AutoRegressive) - Hồi quy tự hồi quy**

Thành phần AR của mô hình ARIMA dựa trên ý tưởng rằng giá trị hiện tại của chuỗi thời gian có thể được dự đoán từ các giá trị trước đó của chính nó.

Công thức AR(p): $y_t = \phi_1\ y_{t-1} + \phi_2\ y_{t-2} + \ldots + \phi_p\ y_{t-p} + \epsilon_t$

Trong đó :

- y_t là giá trị hiện tại của chuỗi thời gian.
- y_{t-1} , y_{t-2} ,.., y_{t-p} là các giá trị trễ (trong quá khứ) của chuỗi thời gian.
- ϕ_1, ϕ_2, ϕ_3 là các hệ số hồi quy.
- ϵ_t là nhiễu trắng (white noise), đại diện cho sai số không thể dự đoán được.

Ví dụ: Giả sử tôi muốn dự đoán doanh số bán hàng hàng tháng của một công ty. Nếu doanh số bán hàng tháng này phụ thuộc nhiều vào doanh số của tháng trước đó, tôi có thể sử dụng mô hình AR(1): $Doanh\ số_t = 0.8\ Doanh\ số_{t-1} + \epsilon_t$

Trong mô hình này, doanh số tháng này $Doanh\ số_t$ sẽ được dự đoán bằng cách lấy 80% của doanh số tháng trước đó $Doanh\ số_{t-1}$

- **Thành phần I (Integrated) - Tích hợp**

Thành phần I của mô hình ARIMA được sử dụng để làm cho chuỗi thời gian trở nên "stationary" (ổn định), tức là không có xu hướng (trend) hoặc sự khác biệt lớn theo thời gian.

Differencing là quá trình lấy hiệu của các giá trị liên tiếp trong chuỗi thời gian. Điều này giúp loại bỏ xu hướng và làm cho chuỗi thời gian ổn định hơn.

Công thức : $y_t' = y_t - y_{t-1}$

Trong đó y_t là chuỗi thời gian sau khi đã lấy khác biệt (difference).

Ví dụ: Giả sử tôi có một chuỗi thời gian với doanh số bán hàng tăng dần mỗi tháng. Để loại bỏ xu hướng này và tập trung vào những biến động thực sự, tôi có thể lấy hiệu giữa doanh số tháng này và tháng trước đó: $y_t' = Doanh\ số_t - Doanh\ số_{t-1}$

Sau khi lấy khác biệt, chuỗi thời gian sẽ phản ánh sự thay đổi của doanh số từ tháng này sang tháng khác, thay vì giá trị tuyệt đối.

- **Thành phần MA (Moving Average) - Trung bình động**

Thành phần MA của mô hình ARIMA dựa trên ý tưởng rằng giá trị hiện tại có thể được dự đoán từ các nhiễu (sai số) trong quá khứ, thay vì từ giá trị của chính chuỗi thời gian.

Công thức MA(q): $y_t = \mu + \epsilon_t + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + \ldots + \phi_q\epsilon_{t-q}$

Trong đó:

- $y_t$là giá trị hiện tại của chuỗi thời gian.
- ϵ_t, ϵ_{t-1}là các nhiễu (sai số) trong quá khứ.
- ϕ_1,ϕ_2,ϕ_3 là các hệ số của mô hình MA.
- μ là giá trị trung bình của chuỗi thời gian.

Ví dụ: Giả sử tôi muốn dự đoán doanh số bán hàng và nhận thấy rằng những biến động không dự đoán trước (nhiễu) trong quá khứ có ảnh hưởng lớn đến doanh số hiện tại. Tôi có thể sử dụng mô hình MA(1): $Doanh\ số_t = \mu + \epsilon_t + 0.5.\ \epsilon_{t-1}$.

Trong mô hình này, doanh số hiện tại được dự đoán bằng cách cộng nhiễu hiện tại và 50% của nhiễu từ tháng trước

4. ARIMA - Kết hợp các thành phần

Mô hình ARIMA kết hợp ba thành phần AR, I, và MA lại với nhau để dự đoán chuỗi thời gian. Một mô hình ARIMA tổng quát sẽ được viết dưới dạng ARIMA(p, d, q), trong đó:

- p là số lượng bậc của thành phần AR.
- d là số lượng khác biệt cần thiết để làm cho chuỗi thời gian ổn định.
- q là số lượng bậc của thành phần MA.

Ví dụ thực tế về ARIMA(1,1,1):

Giả sử tôi có một chuỗi thời gian với doanh số bán hàng hàng tháng và muốn sử dụng mô hình ARIMA để dự đoán doanh số của tháng tiếp theo. **AR(1):** Doanh số tháng này phụ thuộc vào doanh số của tháng trước đó .**I(1):** Tôi cần lấy khác biệt để loại bỏ xu hướng trong doanh số

bán hàng. **MA(1):** Tôi cũng thấy rằng nhiễu từ tháng trước đó có ảnh hưởng đến doanh số hiện tại.Mô hình **ARIMA(1,1,1**) sẽ có dạng: $Doanh\ số_t = \phi_1\ (\ Doanh\ số_{t-1} - \ Doanh\ số_{t-2}) + \epsilon_t + \ \phi_1\ \epsilon_{t-1}$

Trong mô hình này tôi lấy khác biệt của doanh số để loại bỏ xu hướng. và dự đoán doanh số hiện tại dựa trên giá trị doanh số đã lấy khác biệt và nhiễu từ tháng trước đó.

Mô hình ARIMA là một công cụ mạnh mẽ để dự đoán chuỗi thời gian bằng cách kết hợp ba thành phần: AR (AutoRegressive) - Dựa trên giá trị trong quá khứ của chuỗi thời gian, I (Integrated) - Giúp làm cho chuỗi thời gian ổn định thông qua quá trình khác biệt, MA (Moving Average) - Dựa trên nhiễu (sai số) trong quá khứ. Thông qua việc lựa chọn các thông số p, d, q phù hợp, tôi có thể xây dựng một mô hình ARIMA để dự đoán hiệu quả các giá trị tương lai của chuỗi thời gian.

2.2.2 Các biến thể Sarima và Sarimax

SARIMA

SARIMA là viết tắt của Seasonal AutoRegressive Integrated Moving Average, một biến thể mở rộng của mô hình ARIMA, được thiết kế để xử lý các chuỗi thời gian có yếu tố mùa vụ rõ rệt. Trong các chuỗi thời gian này, dữ liệu không chỉ phụ thuộc vào các giá trị gần đây mà còn phụ thuộc vào các giá trị ở cùng một thời điểm trong các chu kỳ trước.

SARIMA mở rộng ARIMA bằng cách thêm các thành phần mùa vụ. Mô hình SARIMA thường được viết dưới dạng SARIMA(p, d, q)(P, D, Q, m) với p,d,q thì tượng tự như mô hình ARIMA truyền thống, còn bộ tham số mới sẽ là **P** (Seasonal AR order): Bậc của phần tự hồi quy theo mùa vụ, **D** (Seasonal differencing order): Bậc của phần khác biệt theo mùa vụ, **Q** (Seasonal MA order): Bậc của phần trung bình động theo mùa vụ, **m** (Period): Chu kỳ mùa vụ (số lượng các bước thời gian trong một chu kỳ).

Ví Dụ Mùa Vụ trong Bán Lẻ:

Giả sử tôi là quản lý của một chuỗi cửa hàng bán lẻ và muốn dự đoán doanh số bán hàng hàng tháng. Doanh số bán hàng của tôi có yếu tố mùa vụ rõ rệt, với sự gia tăng mạnh vào các tháng cuối năm do các kỳ nghỉ lễ. Tôi có thể sử dụng **ARIMA** để dự đoán dựa trên các tháng gần đây, nhưng ARIMA sẽ bỏ qua yếu tố mùa vụ, tức là sẽ không nhận ra rằng doanh số tháng 12 luôn cao hơn các tháng khác trong năm.

Nhưng với Mô hình SARIMA sẽ nhận ra yếu tố mùa vụ này và tính toán một cách riêng biệt: (p, d, q): Phần không mùa vụ có thể là ARIMA(1,1,1), nghĩa là dựa vào doanh số tháng trước đó, loại bỏ xu hướng chung, và tính đến sai số. (P, D, Q, m): Phần mùa vụ có thể là SARIMA(0,1,1,12), nghĩa là lấy khác biệt theo mùa vụ (D=1) và tính đến nhiễu của mùa vụ trước đó với chu kỳ 12 tháng.

SARIMA sẽ giúp dự đoán rằng vào tháng 12, doanh số sẽ tăng đột biến so với các tháng khác, dựa trên các dữ liệu lịch sử từ các kỳ nghỉ lễ của những năm trước.

Ví Dụ Mùa Vụ trong Nông Nghiệp:

Hãy tưởng tượng tôi là một nhà nghiên cứu nông nghiệp muốn dự đoán sản lượng lúa của một vùng qua các mùa vụ khác nhau. Sản lượng lúa có thể phụ thuộc vào các yếu tố như lượng mưa, nhiệt độ, và chu kỳ trồng trọt, vốn là những yếu tố có tính chất mùa vụ. Nếu chỉ sử dụng ARIMA, tôi có thể dự đoán sản lượng lúa dựa trên các yếu tố phi mùa vụ, như sản lượng của vụ trước và các biến đổi ngẫu nhiên.
Và với mô hình SARIMA sẽ cho phép tôi tính đến các yếu tố mùa vụ quan trọng. Ta có thể sử dụng ARIMA(2,1,1) để dự đoán sản lượng dựa trên các vụ gần đây và loại bỏ xu hướng chung và với phần mùa ta sẽ xử lý bằng cách dùng SARIMA (1,1,0,4) với m = 4 đại diện cho 4 mùa vụ trong năm.
Và có thể SARIMA sẽ giúp dự đoán rằng sản lượng lúa trong vụ Đông sẽ khác biệt rõ ràng so với các vụ khác trong năm, dựa trên dữ liệu sản lượng của các vụ Đông từ các năm trước.
Thì nhìn chung SARIMA là một mô hình mạnh mẽ cho việc dự đoán các chuỗi thời gian có yếu tố mùa vụ. Bằng cách kết hợp các thành phần của ARIMA với các tham số mùa vụ, SARIMA có thể nắm bắt các xu hướng và chu kỳ trong dữ liệu, giúp cải thiện độ chính xác trong các dự báo có tính chất định kỳ.

SARIMAX: Mở Rộng của SARIMA với Các Biến Ngoại Sinh
SARIMAX là viết tắt của Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors, một biến thể mở rộng của mô hình SARIMA. SARIMAX không chỉ xử lý các yếu tố mùa vụ và các thành phần ARIMA, mà còn cho phép đưa thêm vào các biến ngoại sinh (exogenous variables) – những yếu tố bên ngoài có thể ảnh hưởng đến chuỗi thời gian tôi đang dự đoán.
Mô hình SARIMAX được viết dưới dạng SARIMAX(p, d, q)(P, D, Q, m, X), trong đó:

- (p, d, q): Các tham số của thành phần ARIMA như đã giải thích trước đó.
- (P, D, Q, m): Các tham số mùa vụ như trong mô hình SARIMA.
- X: Là tập hợp các biến ngoại sinh (exogenous variables), tức là những biến ngoài chuỗi thời gian chính nhưng có khả năng ảnh hưởng đến chuỗi thời gian đó.`

Ví Dụ 1: Dự Đoán Doanh Số Bán Hàng Kết Hợp Với Chiến Dịch Quảng Cáo
Giả sử bạn đang quản lý một chuỗi cửa hàng và muốn dự đoán doanh số bán hàng hàng tháng. Tôi nhận thấy rằng ngoài yếu tố mùa vụ (chẳng hạn, doanh số tăng cao vào dịp cuối năm), doanh số còn bị ảnh hưởng bởi các chiến dịch quảng cáo mà tôi triển khai.

Với ARIMA thông thường tôi chỉ có thể dùng để dự đoán doanh số dựa trên các dữ liệu bán hàng trong quá khứ, nhưng ARIMA sẽ bỏ qua tác động của các chiến dịch quảng cáo.

Ta có thể sử dụng SARIMA để dự đoán doanh số theo mùa vụ, nhưng SARIMA cũng sẽ không tính đến các yếu tố bên ngoài như quảng cáo.

Nhưng với SARIMAX trở nên hữu ích. Với SARIMAX, tôi có thể thêm biến ngoại sinh là ngân sách quảng cáo (hoặc số lần chạy quảng cáo) vào mô hình: với **(p, d, q)(P, D, Q, m),** bạn vẫn sử dụng cấu trúc mùa vụ để dự đoán xu hướng mùa vụ như trong SARIMA. **X:** Tôi thêm biến ngoại sinh là số tiền chi cho quảng cáo mỗi tháng.

SARIMAX sẽ giúp tôi dự đoán rằng doanh số không chỉ tăng vào tháng 12 (theo mùa vụ) mà còn có thể tăng mạnh nếu tôi đầu tư thêm vào quảng cáo trong tháng đó. Điều này cho phép tôi thấy rõ hơn mối quan hệ giữa chi tiêu quảng cáo và doanh số bán hàng.

Ví Dụ 2: Dự Đoán Lượng Điện Năng Tiêu Thụ Kết Hợp Với Nhiệt Độ
Nếu tôi đang làm việc tại một công ty cung cấp điện năng và muốn dự đoán lượng điện tiêu thụ hàng tháng. Tôi biết rằng lượng điện tiêu thụ không chỉ phụ thuộc vào các yếu tố mùa vụ (như mùa hè thì tiêu thụ điện cao hơn) mà còn bị ảnh hưởng bởi nhiệt độ hàng ngày.

- Có thể sử dụng SARIMA để dự đoán lượng điện tiêu thụ dựa trên dữ liệu tiêu thụ trong quá khứ, nhưng mô hình này sẽ không tính đến ảnh hưởng của nhiệt độ.
- Bạn có thể thêm nhiệt độ trung bình hàng ngày hoặc hàng tháng như một biến ngoại sinh trong mô hình SARIMAX (p, d, q)(P, D, Q, m). Tôi vẫn sử dụng cấu trúc mùa vụ để dự đoán xu hướng theo mùa. Ngoài ra sẽ sử dụng thêm biến X là biến ngoại sinh là nhiệt độ trung bình của từng tháng.

SARIMAX sẽ cho phép tôi dự đoán lượng điện tiêu thụ trong tháng hè sẽ tăng mạnh nếu nhiệt độ cao hơn bình thường, không chỉ do mùa hè mà còn vì mức nhiệt độ tăng cao ảnh hưởng đến nhu cầu sử dụng điện.

Nhìn chung thì SARIMAX là một mô hình mạnh mẽ, không chỉ xử lý các yếu tố mùa vụ và chuỗi thời gian như SARIMA mà còn cho phép tích hợp các yếu tố bên ngoài (biến ngoại sinh) vào mô hình dự báo. Điều này giúp cải thiện độ chính xác của dự đoán bằng cách tính đến các yếu tố có ảnh hưởng lớn đến chuỗi thời gian nhưng nằm ngoài phạm vi của dữ liệu chính.

2.2.3 Ưu nhược điểm của ARIMA/SARIMA

Mô hình ARIMA (AutoRegressive Integrated Moving Average) là một công cụ mạnh mẽ trong việc dự báo các chuỗi thời gian, đặc biệt là khi dữ liệu có xu hướng nhưng không có yếu tố mùa vụ rõ rệt. Một ưu điểm lớn của ARIMA là khả năng mô hình hóa các chuỗi thời gian bằng cách kết hợp ba thành phần chính: tự hồi quy (AR), khác biệt (I), và trung bình động (MA). Điều này cho phép ARIMA linh hoạt trong việc điều chỉnh để dự báo các chuỗi thời gian có xu hướng dài hạn, ví dụ như dự đoán giá cổ phiếu, nơi xu hướng chung là tăng hoặc giảm theo thời gian mà không có sự lặp lại rõ ràng theo mùa vụ. Tuy nhiên, một nhược điểm đáng kể của ARIMA là nó không xử lý tốt khi chuỗi thời gian có yếu tố mùa vụ. Khi áp dụng ARIMA để dự đoán doanh số bán hàng của một cửa hàng có xu hướng tăng mạnh vào dịp cuối năm, mô hình này có thể bỏ qua yếu tố mùa vụ, dẫn đến dự báo kém chính xác.

Để khắc phục hạn chế của ARIMA, mô hình SARIMA (Seasonal ARIMA) đã được phát triển, với khả năng bổ sung các tham số mùa vụ. SARIMA không chỉ mô hình hóa các thành phần của ARIMA mà còn thêm vào khả năng dự báo các chu kỳ lặp lại theo mùa. Ví dụ, khi dự đoán lượng tiêu thụ điện năng theo tháng, SARIMA có thể nhận ra các mẫu mùa vụ như mức tiêu thụ điện tăng cao vào mùa hè do nhu cầu làm mát tăng. Điều này làm cho SARIMA trở thành một lựa chọn lý tưởng cho các chuỗi thời gian có yếu tố mùa vụ rõ rệt. Tuy nhiên, SARIMA cũng có những nhược điểm. Mô hình này phức tạp hơn ARIMA, đòi hỏi phải điều chỉnh nhiều

tham số hơn và yêu cầu sự hiểu biết sâu rộng về cấu trúc dữ liệu và cách thức vận hành của mô hình. Trong thực tế, nếu dữ liệu có quá nhiều yếu tố mùa vụ phức tạp hoặc không ổn định, việc sử dụng SARIMA có thể trở nên khó khăn và tốn kém thời gian để đạt được một mô hình chính xác.

Cả ARIMA và SARIMA đều có giá trị sử dụng trong những ngữ cảnh khác nhau. ARIMA thích hợp hơn với các chuỗi thời gian không có yếu tố mùa vụ, hoặc khi dữ liệu có xu hướng dài hạn rõ ràng. Ngược lại, SARIMA lại vượt trội khi dữ liệu có yếu tố mùa vụ, nhưng đòi hỏi sự cẩn thận trong việc cấu hình và điều chỉnh tham số để đảm bảo dự báo chính xác. Quyết định sử dụng mô hình nào phụ thuộc vào bản chất của dữ liệu và mục tiêu dự báo cụ thể.

2.2.4 Một số mô hình khác như:

Linear SVM

Linear Support Vector Machine (SVM) là một kỹ thuật học máy được sử dụng phổ biến trong phân loại và hồi quy. Mặc dù SVM thường được liên kết với bài toán phân loại, nó cũng có thể được áp dụng trong phân tích chuỗi thời gian thông qua một phương pháp gọi là SVM hồi quy (Support Vector Regression - SVR). Khi áp dụng Linear SVM cho phân tích chuỗi thời gian, mục tiêu chính là dự đoán các giá trị tương lai của chuỗi thời gian dựa trên dữ liệu quá khứ.

Khi sử dụng Linear SVM cho chuỗi thời gian, mô hình sẽ cố gắng tìm ra một siêu phẳng (hyperplane) trong không gian đặc trưng để dự đoán giá trị của biến mục tiêu (thường là giá trị tương lai của chuỗi thời gian) sao cho sai số dự đoán nằm trong một ngưỡng cho trước. Cụ thể, Linear SVM trong phân tích chuỗi thời gian sẽ tạo ra một hàm tuyến tính để dự đoán các điểm dữ liệu trong tương lai dựa trên một tập hợp các quan sát quá khứ.

Linear SVM có một số ưu điểm trong phân tích chuỗi thời gian. Một trong những ưu điểm chính là khả năng tìm kiếm một giải pháp tối ưu toàn cục cho bài toán hồi quy, nhờ vào việc tối ưu hóa khoảng cách giữa các điểm dữ liệu và siêu phẳng. Điều này giúp giảm nguy cơ quá khớp (overfitting) và cải thiện khả năng tổng quát của mô hình. Linear SVM cũng hoạt động tốt với các tập dữ liệu có kích thước lớn và không quá phức tạp.

Tuy nhiên, Linear SVM có một số nhược điểm khi áp dụng cho chuỗi thời gian. Vì SVM chủ yếu dựa trên một mô hình tuyến tính, nó có thể không nắm bắt được các mối quan hệ phi tuyến giữa các điểm dữ liệu trong chuỗi thời gian, đặc biệt là khi có các mẫu phức tạp hoặc các yếu tố mùa vụ. Ngoài ra, SVM không được thiết kế đặc biệt cho dữ liệu chuỗi thời gian, vì vậy trong một số trường hợp, các phương pháp truyền thống như ARIMA hoặc SARIMA có thể hiệu quả hơn.

Decision Tree

Decision Tree là một phương pháp học máy mạnh mẽ thường được sử dụng trong các bài toán phân loại và hồi quy. Trong bối cảnh phân tích chuỗi thời gian, Decision Tree có thể được áp dụng để dự đoán các giá trị tương lai của chuỗi thông qua một quá trình gọi là Decision Tree

Regression. Cách tiếp cận này phân tách dữ liệu dựa trên các đặc trưng quan trọng và tạo ra các nhánh quyết định, giúp mô hình dự đoán các giá trị dựa trên dữ liệu quá khứ.

Decision Tree trong phân tích chuỗi thời gian hoạt động bằng cách chia dữ liệu thành các phân đoạn nhỏ hơn dựa trên các giá trị đặc trưng, từ đó tạo ra các nút và nhánh trong cây quyết định. Mỗi nút trong cây tương ứng với một điều kiện trên một đặc trưng của dữ liệu, và các nhánh con từ nút đó biểu diễn các phân đoạn của dữ liệu đáp ứng điều kiện đó. Khi đi qua từng nút, dữ liệu sẽ được phân tách cho đến khi đạt đến các nút lá, nơi dự báo giá trị tương lai được thực hiện.

Một trong những ưu điểm lớn của Decision Tree là khả năng xử lý các mối quan hệ phi tuyến giữa các đặc trưng và khả năng hoạt động tốt với dữ liệu phức tạp, bao gồm cả dữ liệu có nhiều đặc trưng hoặc yếu tố không đồng nhất. Decision Tree cũng rất dễ hiểu và trực quan, vì cấu trúc cây giúp mô hình dễ dàng giải thích các quyết định dự báo.

Tuy nhiên, Decision Tree cũng có một số nhược điểm khi áp dụng cho chuỗi thời gian. Cụ thể, mô hình cây có thể dễ bị quá khớp (overfitting), đặc biệt là khi dữ liệu chứa nhiều nhiễu hoặc khi cây quyết định trở nên quá sâu với quá nhiều nhánh. Ngoài ra, Decision Tree thường không xử lý tốt dữ liệu chuỗi thời gian liên tục, vì nó không trực tiếp tính đến thứ tự thời gian của dữ liệu, mà chỉ dựa trên các đặc trưng rời rạc. Điều này có thể làm giảm độ chính xác của dự báo trong một số trường hợp.

2.2.5 Arima là mô hình tối ưu tốt với Time Series

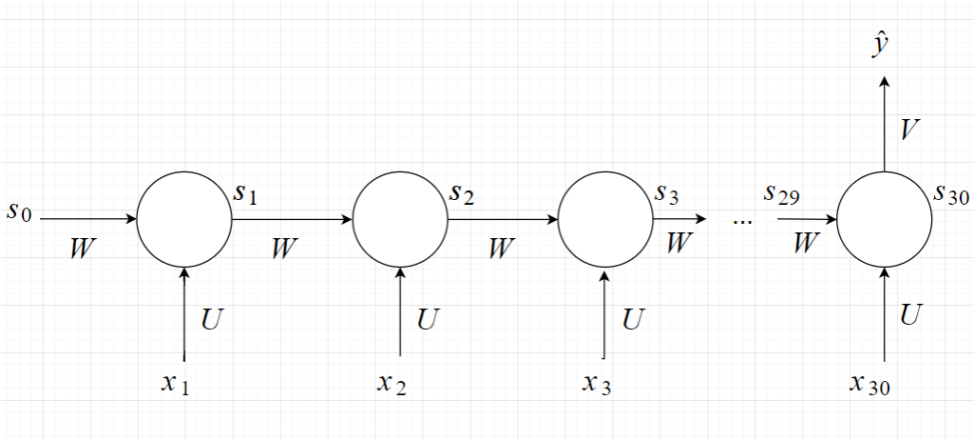
Với những gì đã phân tích ở phía trên thì có thể nói ARIMA được coi là một trong những mô hình hiệu quả nhất khi phân tích chuỗi thời gian, đặc biệt là trong các bài toán dự báo. Một trong những lợi ích lớn của ARIMA là mặc dù có tính toán phức tạp, mô hình này lại dễ dàng triển khai và sử dụng trong thực tế. Các thuật toán ước lượng tham số của ARIMA đã được phát triển hoàn thiện và tích hợp sẵn trong nhiều công cụ phân tích và phần mềm thống kê. Điều này giúp các nhà phân tích có thể dễ dàng áp dụng ARIMA để tạo ra các dự báo chính xác mà không cần phải xử lý các mô hình phức tạp hơn hoặc yêu cầu kỹ thuật tính toán cao hơn.

ARIMA được xây dựng trên một nền tảng lý thuyết vững chắc về phân tích chuỗi thời gian, dựa vào các thuộc tính của chuỗi thời gian dừng (stationary) và các công cụ thống kê như hàm tự tương quan (autocorrelation). Những khái niệm này đã được nghiên cứu và chứng minh qua nhiều thập kỷ, tạo ra một sự tin tưởng rộng rãi trong cộng đồng khoa học và phân tích dữ liệu về tính hiệu quả và chính xác của ARIMA.

ARIMA là một mô hình phân tích chuỗi thời gian hiệu quả nhất bởi vì nó mang lại sự linh hoạt, dễ triển khai, và khả năng dự báo chính xác trong nhiều bối cảnh khác nhau. Tính đơn giản và nền tảng lý thuyết vững chắc của ARIMA cũng giúp nó trở thành lựa chọn hàng đầu cho các bài toán phân tích chuỗi thời gian, đặc biệt là khi dữ liệu không có yếu tố mùa vụ hoặc khi cần một mô hình đơn giản nhưng vẫn đạt được độ chính xác cao.

2.3 Deep Learning trong phân tích dữ liệu chuỗi thời gian

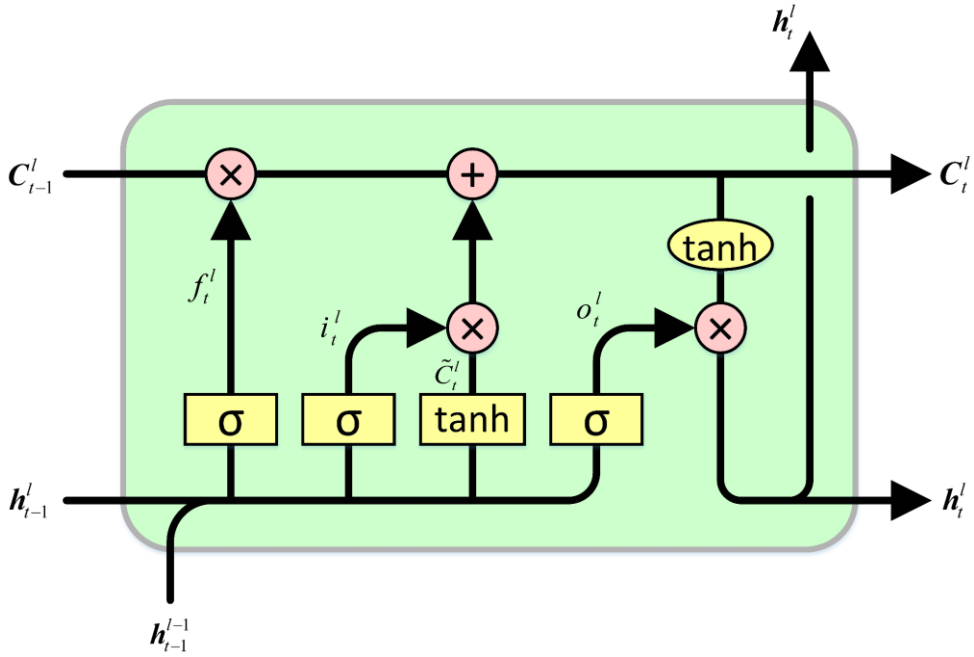
2.3.1 Recurrent Neural Networks - RNN)



Hình 2.5 Kiến trúc mạng nơ ron hồi quy

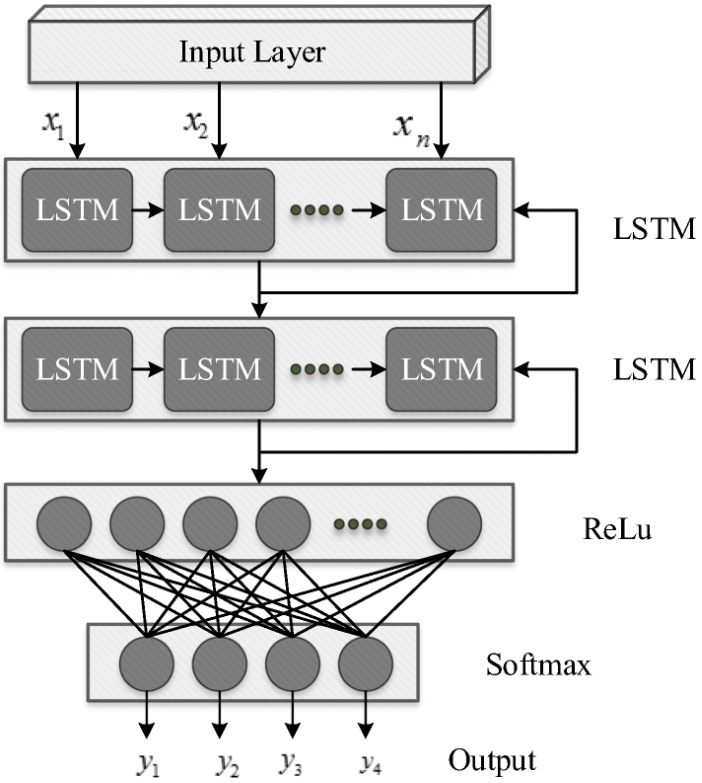
Mạng nơ-ron hồi quy (Recurrent Neural Networks - RNN) được đánh giá cao trong phân tích dữ liệu chuỗi thời gian chứng khoán nhờ khả năng tích hợp sâu sắc thông tin lịch sử, điều này cho phép RNN không chỉ dự báo chính xác giá cổ phiếu và chỉ số thị trường mà còn phân tích các biến số tài chính phức tạp khác. Với cấu trúc đặc biệt cho phép xử lý thông tin liên tục qua thời gian, RNN có thể nhận diện các mẫu hình và xu hướng từ dữ liệu quá khứ, giúp nhận định các điểm mua hoặc bán lý tưởng. Điểm nổi bật của RNN trong lĩnh vực chứng khoán là khả năng học độc lập mà không cần các giả định về tính chất phân phối của dữ liệu, làm cho nó thích hợp với bản chất thường xuyên biến động và không ổn định của thị trường tài chính. Ngoài ra, RNN có thể cập nhật liên tục với dữ liệu mới, cho phép mô hình phản ứng nhanh với các biến động thị trường mới mà không cần huấn luyện lại từ đầu. Những đặc tính này khiến RNN trở thành công cụ không thể thiếu, giúp các nhà đầu tư và chuyên gia tài chính không chỉ hiểu rõ hơn về động thái thị trường mà còn tối ưu hóa các chiến lược đầu tư để đạt hiệu quả cao nhất.

2.3.2 Long short term memory (LSTM - Stacked LSTM)



Hình 2.6 Kiến trúc LSTM

Mạng Long Short-Term Memory (LSTM) là một biến thể tiên tiến của RNN, được thiết kế để khắc phục những hạn chế của RNN như vấn đề mất mát và bùng nổ gradient, đặc biệt hữu ích trong phân tích chuỗi thời gian chứng khoán. LSTM sử dụng cấu trúc cổng đặc biệt để điều chỉnh thông tin được lưu trữ và quên đi trong quá trình học, cho phép nó lưu giữ thông tin quan trọng trong thời gian dài và loại bỏ những thông tin không cần thiết. Điều này làm cho LSTM đặc biệt phù hợp với môi trường chứng khoán, nơi mà dữ liệu có đặc tính phi tuyến tính cao và cần phản ứng nhanh với các biến động thị trường.

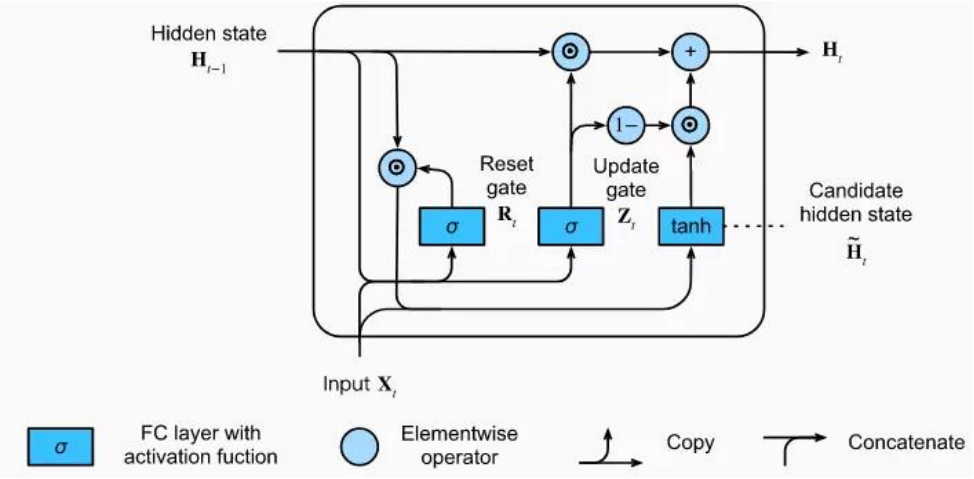


Hình 2.7 Kiến trúc Stacked LSTM (Deep LSTM)

LSTM Stacked, hay mạng LSTM xếp chồng, là một cấu trúc phức tạp hơn nơi nhiều lớp LSTM được xếp chồng lên nhau để tăng cường khả năng học sâu và phức tạp hơn. Việc này tạo ra một mô hình mạnh mẽ hơn, có khả năng nắm bắt các mối quan hệ phức tạp và mẫu hình dài hạn trong dữ liệu chứng khoán, điều không thể thực hiện với một lớp LSTM đơn. LSTM Stacked thường được sử dụng trong các tác vụ dự báo chính xác cao, giúp các nhà phân tích tài chính đưa ra dự báo dài hạn và chi tiết hơn về các chỉ số thị trường.

2.3.3 GRU

Mạng Gated Recurrent Unit (GRU) là một biến thể đơn giản hóa của mạng LSTM (Long Short-Term Memory), được thiết kế để giải quyết các vấn đề tương tự như LSTM, đặc biệt là hiện tượng mất mát và bùng nổ gradient trong mạng nơ-ron hồi quy (RNN). GRU được giới thiệu với mục đích giảm thiểu độ phức tạp tính toán và yêu cầu tài nguyên so với LSTM, trong khi vẫn duy trì hiệu quả cao trong việc xử lý dữ liệu tuần tự và dự đoán chuỗi thời gian.



Hình 2.8 - Kiến trúc GRU (Mạng Hồi Tiếp với nút có Cổng)

GRU cũng sử dụng các cơ chế cổng để kiểm soát luồng thông tin qua mạng, nhưng thay vì ba cổng như LSTM (cổng vào, cổng quên, và cổng đầu ra), GRU chỉ có hai cổng chính: cổng cập nhật và cổng xoá.

- **Cổng cập nhật (Update Gate):** Cổng này quyết định bao nhiêu phần thông tin từ trạng thái trước đó sẽ được chuyển sang trạng thái hiện tại. Nó giúp GRU giữ lại thông tin cần thiết từ quá khứ và giảm thiểu hiện tượng mất mát thông tin trong quá trình truyền qua các bước thời gian.
- **Cổng xoá (Reset Gate):** Cổng này quyết định bao nhiêu phần thông tin từ trạng thái trước đó sẽ được quên đi. Nó giúp mô hình loại bỏ những thông tin không cần thiết, giúp cho quá trình học tập trở nên hiệu quả hơn.

GRU đặc biệt hữu ích trong các bài toán dự đoán chuỗi thời gian, chẳng hạn như dự đoán giá cổ phiếu. Với cấu trúc đơn giản hơn LSTM nhưng vẫn mạnh mẽ, GRU có thể xử lý tốt các đặc tính phi tuyến tính và biến động của dữ liệu thị trường chứng khoán. GRU cũng có xu hướng đòi hỏi ít tài nguyên tính toán hơn, làm cho nó trở thành một lựa chọn hợp lý cho các hệ thống dự báo thời gian thực và các ứng dụng phân tích tài chính nơi mà hiệu suất là một yếu tố quan trọng.

Nhìn chung, mặc dù GRU đơn giản hơn so với LSTM, nhưng nó vẫn duy trì hiệu quả vượt trội trong nhiều bài toán phân tích và dự đoán chuỗi thời gian, đặc biệt trong môi trường tài chính.

2.4 Các nghiên cứu liên quan

Dự báo chuỗi thời gian đã là một chủ đề quan trọng trong nghiên cứu kinh tế và tài chính. Các phương pháp truyền thống như ARIMA đã được sử dụng rộng rãi trong lĩnh vực này. Theo Box và Jenkins (1970), ARIMA là một phương pháp chuẩn mực cho dự báo chuỗi thời gian, dựa trên giả định rằng dữ liệu có tính tuyến tính và độ lệch chuẩn không đổi. Tuy nhiên, mô hình ARIMA có những hạn chế, đặc biệt là khi phải đối mặt với các chuỗi thời gian phi tuyến tính và có biến động cao (Box & Jenkins, 1970). Vantuch và Zelinka (2014) cũng chỉ ra rằng

ARIMA yêu cầu chuỗi thời gian phải được chuyển đổi thành dạng dừng, điều này đôi khi không phù hợp với dữ liệu tài chính thực tế (Vantuch & Zelinka, 2014).

Holt-Winters Exponential Smoothing là một phương pháp khác được sử dụng trong dự báo chuỗi thời gian, đặc biệt là trong các dự báo ngắn hạn. Theo Rahman và cộng sự (2016), phương pháp này xử lý dữ liệu đơn biến tốt nhưng gặp hạn chế khi phải xử lý sự thay đổi theo mùa và chọn giá trị khởi đầu (Rahman et al., 2016).

Trong những năm gần đây, các phương pháp học máy và học sâu đã thu hút sự chú ý của các nhà nghiên cứu do khả năng xử lý dữ liệu phi tuyến tính và có độ phức tạp cao. Random Forest và MARS là hai mô hình học máy phổ biến trong lĩnh vực này. Theo Polamuri và cộng sự (2019), Random Forest có thể gặp khó khăn khi dữ liệu có tính ngẫu nhiên cao, nhưng MARS lại có ưu điểm trong việc lựa chọn tự động các đặc trưng quan trọng và xây dựng các mô hình phi tuyến hiệu quả (Polamuri et al., 2019).

LSTM, một loại mạng nơ-ron hồi quy sâu, đã được Hochreiter và Schmidhuber (1997) giới thiệu và đã chứng minh được khả năng vượt trội trong việc dự báo chuỗi thời gian phức tạp. LSTM có khả năng học các mối quan hệ dài hạn trong dữ liệu, điều mà các mô hình truyền thống như ARIMA khó có thể làm được (Hochreiter & Schmidhuber, 1997). Moghar và Hamiche (2020) cũng đã áp dụng thành công LSTM trong dự báo giá cổ phiếu, cho thấy độ chính xác cao hơn so với các mô hình truyền thống và học máy khác (Moghar & Hamiche, 2020).

Như vậy, mặc dù các phương pháp truyền thống như ARIMA và Holt-Winters vẫn còn giá trị, các mô hình học sâu như LSTM đã chứng minh sự ưu việt trong việc xử lý dữ liệu tài chính có tính phi tuyến và độ biến động cao, mở ra những hướng nghiên cứu mới đầy tiềm năng trong lĩnh vực dự báo tài chính.

CHAPTER 3. TIỀN XỬ LÝ DỮ LIỆU

Commented [2]: @luannt21416c@st.uel.edu.vn
Được giao cho luannt21416c@st.uel.edu.vn

3.1 Dữ liệu

Dữ liệu sẽ được lấy ngẫu nhiên của 3 công ty tương ứng cho 3 lĩnh vực khác nhau

- Ngành ngân hàng: ACB (Ngân hàng Á Châu)
- Ngành bán lẻ: MWG (Công ty cổ phần đầu tư Thế Giới Di Động)
- Ngành công nghệ: CMC (Tập đoàn công nghệ CMC)

Và để có được dữ liệu của các công ty trên thì nhóm chúng tôi sử dụng thư viện vnstock để có thể thu thập dữ liệu. Nói sơ về vnstock thì vnstock là một thư viện Python được thiết kế để truy xuất dữ liệu tài chính từ các sàn giao dịch chứng khoán tại Việt Nam, bao gồm cả giá cổ phiếu, thông tin doanh nghiệp và các chỉ số khác. Nó cho phép người dùng dễ dàng tải dữ liệu lịch sử và thực hiện phân tích tài chính trên các cổ phiếu niêm yết tại thị trường Việt Nam.

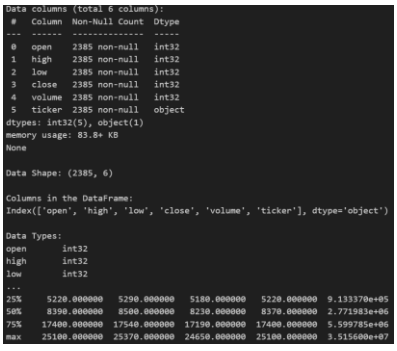
Mô tả sơ về dữ liệu thu thập được thì dữ liệu được thu thập từ ngày 1 tháng 1 năm 2015 đến ngày 28 tháng 7 năm 2024 với tần suất dữ liệu được lấy theo tần suất hàng ngày. Bộ dữ liệu này bao gồm các cột:

- open: Giá mở cửa
- high: Giá cao nhất trong ngày
- low: Giá thấp nhất trong ngày
- close: Giá đóng cửa
- volume: Khối lượng giao dịch
- ticker: Mã chứng khoán của công ty (ACB, MWG, CMG)

3.2 Tiền xử lý dữ liệu và EDA

Tiếp theo chúng ta sẽ đi đến phần tiền xử lý dữ liệu và EDA để có thể hiểu rõ hơn về bộ dữ liệu mà chúng ta thu thập được

3.2.1 EDA ACB

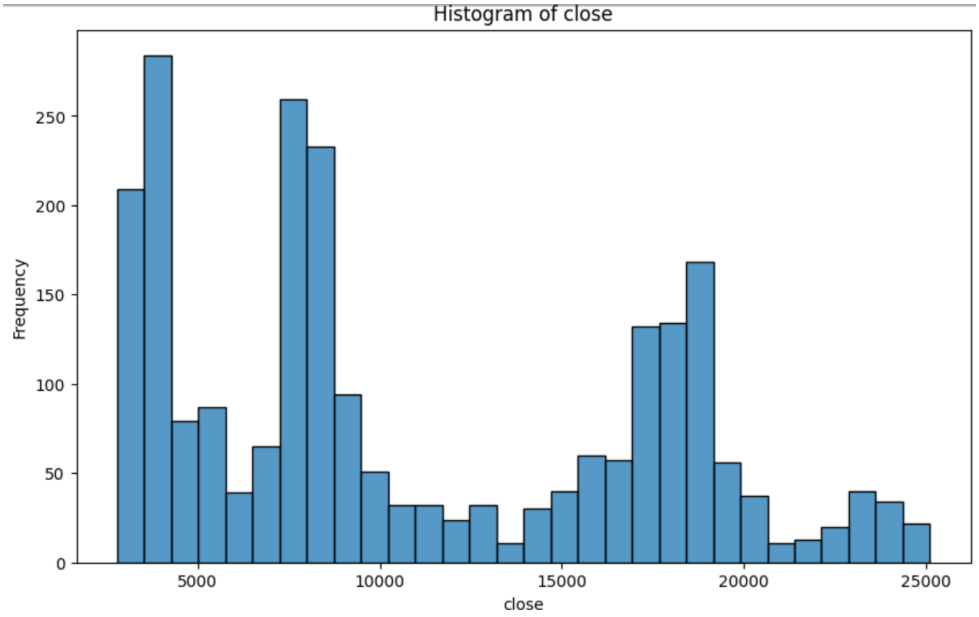


```
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   open    2385 non-null   int32
 1   high    2385 non-null   int32
 2   low     2385 non-null   int32
 3   close   2385 non-null   int32
 4   volume  2385 non-null   int32
 5   ticker  2385 non-null   object
dtypes: int32(5), object(1)
memory usage: 83.8+ KB
None

Data Shape: (2385, 6)

Columns in the DataFrame:
Index(['open', 'high', 'low', 'close', 'volume', 'ticker'], dtype='object')

Data Types:
open       int32
high       int32
low        int32
...
25%     5220.000000   5290.000000   5180.000000   5220.000000  9.133370e+05
50%     8390.000000   8500.000000   8230.000000   8370.000000  2.771983e+06
75%    17400.000000  17540.000000  17190.000000  17400.000000  5.599785e+06
max    25100.000000  25370.000000  24650.000000  25100.000000  3.515600e+07
```

Hình 3.1 Tổng quan bộ dữ liệu ACB

Các giao dịch cổ phiếu của ACB từ ngày 5 tháng 1 năm 2015 đến ngày 26 tháng 7 năm 2024, với tổng cộng 2,385 phiên giao dịch. Các cột dữ liệu bao gồm giá mở cửa (open), giá cao nhất (high), giá thấp nhất (low), giá đóng cửa (close), và khối lượng giao dịch (volume), tất cả đều được lưu trữ dưới dạng số nguyên (int32).

Trong suốt khoảng thời gian này, giá cổ phiếu ACB dao động từ mức thấp nhất là 4,210 đồng đến mức cao nhất là 25,100 đồng, với giá đóng cửa trung bình khoảng 11,031 đồng. Khối lượng giao dịch hàng ngày trung bình là khoảng 3.1 triệu cổ phiếu, nhưng có ngày cao nhất đã đạt tới hơn 35 triệu cổ phiếu.



Hình 3.2 Histogram giá close ACB

Có thể thấy giá close của ACB dao động thường tập trung ở khoảng từ 2000 đến 10000, và dao động ở tần suất cao nhất trong khoảng từ 3000 đến 4000, thấp nhất là 22000 đến 2300. Nhìn vào có thể thấy rõ ràng các cụm mà giá close dao động (2000 đến 5000, 5000 đến 10000, 15000 đến 2000 và cuối cùng là 21000 đến 25000).

Hình 3.3 Giá Close ACB theo thời gian

Nhìn chung thì giá close ACB có xu hướng tăng theo thời gian, và có những mốc chuyển biến mạnh mỗi 2 năm có thể kể đến như tăng mạnh mẽ ở giai đoạn 2018, có phân sụt giảm ở giai đoạn 2020 có thể do đại dịch covid nhưng lại tăng 1 cách vượt bậc ở 2 năm tiếp theo đến 2022. Nhưng sau đó lại có đợt điều chỉnh và giảm mạnh ở năm 2023 nhưng lại ổn định và tiếp tục tăng lại cho đến nay

3.3.2 EDA MWG

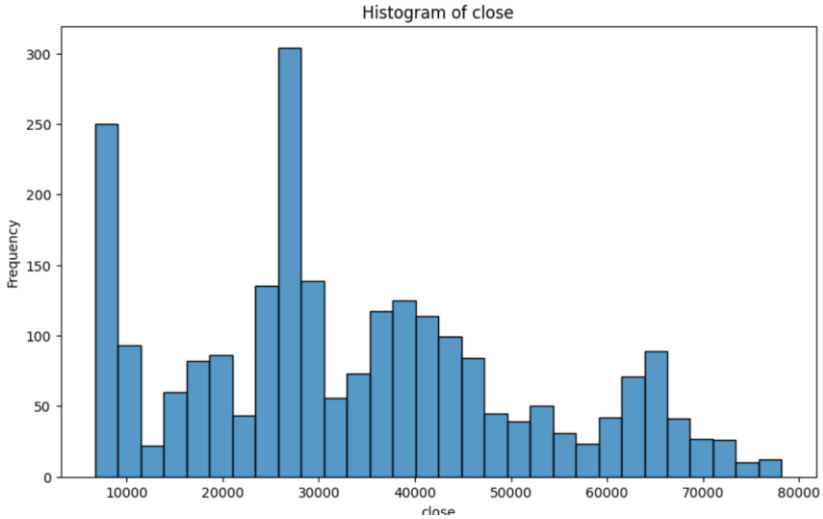
```
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   open    2388 non-null   int32
 1   high    2388 non-null   int32
 2   low     2388 non-null   int32
 3   close   2388 non-null   int32
 4   volume  2388 non-null   int32
 5   ticker  2388 non-null   object
dtypes: int32(5), object(1)
memory usage: 84.0+ KB
None

Data Shape: (2388, 6)

Columns in the DataFrame:
Index(['open', 'high', 'low', 'close', 'volume', 'ticker'], dtype='object')

Data Types:
open       int32
high       int32
low        int32
...
25%    21810.000000  22200.000000  21355.000000  21790.000000  2.670275e+05
50%    30140.000000  30495.000000  29865.000000  30155.000000  6.922500e+05
75%    44650.000000  45210.000000  44020.000000  44665.000000  1.620038e+06
max    78600.000000  79480.000000  77670.000000  78200.000000  2.961670e+07
```

Hình 3.4 MWG Dataset

Dữ liệu này phản ánh giao dịch cổ phiếu của MWG từ ngày 5 tháng 1 năm 2015 đến ngày 26 tháng 7 năm 2024, với tổng cộng 2,388 phiên giao dịch. Tương tự như dữ liệu của ACB, các

cột trong DataFrame này bao gồm giá mở cửa (open), giá cao nhất (high), giá thấp nhất (low), giá đóng cửa (close), và khối lượng giao dịch (volume), tất cả đều là kiểu số nguyên (int32).

Trong suốt giai đoạn này, giá cổ phiếu MWG dao động từ mức thấp nhất là 21,810 đồng đến mức cao nhất là 78,600 đồng, với giá đóng cửa trung bình khoảng 36,334 đồng. Khối lượng giao dịch hàng ngày trung bình là khoảng 800 nghìn cổ phiếu, với ngày cao nhất đạt gần 30 triệu cổ phiếu. Điều này cho thấy MWG đã trải qua những biến động đáng kể về giá trị cổ phiếu, cùng với khối lượng giao dịch lớn, cho thấy sự quan tâm mạnh mẽ từ các nhà đầu tư đối với cổ phiếu này.



Hình 3.5 Histogram MWG

Có thể thấy giá close của MGW dao động thường tập trung ở khoảng từ 21000 đến 45000, và dao động ở tần suất cao nhất trong khoảng từ 28000 đến 29000, thấp nhất là 22000 đến 2300.

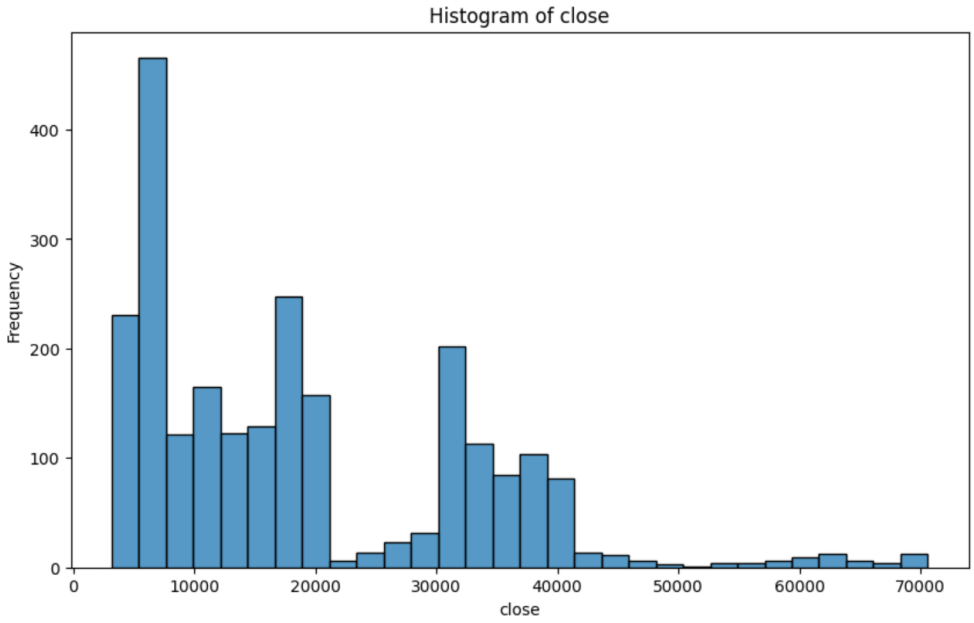
Hình 3.6 Giá Close MWG theo thời gian

Kể từ khi lên sàn ngày 14/07/2014, cổ phiếu MWG luôn giữ xu hướng tăng qua từng năm, nhanh chóng gia nhập nhóm cổ phiếu ngành bán lẻ mạnh nhất trên thị trường. Mức giá cổ phiếu thấp nhất được ghi nhận là 8.900 đồng/cổ phiếu vào ngày 14/07/2014 – ngày đầu tiên phát hành. Đến cuối năm 2020, do ảnh hưởng của dịch bệnh Covid-19 đến ngành bán lẻ nói chung, giá cổ phiếu Thế giới di động giảm mạnh và nhanh chóng chạm đáy. Trước đó, giá cổ phiếu MWG luôn giữ mức trên 50.000đ/cổ phiếu. Bắt đầu từ tháng 11/2019, giá giảm dần và chạm đáy ở mức gần 15000/cổ phiếu ngày 30/3/2020. Giá cổ phiếu MWG vẫn giữ ở mức thấp dưới 40.000đ/cổ phiếu cho đến tháng 9/2020 thì có dấu hiệu tăng trở lại. Mức giá cổ phiếu cao nhất ghi nhận được là gần 80.000đ/cổ phiếu ngày 9/7/2021. Mức giá vẫn giữ ở mức cao trên 560.000đ/cổ phiếu như vậy cho đến giữa tháng 6/2022 thì giảm xuống một nửa còn gần 30.000đ/cổ phiếu. Đây là thời điểm MWG chia cổ tức. Trong thời gian gần đây, giá cổ phiếu MWG đang có xu hướng tăng trở lại.

3.3.3 EDA CMC

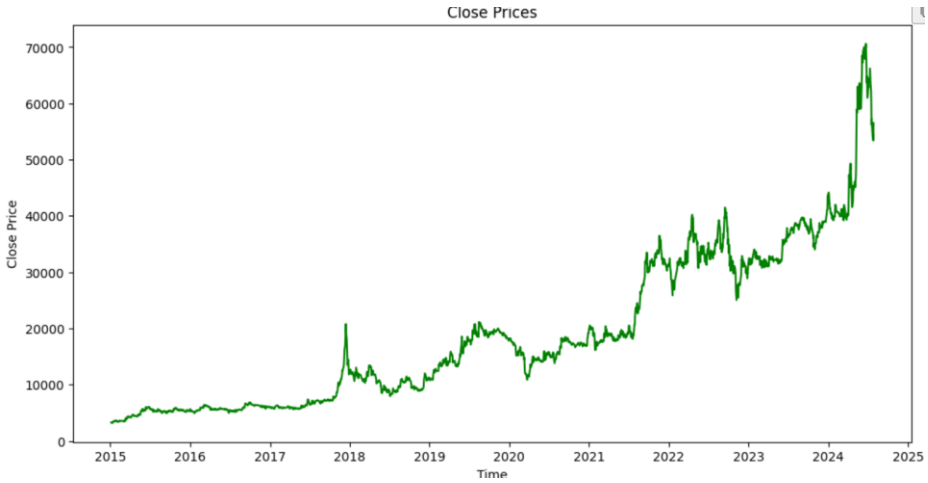
```
DatetimeIndex: 2389 entries, 2015-01-05 to 2024-07-26
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   open    2389 non-null   int32
 1   high    2389 non-null   int32
 2   low     2389 non-null   int32
 3   close   2389 non-null   int32
 4   volume  2389 non-null   int32
 5   ticker  2389 non-null   object
dtypes: int32(5), object(1)
memory usage: 84.0+ KB
None

Data Shape: (2389, 6)

Columns in the DataFrame:
Index(['open', 'high', 'low', 'close', 'volume', 'ticker'], dtype='object')

Data Types:
open       int32
high       int32
low        int32
...
25%     6490.000000   6610.000000   6410.000000   6530.000000  2.090000e+04
50%    15640.000000  15860.000000  15330.000000  15570.000000  4.680000e+04
75%    31400.000000  31840.000000  30820.000000  31500.000000  1.114900e+05
max    71000.000000  74600.000000  70000.000000  70600.000000  4.461800e+06
```

Hình 3.7 CMC Dataset



Hình 3.8 Histogram cổ phiếu CMC

Có thể thấy rằng giá cổ phiếu dao động của ở mức giá thấp từ 5000 đến khoảng 40000 là chủ yếu. Và có sự dao động nhiều vào khoảng giá thấp từ 6000 đến 8000. Ngoài ra các khoảng giá từ 50000 đến 70000 cũng có xuất hiện nhưng với tần số cực kỳ thấp.



Hình 3.9 Giá CMC theo thời gian

Nhìn chung cổ phiếu cmc có xu hướng theo thời gian qua các năm, cùng lắm có những đoạn điều chỉnh mạnh ở giai đoạn 2022 đến 2023. Trong giai đoạn gần đây thì có dấu hiệu suy giảm.

CHAPTER 4. XÂY DỰNG VÀ HUẤN LUYỆN MÔ HÌNH

4.1 ARIMA và biến thể SARIMA

4.1.1 ARIMA

Để phân tích và dự báo giá cổ phiếu của ba công ty MWG, ACB, và CMC, đại diện cho ba ngành bán lẻ, ngân hàng, và công nghệ, mô hình ARIMA (AutoRegressive Integrated Moving Average) đã được sử dụng trên dữ liệu chuỗi thời gian từ 1/1/2015 đến 31/7/2024. ARIMA là một trong những mô hình thống kê phổ biến và hiệu quả nhất để dự đoán dữ liệu chuỗi thời gian, đặc biệt khi dữ liệu có xu hướng hoặc tính mùa vụ rõ ràng. Mô hình được huấn luyện trên dữ liệu từ 1/1/2015 đến 31/12/2023 và được kiểm thử trên dữ liệu từ 1/1/2024 đến 31/7/2024, với mục tiêu xây dựng một mô hình dự báo chính xác giá cổ phiếu trong tương lai dựa trên các yếu tố tự hồi quy (AR), tích hợp (I), và trung bình trượt (MA).

Quá trình xây dựng mô hình bắt đầu bằng việc kiểm định tính dừng của chuỗi thời gian bằng kiểm định Augmented Dickey-Fuller (ADF) để xác định liệu chuỗi có xu hướng hay mùa vụ, từ đó quyết định số lần cần lấy sai phân (d) để đạt tính dừng. Sau đó, các tham số của mô hình ARIMA bao gồm p (bậc của phần tự hồi quy), d (bậc của sai phân), và q (bậc của phần trung bình trượt) được xác định thông qua quá trình thử nghiệm nhiều tổ hợp khác nhau để tìm ra tổ hợp tốt nhất cho mô hình.

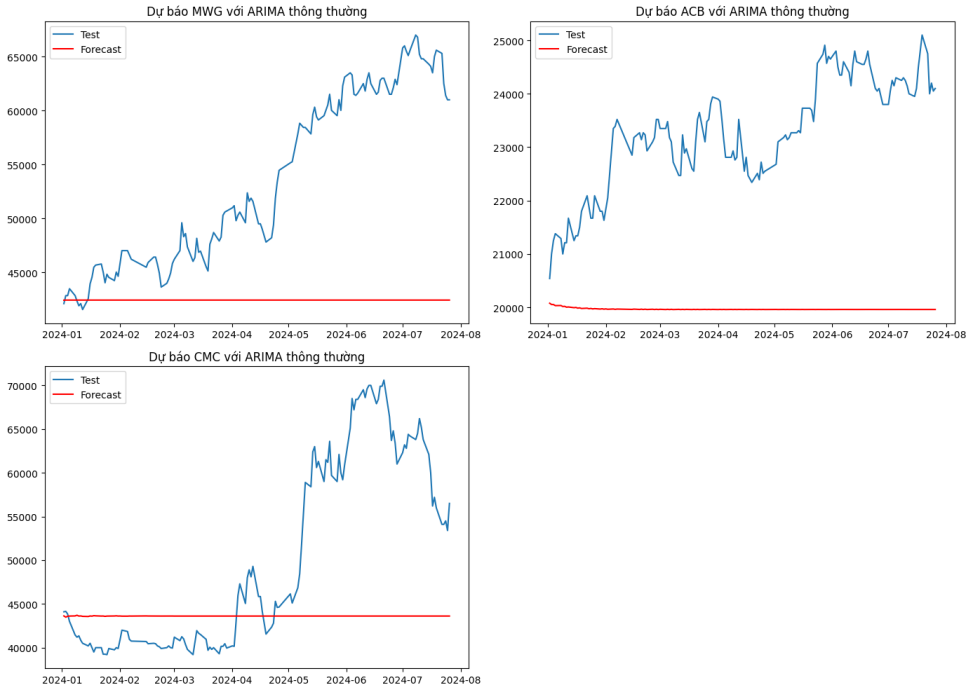
Dữ liệu sau đó được chia thành hai phần: tập huấn luyện (train) từ 1/1/2015 đến 31/12/2023 để mô hình học hỏi và nhận diện các mẫu hình trong dữ liệu, và tập kiểm thử (test) từ 1/1/2024 đến 31/7/2024 để đánh giá độ chính xác của dự báo. Để tối ưu hóa mô hình, phương pháp auto_arima được sử dụng nhằm tự động tìm ra bộ tham số tốt nhất, giúp tiết kiệm thời gian và tăng độ chính xác.

Tiếp theo, mô hình ARIMA được áp dụng với kỹ thuật sliding window, trong đó dữ liệu được chia thành các cửa sổ thời gian di chuyển, giúp mô hình học từ các phân đoạn khác nhau của dữ liệu, đặc biệt hữu ích khi dữ liệu có tính biến động cao. Đồng thời, kỹ thuật expanding window cũng được sử dụng, nơi cửa sổ thời gian được mở rộng dần theo thời gian để mô hình có thể học từ lượng dữ liệu ngày càng lớn, tăng cường khả năng dự báo khi dữ liệu có xu hướng thay đổi theo thời gian.

Bằng cách kết hợp những kỹ thuật này, mô hình ARIMA trở thành một công cụ mạnh mẽ để phân tích và dự báo giá cổ phiếu, đặc biệt khi đối mặt với các dữ liệu có tính chu kỳ và biến động trong dài hạn. Kết quả từ mô hình sẽ được đánh giá trên tập kiểm thử để xác định độ chính xác và tiềm năng ứng dụng trong thực tế.

Bảng 4.1 - Kết quả thực nghiệm với ARIMA thông thường

ARIMA			
	MWG	**ACB**	**CMC**
MAE	11188.04	3306.91	9595.03
MSE	188429769.25	12046450.62	165753828.13
RMSE	13726.97	3470.79	12874.54
MAPE	19.11%	14.03%	16.52%
R2-Score	-1.94	-10.11	-0.40

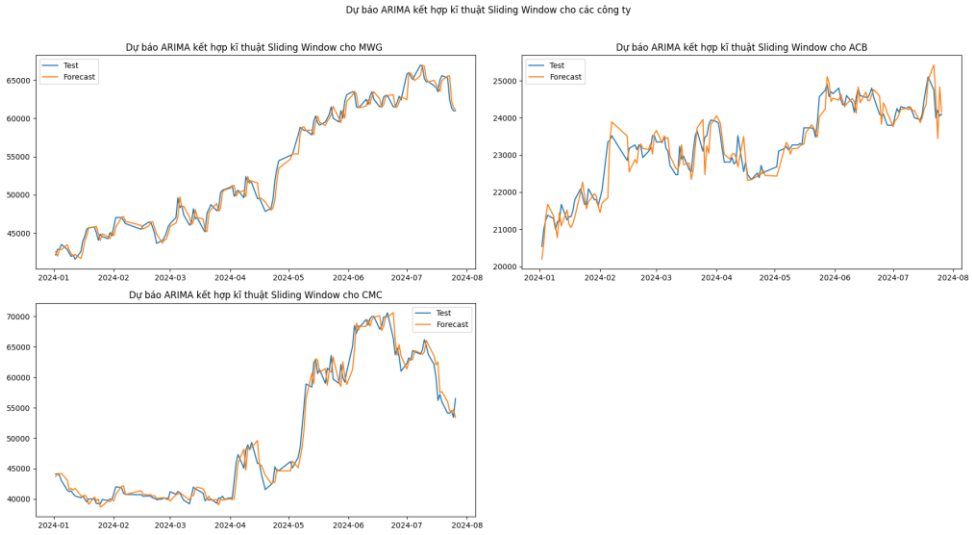


Hình 4.1 Dự báo với ARIMA thông thường

Bảng 4.2 - Kết quả thực nghiệm với ARIMA kết hợp kỹ thuật Sliding Window

ARIMA_SLIDING			
	MWG	**ACB**	**CMC**
MAE	778.48	258.96	1158.29

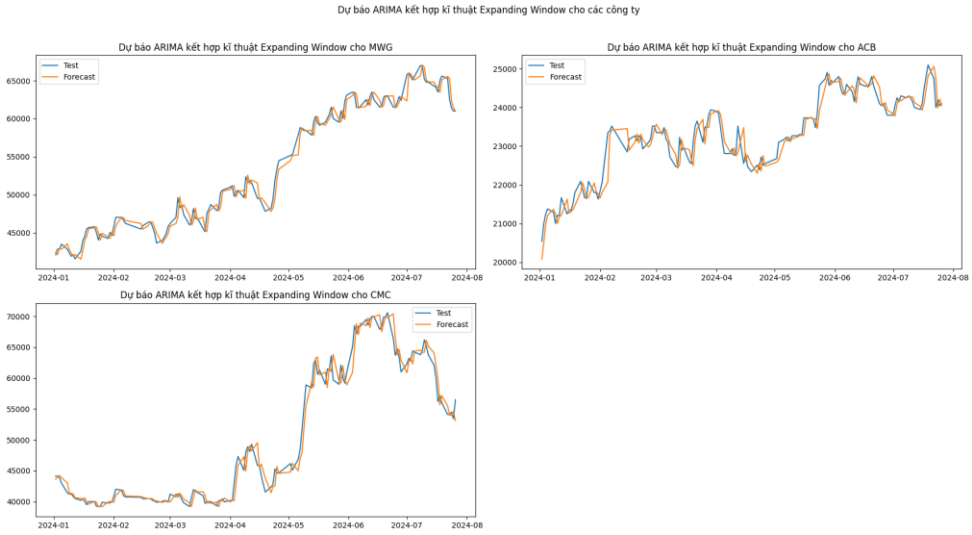
MSE	1040860.24	121403.44	2694080.22
RMSE	1020.22	348.42	1641.36
MAPE	1.46%	1.11%	2.21%
R2-Score	0.98	0.88	0.97



Hình 4.2 Dự báo với ARIMA kết hợp kỹ thuật Sliding Window

Bảng 4.3 - kết quả thực nghiệm với ARIMA kết hợp kỹ thuật Expanding Window

ARIMA_EXPANDING			
	MWG	**ACB**	**CMC**
MAE	775.77	218.99	1137.14
MSE	1042895.45	85794.03	2463070.39
RMSE	1021.22	292.91	1569.41
MAPE	1.46%	0.94%	2.15%
R2-Score	0.98	0.92	0.97



Hình 4.3 Dự báo với ARIMA kết hợp kỹ thuật Expanding Window

Ba phương pháp xây dựng mô hình ARIMA đã được áp dụng để phân tích và dự báo giá cổ phiếu cho ba công ty đại diện cho ba ngành khác nhau: bán lẻ (MWG), ngân hàng (ACB), và công nghệ (CMC). Các phương pháp gồm ARIMA thông thường, ARIMA kết hợp với kỹ thuật Sliding Window và Expanding Window. Kết quả cho thấy, mô hình ARIMA thông thường không mang lại hiệu quả cao với R2-Score âm và các chỉ số sai số cao, phản ánh mô hình này không phù hợp với tính chất dữ liệu được nghiên cứu.

Trong khi đó, việc ứng dụng các kỹ thuật Sliding Window và Expanding Window đã cải thiện đáng kể chất lượng dự báo. Cụ thể, mô hình ARIMA kết hợp Sliding Window đã giảm thiểu được sai số và tăng R2-Score lên mức dương, cho thấy khả năng dự báo chính xác hơn hẳn so với phương pháp thông thường. Tuy nhiên, mô hình ARIMA kết hợp với kỹ thuật Expanding Window lại cho thấy hiệu quả vượt trội nhất với các chỉ số MAE, MSE, và RMSE thấp nhất, cùng R2-Score cao nhất trong ba phương pháp. Điều này chứng tỏ rằng Expanding Window, với ưu điểm là khả năng mở rộng cửa sổ dữ liệu theo thời gian giúp mô hình tiếp cận nhiều thông tin hơn, đã tối ưu hóa khả năng dự báo của mô hình ARIMA, đặc biệt là trong bối cảnh các dữ liệu có tính chu kỳ và biến động cao như giá cổ phiếu.

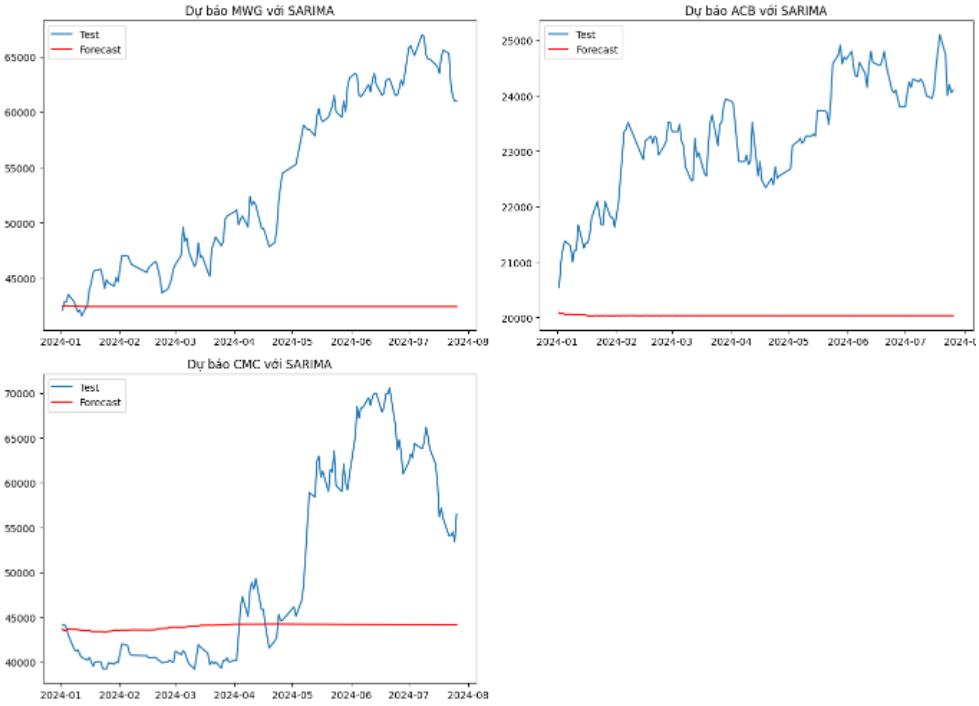
Vì vậy, ARIMA kết hợp với Expanding Window được đề xuất là phương pháp tối ưu nhất trong bài toán dự báo giá cổ phiếu của chúng ta, với khả năng dự báo chính xác cao, đặc biệt trong việc thích ứng với sự thay đổi của dữ liệu qua thời gian. Những kết quả này không chỉ góp phần vào việc lựa chọn mô hình tốt nhất cho dự báo giá cổ phiếu mà còn mở ra hướng tiếp cận mới trong việc sử dụng các kỹ thuật thống kê để cải thiện hiệu quả của mô hình dự báo trong các ngành có tính biến động cao.

4.2.2 SARIMA

Tiếp theo tôi áp dụng mô hình SARIMA cho phép ta mô hình hóa không chỉ xu hướng tổng quát mà còn cả các biến đổi theo mùa, giúp dự báo chính xác hơn khi dữ liệu có yếu tố mùa vụ rõ ràng

Bảng 4.4 - Kết quả thực nghiệm với SARIMA thông thường

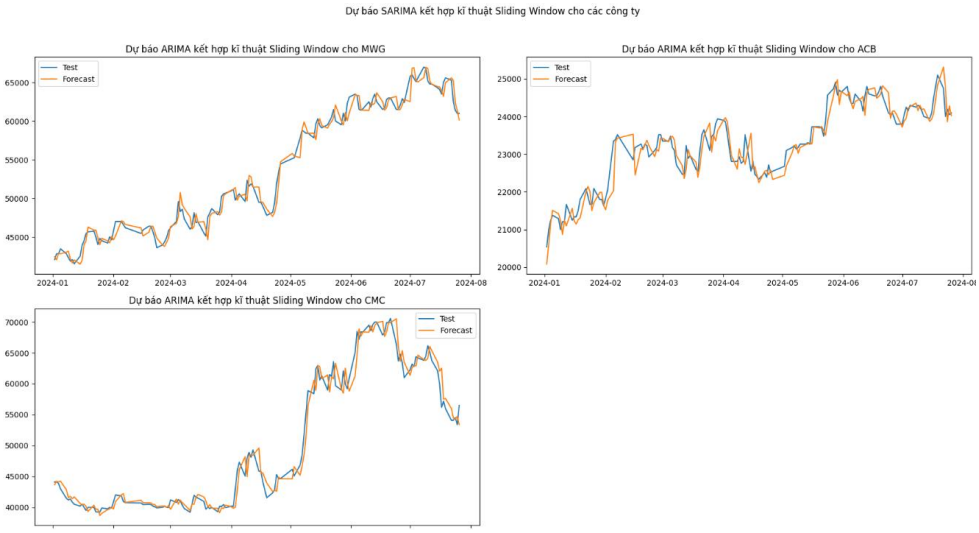
SARIMA			
	MWG	**ACB**	**CMC**
MAE	11224.51	3238.67	9371.58
MSE	189337657.33	11580712.04	157364284.31
RMSE	13760	3403	12544.49
MAPE	19.17%	13.73%	16.18
R2-Score	-1.94	-10.11	-0.40



Hình 4.4 Dự báo với SARIMA bình thường

Bảng 4.5 - Kết quả thực nghiệm với SARIMA kết hợp Sliding Window

SARIMA			
	MWG	**ACB**	**CMC**
MAE	839.62	234.45	1149.73
MSE	1206238.98	99283.9	2670946.31
RMSE	13760	3403	1634.3
MAPE	1.58%	1.01%	2.18%
R2-Score	-1.04	-8.11	-0.50



Hình 4.5 Dự báo với SARIMA kết hợp kỹ thuật Sliding Window

4.2 RNN

SimpleRNN là một mô hình mạng nơ-ron hồi quy đơn giản nhưng hiệu quả trong việc xử lý các vấn đề liên quan đến chuỗi thời gian. Đặc điểm chính của SimpleRNN là khả năng ghi nhớ và duy trì trạng thái của các bước thời gian trước đó thông qua cơ chế hồi quy, giúp mô hình có thể nắm bắt được xu hướng dài hạn trong dữ liệu. Điều này rất quan trọng khi phân tích dữ liệu chuỗi thời gian như giá cổ phiếu, nơi mà các biến động trong quá khứ có thể ảnh hưởng trực tiếp đến giá trị trong tương lai.

Trong quá trình xây dựng mô hình, dữ liệu được chia thành hai giai đoạn chính: giai đoạn huấn luyện từ 1/1/2015 đến 31/12/2023, nhằm giúp mô hình học và nhận diện các mẫu hình giá, và giai đoạn kiểm thử từ 1/1/2024 đến 31/7/2024, dùng để đánh giá khả năng dự đoán của mô hình trong điều kiện thực tế. Kết quả từ mô hình SimpleRNN sẽ được so sánh với các mô hình truyền thống và các phương pháp học sâu khác để đánh giá tính hiệu quả và ứng dụng thực tiễn của nó trong việc phân tích và dự báo giá cổ phiếu.

Quy trình thực hiện

Để xây dựng mô hình SimpleRNN phân tích dữ liệu chuỗi thời gian của giá cổ phiếu từ ba công ty MWG, ACB, và CMC, quy trình bắt đầu với việc lọc ra cột giá đóng cửa (close) từ dữ liệu chứng khoán. Giá đóng cửa là một chỉ số quan trọng, phản ánh tình hình hoạt động của cổ phiếu trong ngày giao dịch và được sử dụng làm cơ sở để dự đoán giá trong tương lai. Tiếp theo, dữ liệu giá đóng cửa này được scale về khoảng (0,1) để phù hợp với mô hình. Việc chuẩn hóa này đảm bảo các giá trị đầu vào nằm trong một khoảng nhất định, giúp mô hình học tốt hơn và tránh tình trạng gradient quá lớn hoặc quá nhỏ trong quá trình huấn luyện. Sau khi chuẩn hóa, dữ liệu được chia thành hai tập: tập huấn luyện (train) từ 1/1/2015 đến 31/12/2023 và tập kiểm thử (test) từ 1/1/2024 đến 31/7/2024. Điều này giúp đánh giá khả năng dự đoán của mô hình trên dữ liệu mới chưa từng được thấy trước đó.

Trong trường hợp đầu tiên, mô hình SimpleRNN cơ bản được xây dựng và huấn luyện với toàn bộ chuỗi thời gian. Dữ liệu đầu vào là toàn bộ chuỗi thời gian của giá đóng cửa đã được scale, và mô hình được thiết kế với 50 đơn vị nơ-ron, sử dụng hàm kích hoạt 'relu', với đầu ra là một giá trị duy nhất đại diện cho dự đoán giá trong tương lai. Mô hình sau đó được huấn luyện với tập huấn luyện và được sử dụng để dự đoán trên tập kiểm thử.

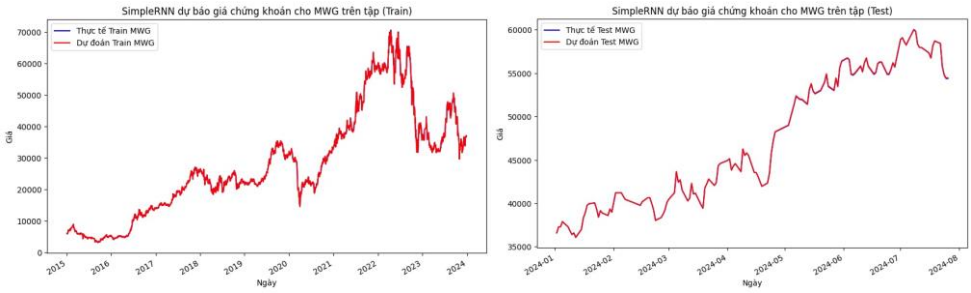
Trong trường hợp thứ hai, mô hình SimpleRNN được kết hợp với kỹ thuật sliding window để tạo ra các chuỗi dữ liệu ngắn hơn, mỗi chuỗi bao gồm một số bước thời gian nhất định (n_steps). Dữ liệu đầu vào được tạo thành các chuỗi con, mỗi chuỗi bao gồm một số bước thời gian n_steps trước đó và giá trị cần dự đoán. Mô hình được thiết kế với 50 đơn vị nơ-ron và đầu vào là các chuỗi có kích thước (n_steps, 1). Sau khi huấn luyện, mô hình sử dụng các chuỗi con từ tập kiểm thử để dự đoán giá trị tương lai.

4.2.1 SimpleRNN dự báo trên tập Test

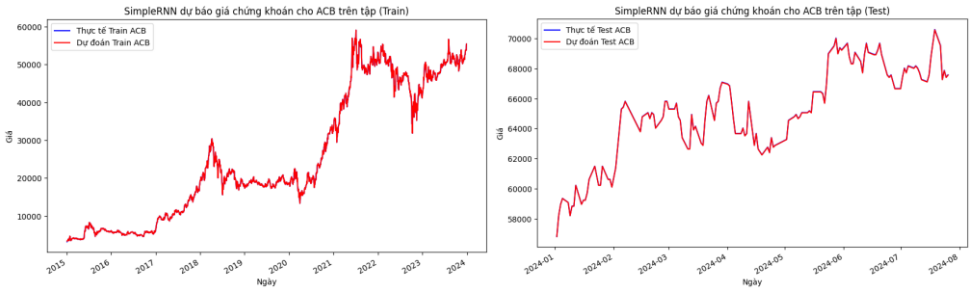
Bảng 4.6 - Kết quả thực nghiệm trường hợp 1:

SimpleRNN dự báo trên tập Test			
	MWG	**ACB**	**CMC**
MAE	21.17	19.71	138.82
MSE	1117.66	432.77	27116.71
RMSE	33.43	20.80	164.67

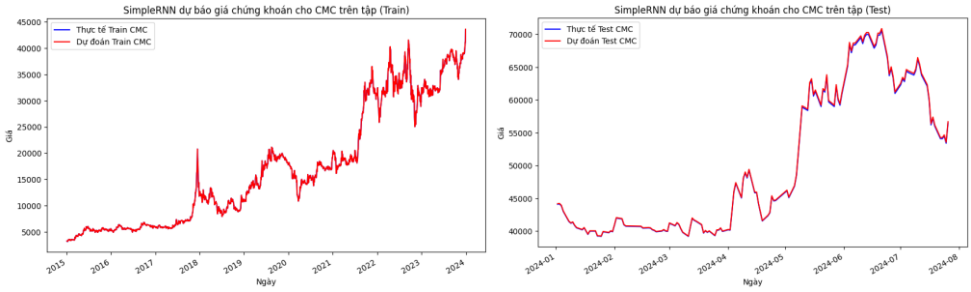
MAPE	0.04 %	0.03%	0.25%
R2-Score	99.98%	99.99%	99.97%



Hình 4.6 MWG với SimpleRNN



Hình 4.7 ACB với SimpleRNN

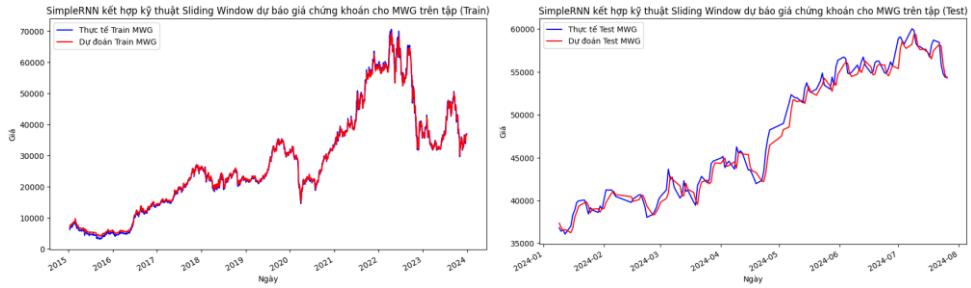


Hình 4.8 CMC với SimpleRNN

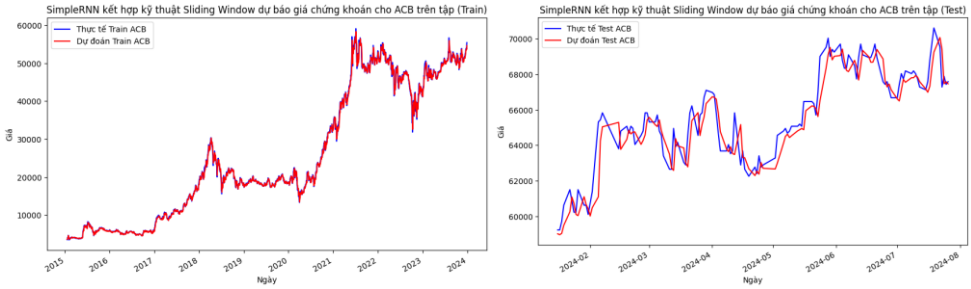
4.2.2 SimpleRNN với kỹ thuật Sliding Window dự báo trên tập Test

Bảng 4.7 - Kết quả thực nghiệm trường hợp 2:

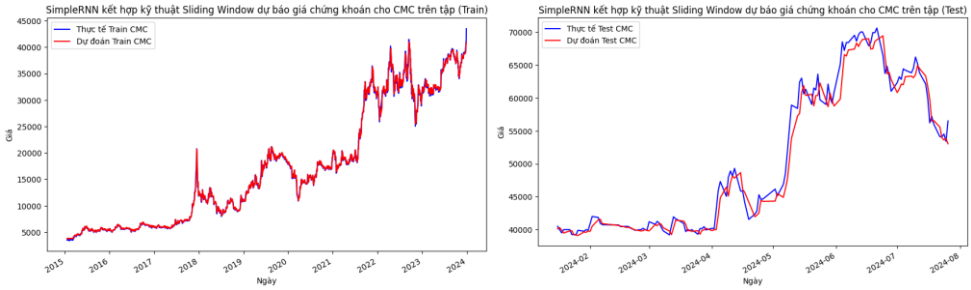
SimpleRNN với kỹ thuật Sliding Window dự báo trên tập Test			
	MWG	**ACB**	**CMC**
MAE	823.07	690.16	138.82
MSE	1141131.53	863453.95	3134674.08
RMSE	1068.23	929.22	1770.50
MAPE	1.73 %	1.05%	2.32%
R2-Score	97.92%	88.14%	97.41%



Hình 4.9 MWG với SimpleRNN kết hợp kỹ thuật Sliding Window



Hình 4.10 ACB với SimpleRNN kết hợp kỹ thuật Sliding Window



Hình 4.11 CMC với SimpleRNN kết hợp kỹ thuật Sliding Window

Mặc dù đã thử nghiệm nhiều lần với các cấu hình và phương pháp khác nhau, mô hình SimpleRNN hiện tại là kết quả tốt nhất sau nhiều lần thực nghiệm. Tuy nhiên, nhìn chung, kết quả này vẫn chưa thực sự ổn định và thỏa mãn.

Trong trường hợp SimpleRNN thông thường (trường hợp 1), mô hình cho thấy hiệu suất khá tốt trên tập huấn luyện và kiểm tra đối với các tập MWG và ACB, nhưng lại xuất hiện dấu hiệu overfitting với tập CMC, khi sai số trên tập kiểm tra tăng mạnh so với tập huấn luyện. Điều này cho thấy mô hình có thể đã học quá mức từ dữ liệu huấn luyện của CMC và không tổng quát hóa tốt khi áp dụng cho dữ liệu mới. Nhưng ta cũng nên xem xét lại là độ sai số này chỉ được coi là tăng mạnh khi trên tập train dự đoán quá chính xác, còn xét về mặt bằng chung thì vẫn ổn, dự báo trên tập test khá tốt.

Ở trường hợp sử dụng kĩ thuật Sliding Window (trường hợp 2), mặc dù mô hình đã được điều chỉnh để cải thiện khả năng dự đoán qua các thử nghiệm, kết quả vẫn chưa đạt được hiệu quả mong muốn. Sự giảm đáng kể của R^2 và sự tăng cao của các chỉ số lỗi trên cả tập huấn luyện và kiểm tra, đặc biệt đối với tập MWG, cho thấy mô hình đang gặp vấn đề với cả overfitting và underfitting. Điều này có thể do bản chất dữ liệu không đủ phức tạp hoặc không phù hợp với kiến trúc RNN đơn giản, hoặc cũng có thể do bản thân kĩ thuật Sliding Window chưa được tối ưu hóa tốt. Một lý do khác mà nhóm nghi ngờ đó là khi áp dụng kỹ thuật sliding window, model sẽ không ghi nhớ trong dài hạn mà chỉ học được trong một chuỗi nhất định, do đó mà kết quả dự đoán trên tập test không được tốt lắm, nhưng nhìn chung là cũng rất ổn rồi.

Tổng hợp lại thì đối với ngữ cảnh này, mô hình RNN thuần khiết đang có một sự chính xác nhỉnh hơn là khi sử dụng kĩ thuật sliding window. Trong tương lai, ta có thể xem xét cập nhật và thử sai nhiều giá trị các tham số hơn để tìm ra model tốt nhất.

4.3 LSTM

Mô hình LSTM (Long Short-Term Memory) thường được lựa chọn cho bài toán dự đoán chuỗi thời gian nhờ vào khả năng ghi nhớ các thông tin từ những bước thời gian trước đó mà không gặp phải vấn đề vanishing gradient như trong các mạng nơ-ron truyền thống. Trong nghiên cứu này, chúng tôi sử dụng một mô hình LSTM đơn giản nhưng hiệu quả, không cần phải xếp chồng nhiều lớp phức tạp. Mô hình LSTM được xây dựng với cấu trúc cơ bản, đủ để nắm bắt

các quy luật từ dữ liệu chuỗi thời gian mà vẫn đảm bảo tính đơn giản và dễ triển khai. Lớp LSTM chính trong mô hình sẽ học từ các chuỗi thời gian và lớp Dense đầu ra sẽ đưa ra dự đoán giá trị cho các bước thời gian tiếp theo.

```
# Xây dựng mô hình LSTM
model = Sequential()

model.add(LSTM(units=50, return_sequences=True,
input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=False))
model.add(Dropout(0.2))

model.add(Dense(units=25))
model.add(Dense(units=1))

# Compile mô hình
model.compile(optimizer='adam', loss='mean_squared_error')
```

Đầu tiên, lớp LSTM đầu tiên được thêm vào với 50 đơn vị nơ-ron (units). Lớp này có thuộc tính return_sequences=True, nghĩa là nó sẽ trả về toàn bộ chuỗi đầu ra của từng bước thời gian, thay vì chỉ trả về đầu ra cuối cùng. Điều này cho phép mô hình truyền toàn bộ thông tin của chuỗi thời gian đến lớp tiếp theo. Kế tiếp, lớp Dropout với tỷ lệ 20% được sử dụng để ngẫu nhiên loại bỏ một số nơ-ron trong quá trình huấn luyện, nhằm giảm thiểu nguy cơ overfitting – tức là mô hình không học quá kỹ các đặc điểm của tập huấn luyện mà có khả năng tổng quát tốt hơn.

Tiếp theo, một lớp LSTM thứ hai cũng được thêm vào với 50 đơn vị nơ-ron, nhưng lần này với return_sequences=False. Điều này có nghĩa là lớp này sẽ chỉ trả về đầu ra cuối cùng của chuỗi thời gian, giúp mô hình tập trung vào thông tin cốt lõi nhất đã được trích xuất từ chuỗi thời gian. Một lớp Dropout khác với tỷ lệ 20% được thêm vào sau lớp LSTM thứ hai để tiếp tục giảm thiểu nguy cơ overfitting.

Cuối cùng, hai lớp Dense được thêm vào mô hình. Lớp Dense đầu tiên có 25 đơn vị nơ-ron, giúp tạo ra một mạng nơ-ron có khả năng học được các đặc điểm phức tạp từ dữ liệu. Lớp Dense thứ hai có 1 đơn vị nơ-ron, đóng vai trò là lớp đầu ra, nơi mô hình đưa ra dự đoán cuối cùng – trong trường hợp này là giá cổ phiếu.

Mô hình được biên dịch (compile) bằng cách sử dụng bộ tối ưu hóa Adam, một phương pháp tối ưu hóa rất phổ biến và hiệu quả trong học sâu, kết hợp với hàm mất mát mean_squared_error, thích hợp cho các bài toán dự đoán chuỗi thời gian khi cần tối thiểu hóa sai số dự đoán bình phương giữa giá trị thực tế và giá trị dự đoán.

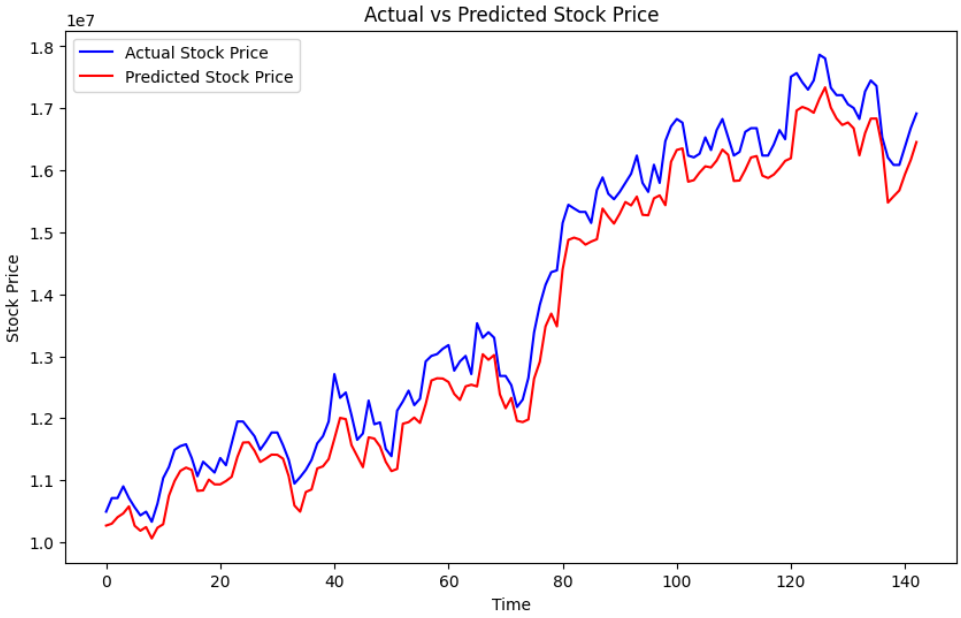
4.3.1 LSTM

Sau khi xây dựng xong mô hình, tôi bắt đầu kiểm thử mô hình bằng các chỉ số đánh giá, dưới đây và kết quả của mô hình chúng ta đã xây dựng ở phía trên.

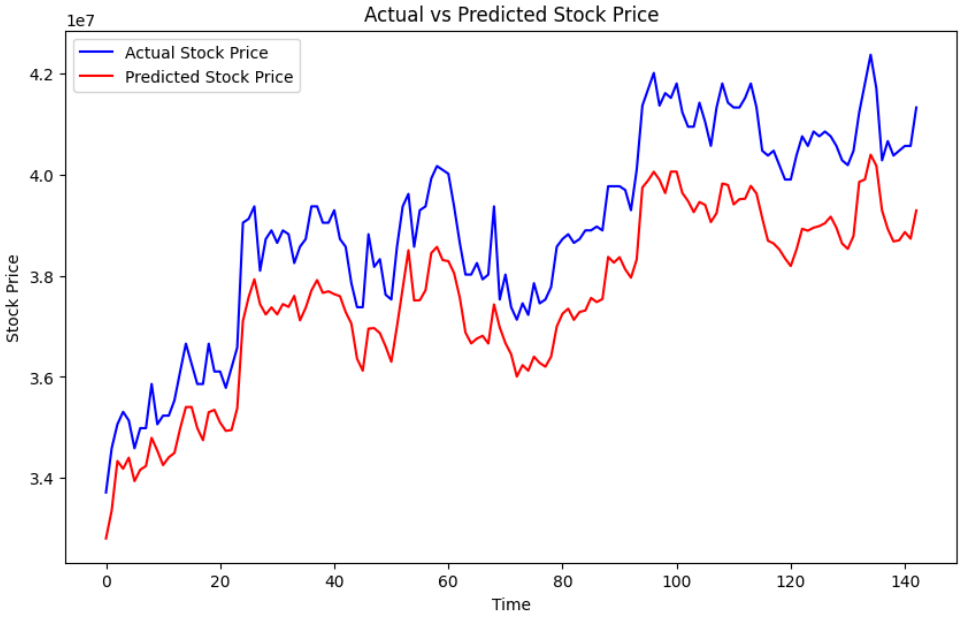
Bảng 4.8 - Kết quả thực nghiệm với mô hình LSTM

	ACB	CMC	MWG
MAE	1424998.521	744.92	449529.8
MSE	2159149395378	1348498	240906408870
RMSE	1469404	1161	490822.18
MAPE	3.629%	2.267%	3.241%
R2-Score	44.703%	99.455%	95.737%

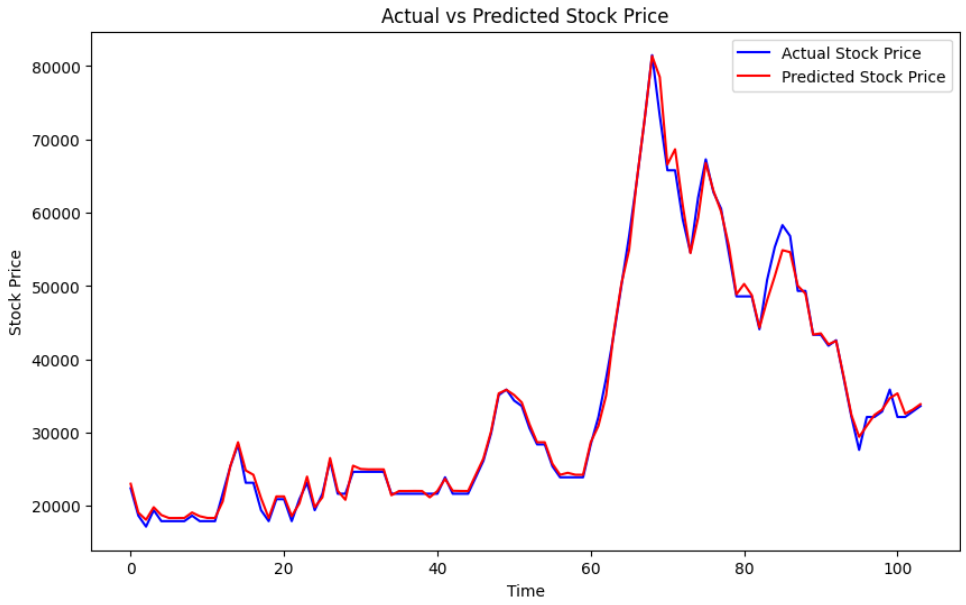
Mô hình LSTM đã cho thấy hiệu quả dự đoán tốt đối với các mã cổ phiếu ACB, CMC và MWG, mặc dù mức độ hiệu quả có sự khác biệt rõ rệt. Đối với mã ACB, mô hình đạt được MAE là 1,424,998.521 và MSE là 2,159,149,395,378, với RMSE là 1,469,404 và MAPE là 3.629%. Chỉ số R2-Score của ACB là 44.703%, cho thấy mô hình có khả năng dự đoán ở mức chấp nhận được, mặc dù vẫn còn một số sai số. Đối với mã CMC, mô hình có MAE là 744.92 và MSE là 1,348,498, với RMSE là 1,161 và MAPE thấp nhất là 2.267%. Chỉ số R2-Score của CMC đạt 99.455%, cho thấy khả năng dự đoán rất chính xác và giải thích hầu như hoàn toàn biến động giá cổ phiếu. Mã MWG cũng đạt kết quả tốt, với MAE là 449,529.8 và MSE là 240,906,408,870, cùng RMSE là 490,822.18 và MAPE là 3.241%. Chỉ số R2-Score của MWG là 95.737%, cho thấy mô hình có khả năng dự đoán cao và giải thích tốt. Nhìn chung, mô hình LSTM hoạt động hiệu quả nhất đối với mã CMC, thể hiện qua sai số thấp và khả năng giải thích cao, nhưng cũng cung cấp dự đoán chính xác cho các mã còn lại.



Hình 4.12 LSTM dự đoán trên tập test cho MWG



Hình 4.13 LSTM dự đoán trên tập test cho ACB



Hình 4.14 LSTM dự đoán trên tập test cho CMC

4.3.2 LSTM và Sliding Window

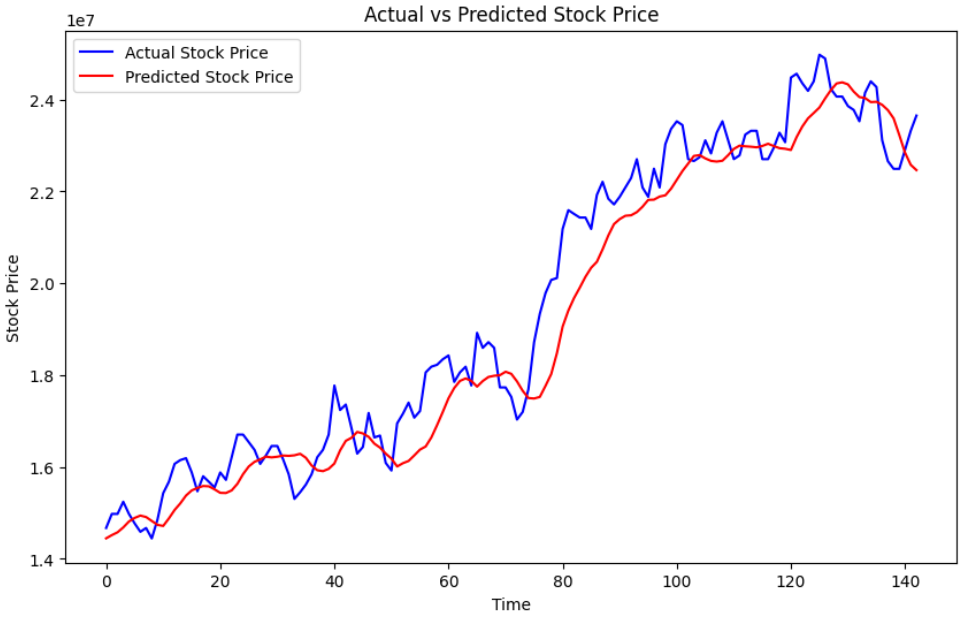
Trong quá trình xây dựng mô hình dự đoán giá cổ phiếu, một yếu tố quan trọng là cách xử lý dữ liệu đầu vào để tối ưu hóa khả năng học của mô hình. Kỹ thuật sliding window đã được áp dụng để tạo ra các tập dữ liệu huấn luyện từ chuỗi thời gian. Cụ thể, sliding window cho phép mô hình sử dụng một cửa sổ thời gian cố định để xem xét một khoảng dữ liệu trước đó và dự đoán giá trị tiếp theo. Khi cửa sổ trượt qua từng bước thời gian, nó tạo ra các tập dữ liệu liên tục, giúp mô hình có khả năng học hỏi từ các mẫu dữ liệu có xu hướng và quy luật lặp lại theo thời gian. Việc sử dụng sliding window với khoảng thời gian 60 ngày trong nghiên cứu này giúp mô hình tập trung vào những biến động ngắn hạn, từ đó nâng cao khả năng dự đoán các thay đổi đột ngột trong dữ liệu tài chính. Dưới đây là kết quả khi sử dụng model ở trên kết hợp với kỹ thuật Sliding Window.

Bảng 4.9 - Kết quả thực nghiệm với mô hình LSTM kết hợp Sliding Window

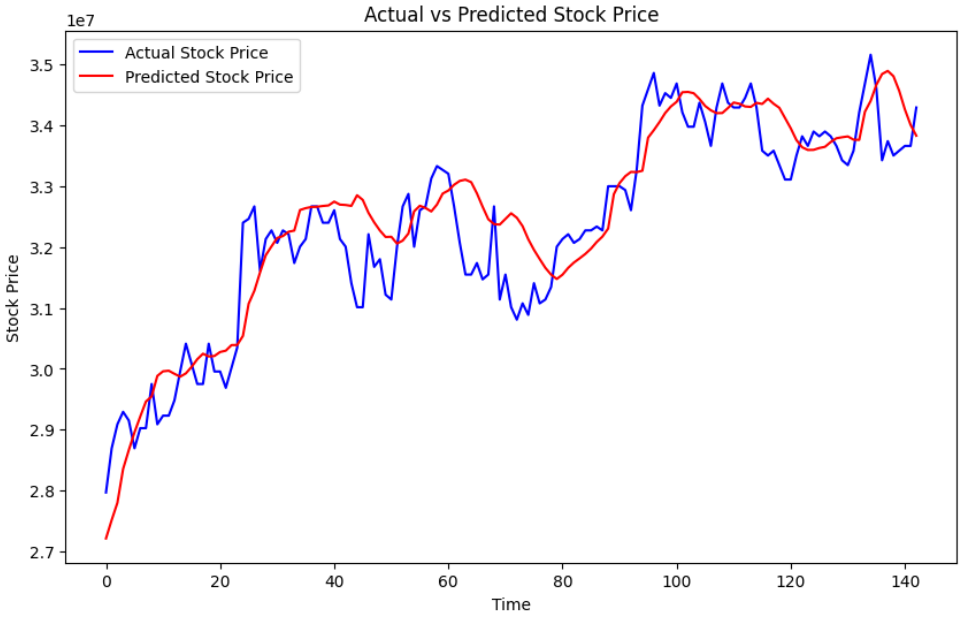
	ACB	CMC	MWG
MAE	554292	3917	657162
MSE	502925104473	31506972	690226306976
RMSE	709172	5613	830798

MAPE	1.735%	10.865%	3.371%
R2-Score	81.288%	87.275%	93.753%

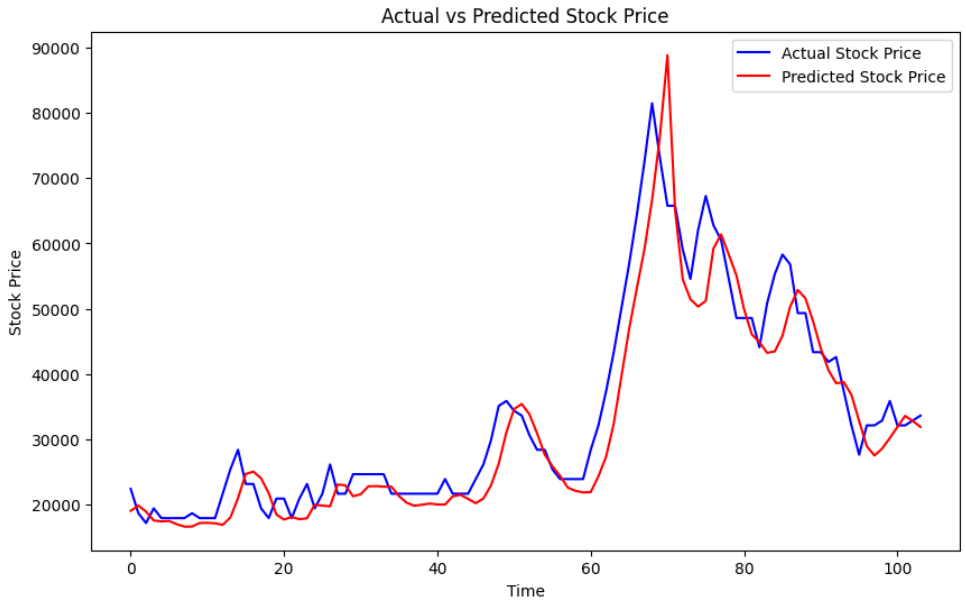
Mô hình LSTM đã cho thấy hiệu quả dự đoán tốt đối với cả ba mã cổ phiếu ACB, CMC, và MWG, nhưng mức độ hiệu quả có sự khác biệt rõ rệt. Đối với mã ACB, mô hình đạt được chỉ số MAPE thấp (1.735%) và R2-Score là 81.288%, cho thấy mô hình có khả năng dự đoán khá chính xác và giải thích được phần lớn biến động của giá cổ phiếu. Tương tự, mô hình cũng thể hiện khả năng dự đoán tốt đối với mã CMC, tuy nhiên, chỉ số MAPE cao hơn (10.865%) và R2-Score đạt 87.275% cho thấy mức độ sai số lớn hơn so với ACB, nhưng vẫn ở mức chấp nhận được. Mã MWG lại có kết quả xuất sắc nhất, với MAPE chỉ 3.371% và R2-Score cao nhất là 93.753%, cho thấy mô hình LSTM có khả năng dự đoán rất chính xác đối với mã này. Nhìn chung, mô hình LSTM mang lại kết quả tốt nhất cho mã MWG, thể hiện qua sai số thấp và khả năng giải thích cao.



Hình 4.15 LSTM sliding dự báo trên tập test cho MWG



Hình 4.16 LSTM sliding dự báo trên tập test cho ACB



Hình 4.17 LSTM sliding dự báo trên tập test cho CMC

4.4 GRU

Commented [3]: @thanhvm21416c@st.uel.edu.vn @khoipcn21416c@st.uel.edu.vn

GRU (Gated Recurrent Unit) là một biến thể tiên tiến hơn của mạng nơ-ron hồi quy, được thiết kế để khắc phục một số hạn chế của SimpleRNN, đặc biệt là vấn đề về "vanishing gradient" khi xử lý các chuỗi thời gian dài. Đặc điểm chính của GRU là cơ chế cổng (gating mechanism), cho phép mô hình tự động học cách duy trì hoặc quên đi thông tin không cần thiết trong các bước thời gian trước đó. Điều này giúp GRU không chỉ nắm bắt được xu hướng dài hạn trong dữ liệu mà còn quản lý tốt hơn các thông tin liên quan trong chuỗi thời gian, làm cho mô hình này trở nên hiệu quả hơn trong việc phân tích và dự đoán giá cổ phiếu.

Quá trình xây dựng mô hình GRU tương tự như với SimpleRNN, trong đó dữ liệu được chia thành hai giai đoạn chính: giai đoạn huấn luyện từ 1/1/2015 đến 31/12/2023, nhằm giúp mô hình học và nhận diện các mẫu hình giá, và giai đoạn kiểm thử từ 1/1/2024 đến 31/7/2024, dùng để đánh giá khả năng dự đoán của mô hình trong điều kiện thực tế. Kết quả từ mô hình GRU sẽ được so sánh với các mô hình truyền thống và các phương pháp học sâu khác để đánh giá tính hiệu quả và ứng dụng thực tiễn của nó trong việc phân tích và dự báo giá cổ phiếu.

Tóm lại, việc sử dụng GRU thay vì SimpleRNN có thể giúp cải thiện khả năng dự đoán của mô hình trong các nhiệm vụ phân tích dữ liệu chuỗi thời gian phức tạp như dự báo giá cổ phiếu.

4.4.1 Xây dựng Mô hình

```
# Build GRU model function
def build_gru_model(input_shape):
```

```
    model = Sequential()
    model.add(GRU(units=50,              return_sequences=True,
input_shape=input_shape))
    model.add(Dropout(0.2))
    model.add(GRU(units=50, return_sequences=False))
    model.add(Dropout(0.2))
    model.add(Dense(units=25))
    model.add(Dense(units=1))
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model
```

Hàm build_gru_model được thiết kế để xây dựng mô hình GRU (Gated Recurrent Unit) nhằm giải quyết các bài toán dự đoán chuỗi thời gian, như dự báo giá cổ phiếu. Mô hình này bao gồm hai lớp GRU, mỗi lớp có 50 đơn vị (units), giúp nắm bắt các mẫu hình trong dữ liệu chuỗi thời gian. Sau mỗi lớp GRU là một lớp Dropout với tỷ lệ 0.2, giúp giảm thiểu hiện tượng overfitting bằng cách ngẫu nhiên loại bỏ một số neuron trong quá trình huấn luyện. Sau đó, một lớp Dense với 25 đơn vị được sử dụng để học các mối quan hệ phi tuyến từ các đặc trưng đã được trích xuất, trước khi đầu ra cuối cùng được tạo ra thông qua một lớp Dense khác với 1 đơn vị. Mô hình được biên dịch bằng bộ tối ưu hóa 'adam' và hàm mất mát 'mean_squared_error', đảm bảo quá trình huấn luyện được thực hiện một cách hiệu quả và chính xác.

4.4.2 GRU dự báo trên tập Test

Sau khi xây dựng xong mô hình, tôi bắt đầu kiểm thử mô hình bằng các chỉ số đánh giá, dưới đây và kết quả của mô hình chúng ta đã xây dựng ở phía trên.

Bảng 4.10 - Kết quả thực nghiệm trên tập test với mô hình GRU

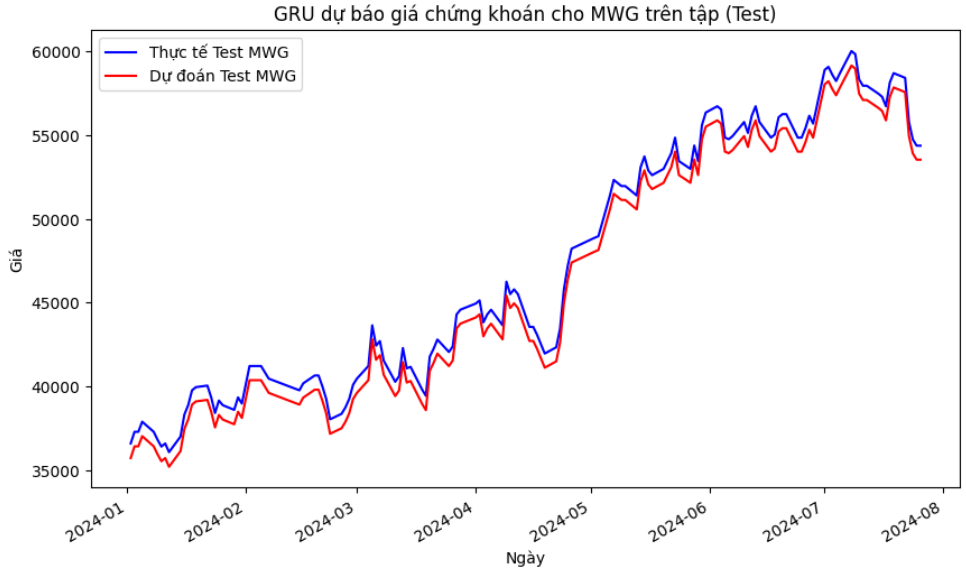
	ACB	CMC	MWG
MAE	1435.9927	1544.6443	1436.2068
MSE	2090087.0107	2646313.7551	2096030.9163
RMSE	1445.7133	1626.7494	1447.76756
MAPE	2.1989	2.9881	3.0452
R2-Score	0.7888	0.9776	0.9631

Trong quá trình thử nghiệm dự báo giá chứng khoán sử dụng mô hình GRU (Gated Recurrent Unit), các kết quả đã chỉ ra rằng mô hình này có khả năng dự đoán tương đối chính xác xu hướng của thị trường chứng khoán. Cụ thể, biểu đồ dự báo giá cổ phiếu cho các công ty như ACB, CMC, và MWG đã thể hiện rõ rằng đường dự đoán (màu đỏ) thường bám sát khá tốt với đường giá trị thực tế (màu xanh) ở các giai đoạn quan trọng.

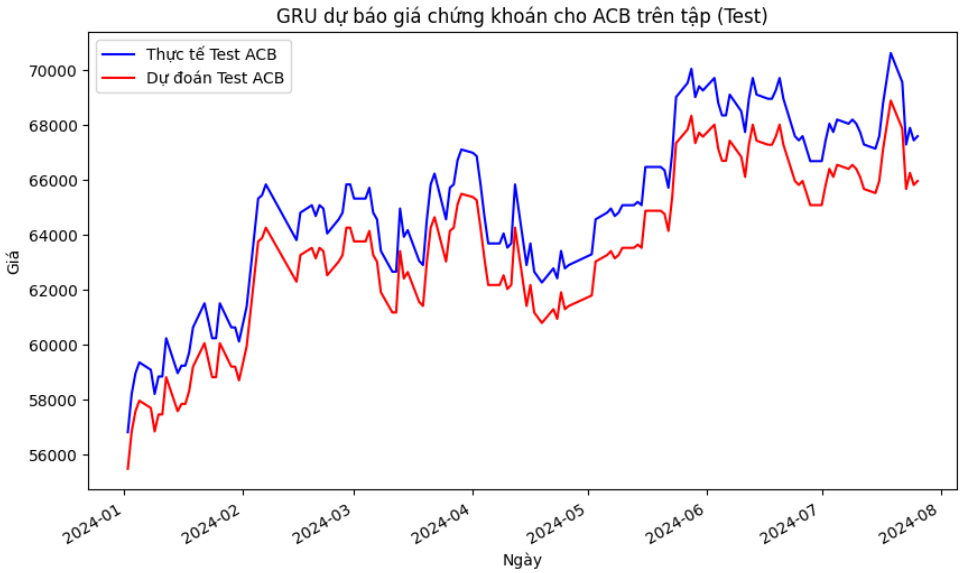
Mặc dù có những biến động đột ngột trong giá cổ phiếu, mô hình GRU vẫn thể hiện khả năng duy trì độ chính xác tương đối cao, với sự chênh lệch giữa dự báo và thực tế không quá lớn. Điều này cho thấy GRU là một mô hình có thể ứng dụng tốt trong việc dự báo các chuỗi thời gian có tính biến động cao như giá cổ phiếu.
Để đánh giá chi tiết hơn, bảng kết quả thực nghiệm đã được trình bày, cho thấy các chỉ số đánh giá như MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAPE (Mean Absolute Percentage Error), và R2-Score. Kết quả cho thấy mô hình GRU đã đạt được các chỉ số khá ấn tượng:

- MAE: Các giá trị MAE cho thấy độ sai lệch trung bình giữa giá dự đoán và giá thực tế là khá thấp, đặc biệt là đối với các cổ phiếu ACB (1435.9927) và MWG (1436.2068), cho thấy mô hình GRU có khả năng dự báo chính xác.
- MSE và RMSE: Mô hình GRU có mức MSE và RMSE thấp, với RMSE của ACB là 1445.7133 và của MWG là 1447.76756, chỉ ra rằng mô hình có khả năng hạn chế lỗi dự đoán.
- MAPE: Chỉ số MAPE cho tất cả các cổ phiếu đều dưới 4%, với giá trị thấp nhất là 2.1989% đối với ACB, thể hiện mức độ chính xác cao trong các dự báo của GRU.
- R2-Score: R2-Score cao, đặc biệt là đối với CMC (0.9776), chứng tỏ mô hình GRU có khả năng giải thích tốt sự biến thiên của giá cổ phiếu, điều này rất quan trọng trong các dự báo tài chính.

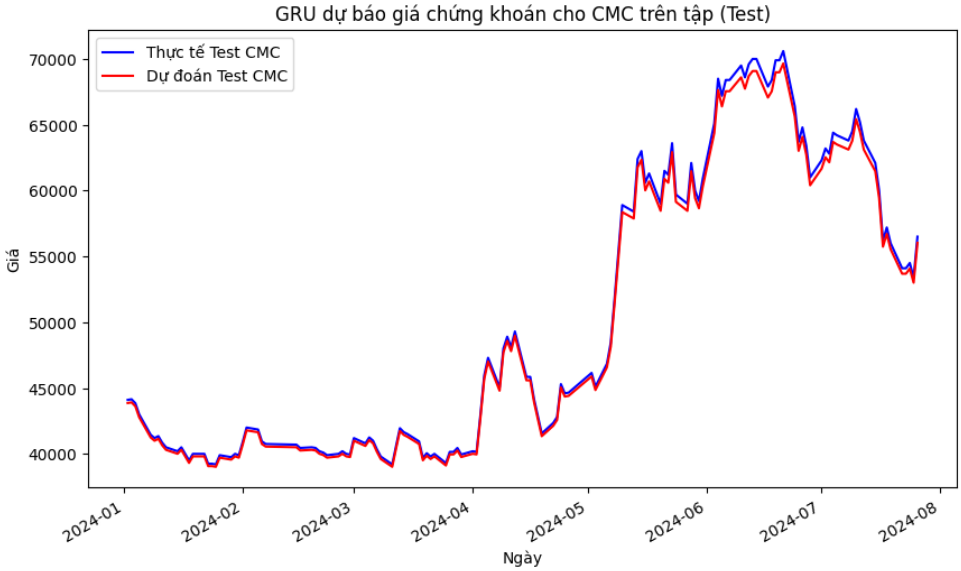
Tổng hợp lại, mô hình GRU đã chứng minh được hiệu quả và tính chính xác của nó trong việc dự báo giá cổ phiếu. Với các kết quả ấn tượng từ MAE, MSE, RMSE đến R2-Score, GRU không chỉ phù hợp mà còn là một lựa chọn tối ưu cho các bài toán dự báo chuỗi thời gian, đặc biệt trong bối cảnh thị trường chứng khoán có nhiều biến động phức tạp.



Hình 4.18 GRU dự báo trên tập test cho MWG



Hình 4.19 GRU dự báo trên tập test cho ACB



Hình 4.20 GRU dự báo trên tập test cho CMC

4.4.3 GRU dự báo trên tập Test kết hợp Sliding Window

Bảng 4.11 - Kết quả thực nghiệm trên tập test với mô hình GRU kết hợp Sliding Window

	ACB	CMC	MWG
MAE	1205.4378	1895.6009	1695.4367
MSE	2324392.2016	6618835.6238	4238675.7346
RMSE	1524.5957	2572.7097	2058.80444
MAPE	1.8254	3.5491	3.4982
R2-Score	0.6809	0.9454	0.9229

Trong quá trình thử nghiệm dự báo giá chứng khoán với mô hình GRU kết hợp với kỹ thuật Sliding Window, các kết quả cho thấy mô hình này có hiệu quả tương đối cao trên các tập dữ liệu khác nhau, bao gồm ACB, CMC, và MWG. Dưới đây là các phân tích chi tiết dựa trên các chỉ số đánh giá chính như MAE, MSE, RMSE, MAPE và R2-Score.

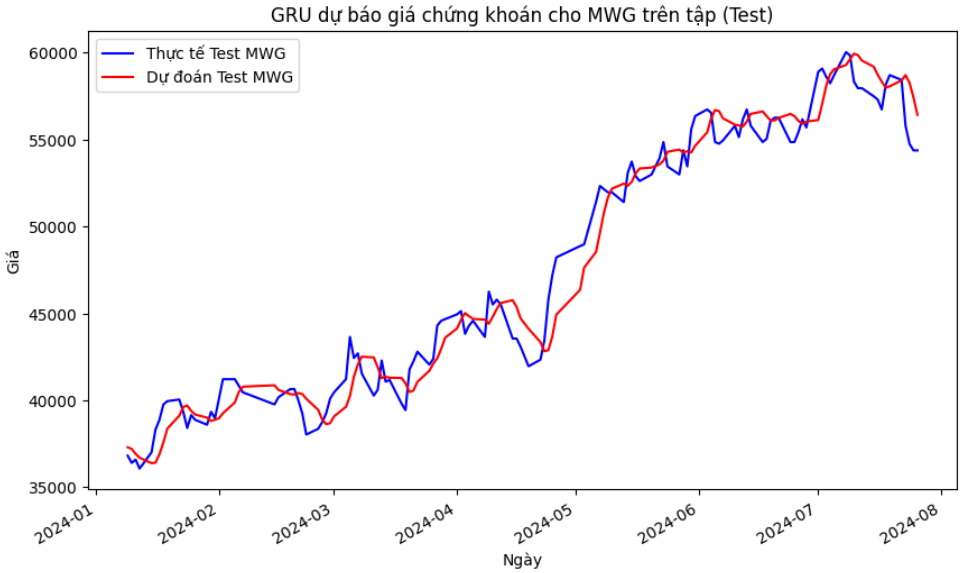
Hiệu suất của mô hình GRU:

- MAE (Mean Absolute Error): Mô hình GRU đạt giá trị MAE thấp nhất đối với ACB (1205.4378), cho thấy độ sai lệch trung bình giữa giá trị dự đoán và giá trị thực tế là nhỏ. Trong khi đó, CMC có giá trị MAE cao nhất (1895.6009), gợi ý rằng mô hình có thể gặp khó khăn trong việc dự báo chính xác giá cổ phiếu CMC.
- MSE (Mean Squared Error) và RMSE (Root Mean Squared Error): Đối với chỉ số MSE, ACB có giá trị thấp nhất (2324392.2016), trong khi CMC có giá trị cao nhất (6618835.6238). Điều này đồng nghĩa với việc mô hình dự báo chính xác hơn với ACB và kém hơn đối với CMC. Tương tự, RMSE cũng cho thấy sự biến động tương tự với ACB có RMSE là 1524.5957 và CMC là 2572.7097, khẳng định thêm về độ chính xác dự báo của mô hình GRU đối với các cổ phiếu khác nhau.
- MAPE (Mean Absolute Percentage Error): Chỉ số MAPE, một chỉ số quan trọng để đánh giá tỷ lệ phần trăm sai lệch giữa dự báo và thực tế, cho thấy giá trị thấp nhất đối với ACB (1.8254%) và cao nhất đối với CMC (3.5491%). Điều này cho thấy mô hình có thể dự đoán tương đối chính xác đối với ACB, trong khi dự báo cho CMC gặp nhiều khó khăn hơn.
- R2-Score: R2-Score, chỉ số đánh giá mức độ mà mô hình có thể giải thích được sự biến động của dữ liệu, cho thấy kết quả khả quan nhất đối với CMC (0.9454), trong khi ACB có giá trị R2 thấp nhất (0.6809). Điều này cho thấy mô hình GRU có khả năng giải thích khá tốt sự biến động của CMC, nhưng lại gặp khó khăn hơn với ACB.

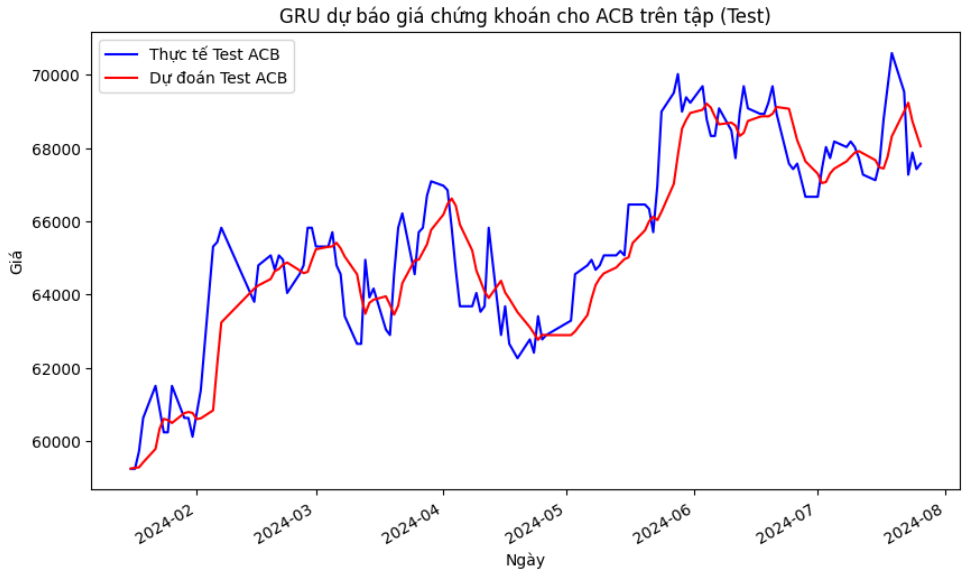
Mô hình GRU kết hợp với kỹ thuật Sliding Window đã cho thấy những kết quả tích cực trong dự báo giá cổ phiếu, đặc biệt là đối với tập dữ liệu CMC, nơi mô hình đạt R2-Score cao nhất (0.9454). Tuy nhiên, sự khác biệt trong các chỉ số như MAE, MSE, và MAPE giữa các cổ phiếu cho thấy rằng mô hình GRU hoạt động tốt với một số cổ phiếu cụ thể hơn là với các cổ phiếu

khác. Đặc biệt, mô hình này tỏ ra ít chính xác hơn khi áp dụng cho ACB, với R2-Score thấp nhất là 0.6809.

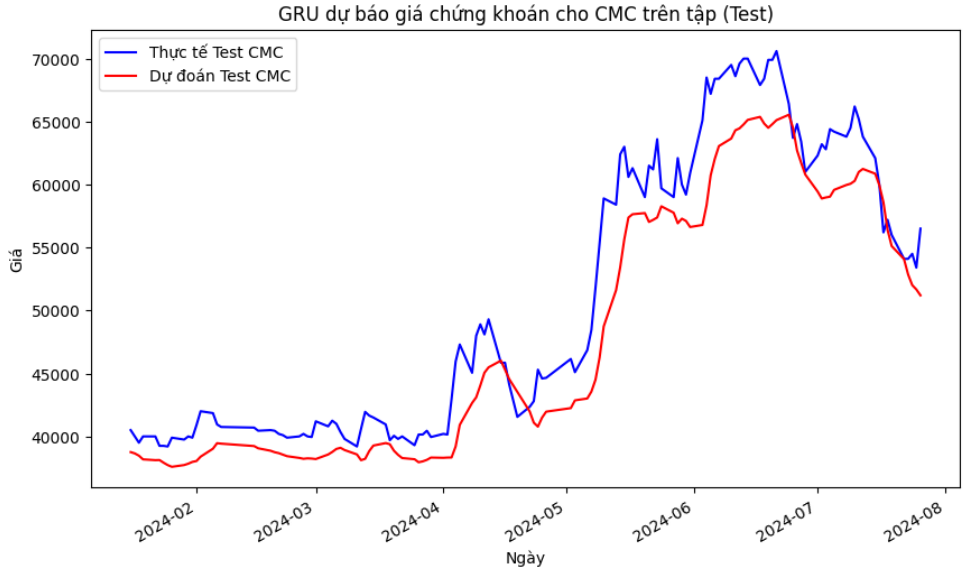
Nhìn chung, kết quả thử nghiệm chỉ ra rằng mô hình GRU có tiềm năng lớn trong việc dự báo giá cổ phiếu, nhưng có thể cần điều chỉnh thêm các tham số hoặc cải thiện kỹ thuật tiền xử lý dữ liệu để nâng cao hiệu quả dự báo trên các tập dữ liệu khác nhau. Điều này là cần thiết để đảm bảo tính ổn định và độ chính xác cao hơn trong các ứng dụng thực tế.



Hình 4.21 GRU Sliding dự báo trên tập test cho MWG



Hình 4.22 GRU Sliding dự báo trên tập test cho ACB



Hình 4.23 GRU Sliding dự báo trên tập test cho CMC

CHAPTER 5. ĐÁNH GIÁ VÀ SO SÁNH HIỆU NĂNG CÁC MÔ HÌNH

5.1 Giới thiệu một số chỉ số đánh giá

Khi phân tích dữ liệu chuỗi thời gian và so sánh hiệu năng giữa các mô hình, việc sử dụng các chỉ số đánh giá phù hợp là vô cùng quan trọng. Những chỉ số này không chỉ giúp đánh giá mức độ chính xác của dự báo mà còn cung cấp thông tin chi tiết về hiệu quả và khả năng áp dụng của mô hình trong thực tế. Một mô hình tốt cần dự báo chính xác, ổn định, dễ triển khai và có khả năng tổng quát hóa tốt khi gặp dữ liệu mới.

Trong phân tích dữ liệu chuỗi thời gian, các chỉ số đánh giá chủ yếu tập trung vào việc đo lường sự khác biệt giữa giá trị thực tế và giá trị dự báo. Những sai số này có thể do nhiều yếu tố gây ra, chẳng hạn như tính phức tạp của dữ liệu, độ dài của chuỗi thời gian, sự hiện diện của nhiễu hoặc các yếu tố ngoại cảnh không lường trước. Do đó, việc lựa chọn các chỉ số đánh giá phải đảm bảo rằng chúng phản ánh chính xác các đặc điểm của dữ liệu cũng như hiệu suất của mô hình.

Các chỉ số như Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), và R^2 (Coefficient of Determination) thường được sử dụng trong phân tích chuỗi thời gian. Những chỉ số này đo lường sự khác biệt giữa giá trị dự báo và giá trị thực tế, giúp đánh giá mức độ sai số trong các dự báo liên tục. Mỗi chỉ số có ưu và nhược điểm riêng, cung cấp những góc nhìn khác nhau về hiệu năng của mô hình.

- MSE, RMSE, MAE: Đánh giá mức độ sai số giữa giá trị dự báo và giá trị thực tế, với MSE và RMSE nhạy cảm hơn với các sai số lớn, trong khi MAE cung cấp một cái nhìn trung bình về sai số.
- MAPE: Đánh giá sai số dự báo dưới dạng phần trăm, giúp dễ dàng so sánh giữa các chuỗi thời gian khác nhau.

- R^2: Đo lường tỷ lệ phương sai của dữ liệu thực tế được giải thích bởi mô hình dự báo, với giá trị từ 0 đến 1, càng gần 1 càng tốt.

Ngược lại, các chỉ số như Precision, Recall, và F1-score thường được sử dụng trong các bài toán phân loại, nơi mục tiêu là phân loại đúng các đối tượng vào các nhóm cụ thể. Các chỉ số này đánh giá hiệu suất của mô hình trong việc phân biệt giữa các lớp khác nhau, chẳng hạn như xác định các trường hợp dương tính hoặc âm tính trong dữ liệu y tế. Tuy nhiên, chúng không phù hợp để đánh giá hiệu suất của các mô hình dự báo chuỗi thời gian, nơi mà các chỉ số sai số trực tiếp như MSE, RMSE, MAE, MAPE, và R^2 mới phản ánh đúng mức độ chính xác và hiệu quả của các dự báo liên tục.

Vì vậy, để đánh giá hiệu suất của các mô hình dự báo chuỗi thời gian một cách toàn diện và chính xác, chúng ta sẽ tập trung vào các chỉ số sai số trực tiếp như MSE, RMSE, MAE, MAPE, và R^2, vốn là những công cụ đo lường hiệu quả nhất cho các dự báo liên tục.

- **Mean Squared Error (MSE):**

Công thức: $MSE = \frac{1}{n}\sum_{i=1}^{n} ((y_i - \hat{y_i})^2$

Ý nghĩa: MSE là một chỉ số đo lường sai số trung bình bình phương giữa giá trị thực tế và giá trị dự báo. Chỉ số này đặc biệt nhạy cảm với các sai số lớn, do đó, nó có thể giúp phát hiện ra các dự báo cực kỳ sai lệch. Một mô hình với MSE nhỏ thường cho thấy sự chính xác cao, tuy nhiên, MSE cũng có thể bị ảnh hưởng bởi các ngoại lệ (outliers).

Ứng dụng: MSE thường được sử dụng khi cần đánh giá tổng thể mức độ sai số của mô hình, đặc biệt trong các trường hợp mà sai số lớn không được chấp nhận.

- **Root Mean Squared Error (RMSE):**

Công thức: $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y_i})^2}$

Ý nghĩa: RMSE là căn bậc hai của MSE, cung cấp thông tin về độ lớn của sai số trung bình và được biểu diễn cùng đơn vị với dữ liệu đầu vào. RMSE nhấn mạnh hơn vào các sai số lớn, và do đó, nó hữu ích khi ta muốn các sai số lớn có tác động mạnh hơn trong quá trình đánh giá mô hình.

Ứng dụng: RMSE thường được sử dụng khi độ lớn của sai số có ý nghĩa quan trọng, chẳng hạn như trong dự báo tài chính, nơi mà các sai số lớn có thể dẫn đến tổn thất nghiêm trọng.

- **Mean Absolute Error (MAE):**

Công thức: $MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}|$

Ý nghĩa: MAE đo lường sai số trung bình tuyệt đối giữa giá trị thực tế và giá trị dự báo. Không giống như MSE và RMSE, MAE không phóng đại ảnh hưởng của các sai số lớn, mà phản ánh một cách trung thực sai số trung bình của mô hình.

Ứng dụng: MAE là lựa chọn tốt khi cần một chỉ số đơn giản và dễ hiểu để đánh giá sai số tổng thể mà không bị ảnh hưởng bởi các giá trị ngoại lệ. Nó đặc biệt hữu ích trong các trường hợp mà sai số lớn không quá quan trọng.

- **Mean Absolute Percentage Error (MAPE):**

Công thức: $MAPE = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right|$

Ý nghĩa: MAPE đo lường sai số dưới dạng phần trăm so với giá trị thực tế, giúp dễ dàng so sánh mức độ sai số giữa các chuỗi thời gian hoặc các mô hình khác nhau. MAPE rất hữu ích khi ta cần hiểu rõ về tỷ lệ sai số so với giá trị thực, đặc biệt trong các lĩnh vực mà việc đánh giá dự báo theo tỷ lệ phần trăm là quan trọng.

Ứng dụng: MAPE thường được sử dụng trong các trường hợp mà tính chính xác tương đối là quan trọng, chẳng hạn như trong phân tích kinh doanh hoặc các dự báo tài chính. Tuy nhiên, MAPE có thể bị ảnh hưởng bởi các giá trị nhỏ của biến dự báo, dẫn đến các phần trăm sai số rất lớn.

- **Coefficient of Determination (R^2):**

Công thức: $R^2 = 1 - \frac{\sum_{i=1}^{n} ((y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} ((y_i - \underline{y})^2}$

Ý nghĩa: R^2 đo lường tỷ lệ phương sai của biến phụ thuộc được giải thích bởi mô hình dự báo so với tổng phương sai của dữ liệu. R^2 dao động từ 0 đến 1, trong đó giá trị càng gần 1 thì mô hình càng tốt trong việc giải thích biến động của dữ liệu thực tế. $R^2 = 0$ nghĩa là mô hình không giải thích được gì về sự thay đổi của dữ liệu, trong khi $R^2 = 1$ cho thấy mô hình giải thích hoàn toàn các biến động.

Ứng dụng: R^2 thường được sử dụng để đánh giá mức độ phù hợp của mô hình trong dự báo dữ liệu chuỗi thời gian. Nó giúp xác định xem mô hình có khả năng dự báo chính xác các biến động trong dữ liệu hay không, đặc biệt quan trọng trong các ứng dụng tài chính và kinh doanh, nơi sự chính xác trong dự báo có thể ảnh hưởng đến quyết định chiến lược.

Việc sử dụng các chỉ số đánh giá phù hợp không chỉ giúp so sánh và lựa chọn mô hình dự báo tối ưu mà còn cung cấp hiểu biết sâu sắc về hiệu suất của từng mô hình đối với các thách thức khác nhau trong dữ liệu chuỗi thời gian. Điều này đóng vai trò quan trọng trong việc hỗ trợ các quyết định đầu tư chính xác và xây dựng chiến lược kinh doanh bền vững, tạo ra lợi thế cạnh tranh và tối ưu hóa lợi nhuận dài hạn.

5.2 Tổng hợp kết quả từ chương 4 và phân tích so sánh

5.2.1 ARIMA

Bảng 5.1 - Kết quả thực nghiệm với ARIMA thông thường

ARIMA			
	MWG	**ACB**	**CMC**
MAE	11188.04	3306.91	9595.03
MSE	188429769.25	12046450.62	165753828.13
RMSE	13726.97	3470.79	12874.54
MAPE	19.11%	14.03%	16.52%
R2-Score	-1.94	-10.11	-0.40

Bảng 5.2 - Kết quả thực nghiệm với ARIMA kết hợp kỹ thuật Sliding Window

ARIMA_SLIDING			
	MWG	**ACB**	**CMC**
MAE	778.48	258.96	1158.29
MSE	1040860.24	121403.44	2694080.22
RMSE	1020.22	348.42	1641.36
MAPE	1.46%	1.11%	2.21%
R2-Score	0.98	0.88	0.97

Bảng 5.3 - kết quả thực nghiệm với ARIMA kết hợp kỹ thuật Expanding Window

ARIMA_EXPANDING			
	MWG	**ACB**	**CMC**
MAE	775.77	218.99	1137.14
MSE	1042895.45	85794.03	2463070.39
RMSE	1021.22	292.91	1569.41
MAPE	1.46%	0.94%	2.15%

R2-Score	0.98	0.92	0.97

5.2.2 SARIMA

Bảng 5.4 - Kết quả thực nghiệm với SARIMA thông thường

SARIMA			
	MWG	**ACB**	**CMC**
MAE	11224.51	3238.67	9371.58
MSE	189337657.33	11580712.04	157364284.31
RMSE	13760	3403	12544.49
MAPE	19.17%	13.73%	16.18
R2-Score	-1.94	-10.11	-0.40

Bảng 5.5 - Kết quả thực nghiệm với SARIMA kết hợp Sliding Window

SARIMA			
	MWG	**ACB**	**CMC**
MAE	839.62	234.45	1149.73
MSE	1206238.98	99283.9	2670946.31
RMSE	13760	3403	1634.3
MAPE	1.58%	1.01%	2.18%
R2-Score	-1.04	-8.11	-0.50

5.2.3 Recurrent Neural Networks (RNN)

Bảng 5.6 - Kết quả thực nghiệm với RNN đơn giản nhất

SimpleRNN dự báo trên tập Test				
	ACB	**MWG**	**ACB**	**CMC**
MAE	19.71	21.17	19.71	138.82
MSE	432.77	1117.66	432.77	27116.71

RMSE	20.80	33.43	20.80	164.67
MAPE	0.03%	0.04 %	0.03%	0.25%
R2-Score	99.99%	99.98%	99.99%	99.97%

Bảng 5.7 - Kết quả thực nghiệm với RNN với kỹ thuật Sliding Window

SimpleRNN với kỹ thuật Sliding Window dự báo trên tập Test			
	MWG	**ACB**	**CMC**
MAE	823.07	690.16	138.82
MSE	1141131.53	863453.95	3134674.08
RMSE	1068.23	929.22	1770.50
MAPE	1.73 %	1.05%	2.32%
R2-Score	97.92%	88.14%	97.41%

5.2.4 LSTM

Bảng 5.8 - Kết quả thực nghiệm với mô hình LSTM

	ACB	**CMC**	**MWG**
MAE	1424998.521	744.92	449529.8
MSE	2159149395378	1348498	240906408870
RMSE	1469404	1161	490822.18
MAPE	3.629%	2.267%	3.241%
R2-Score	44.703%	99.455%	95.737%

Bảng 5.9 - Kết quả thực nghiệm với mô hình LSTM kết hợp Sliding Window

	ACB	**CMC**	**MWG**
MAE	554292	3917	657162
MSE	502925104473	31506972	690226306976
RMSE	709172	5613	830798
MAPE	1.735%	10.865%	3.371%

R2-Score	81.288%	87.275%	93.753%

5.2.5 GRU

Bảng 5.10 - Kết quả thực nghiệm trên tập test với mô hình GRU

	ACB	**CMC**	**MWG**
MAE	1435.9927	1544.6443	1436.2068
MSE	2090087.0107	2646313.7551	2096030.9163
RMSE	1445.7133	1626.7494	1447.76756
MAPE	2.1989	2.9881	3.0452
R2-Score	0.7888	0.9776	0.9631

Bảng 5.11 - Kết quả thực nghiệm trên tập test với mô hình GRU kết hợp Sliding Window

	ACB	**CMC**	**MWG**
MAE	1205.4378	1895.6009	1695.4367
MSE	2324392.2016	6618835.6238	4238675.7346
RMSE	1524.5957	2572.7097	2058.80444
MAPE	1.8254	3.5491	3.4982
R2-Score	0.6809	0.9454	0.9229

5.3 Đánh giá

Trong quá trình thực nghiệm, nghiên cứu và so sánh các mô hình dự báo chuỗi thời gian trên dữ liệu giá cổ phiếu của ba công ty MWG, ACB, và CMC, các mô hình ARIMA, SARIMA, SimpleRNN, LSTM, và GRU đã được áp dụng với các kỹ thuật khác nhau như Sliding Window và Expanding Window. Kết quả cho thấy có sự khác biệt đáng kể giữa hiệu suất của các mô hình, từ đó xác định được mô hình phù hợp nhất cho từng mã cổ phiếu cụ thể.

Trước tiên, các mô hình ARIMA và SARIMA, mặc dù phổ biến trong việc dự báo chuỗi thời gian, đã không đạt được kết quả tốt trong nghiên cứu này. Đối với cả ba mã cổ phiếu MWG, ACB, và CMC, các mô hình này cho thấy giá trị lỗi (MAE, MSE, RMSE, MAPE) cao và R2-Score rất thấp, thậm chí âm đối với MWG và ACB. Điều này chỉ ra rằng các mô hình này không thể nắm bắt tốt các đặc điểm của dữ liệu giá cổ phiếu. Cụ thể, với ARIMA thông thường, mô hình này có R2-Score -1.94 đối với MWG, cho thấy khả năng dự báo kém hơn nhiều so

với việc sử dụng giá trị trung bình. Kết quả tương tự cũng xảy ra với ACB (R2-Score -10.11) và CMC (R2-Score -0.40). Mô hình SARIMA cũng không mang lại kết quả khả quan hơn, với các chỉ số lỗi tương tự và R2-Score âm trên cả ba mã cổ phiếu. Khi áp dụng kỹ thuật Sliding Window và Expanding Window, mặc dù có cải thiện nhẹ về R2-Score, nhưng các chỉ số lỗi vẫn cao và hiệu suất dự báo vẫn không đạt yêu cầu. Điều này cho thấy sự hạn chế của ARIMA và SARIMA trong việc xử lý dữ liệu tài chính có biến động và tính phi tuyến cao như giá cổ phiếu.

Ngược lại, SimpleRNN đã cho thấy hiệu suất vượt trội so với ARIMA và SARIMA, đặc biệt là trên mã MWG và ACB. Với SimpleRNN thông thường, mô hình này đã đạt được R2-Score 99.98% trên mã MWG, với MAE chỉ 21.17, cho thấy khả năng dự báo rất tốt giá cổ phiếu. Trên ACB và CMC, R2-Score lần lượt là 99.99% và 99.97%, cũng cho thấy sự ổn định và hiệu quả của mô hình này. Tuy nhiên, khi sử dụng kỹ thuật Sliding Window, hiệu suất của SimpleRNN giảm đáng kể, đặc biệt là trên MWG với R2-Score giảm xuống còn 97.92% và các chỉ số lỗi như MAE và RMSE tăng mạnh. Điều này cho thấy mô hình có xu hướng overfitting khi áp dụng kỹ thuật Sliding Window và không thể tổng quát hóa tốt cho các tập dữ liệu kiểm tra.

LSTM, một mô hình học sâu nổi bật với khả năng xử lý các chuỗi thời gian dài hạn và có tính phi tuyến, lại cho kết quả không đồng đều giữa các mã cổ phiếu. Trên mã CMC, LSTM cho kết quả rất tốt với R2-Score 99.455% và chỉ số lỗi thấp, cho thấy khả năng dự báo chính xác và tổng quát hóa tốt. Tuy nhiên, trên ACB, LSTM chỉ đạt R2-Score 44.703%, cho thấy mô hình này không phù hợp với dữ liệu này. Khi kết hợp với Sliding Window, kết quả trên ACB được cải thiện với R2-Score tăng lên 81.288%, nhưng mô hình vẫn gặp khó khăn trong việc dự báo trên MWG, với các chỉ số lỗi vẫn cao.

Cuối cùng, GRU, một biến thể đơn giản hơn của LSTM, đã thể hiện sự cân bằng tốt trong dự báo giá cổ phiếu. Trên mã CMC, GRU đạt R2-Score 0.9776, gần như tương đương với LSTM, nhưng lại có sự ổn định hơn trên các mã khác như ACB và MWG. Tuy nhiên, khi áp dụng Sliding Window, hiệu suất của GRU có phần giảm sút, đặc biệt trên MWG với R2-Score giảm xuống 0.9229 và các chỉ số lỗi tăng lên so với mô hình GRU thông thường.

Dựa trên những kết quả thu được, có thể kết luận rằng SimpleRNN thông thường là lựa chọn tốt nhất cho mã MWG, nhờ khả năng dự báo chính xác và ít sai số nhất trong tất cả các mô hình thử nghiệm. Đối với mã ACB, SimpleRNN thông thường cũng đã cho hiệu quả tốt nhất, giúp mô hình hóa tốt hơn các biến động phức tạp trong dữ liệu của ACB. Trong khi đó, LSTM thông thường vượt trội trong việc dự báo giá cổ phiếu CMC, đặc biệt với khả năng xử lý các chuỗi thời gian dài hạn và phức tạp.

Mặc dù ARIMA và SARIMA là các mô hình mạnh mẽ trong nhiều tình huống, nhưng chúng không thể so sánh với các mô hình học sâu như SimpleRNN, LSTM, và GRU trong việc dự báo giá cổ phiếu có tính biến động cao. Các mô hình học sâu không chỉ có khả năng xử lý tốt các dữ liệu phức tạp và phi tuyến mà còn cho phép ứng dụng các kỹ thuật như Sliding Window

để cải thiện hiệu suất dự báo. Tùy thuộc vào mã cổ phiếu cụ thể, việc lựa chọn mô hình phù hợp là yếu tố quan trọng quyết định sự thành công của dự báo.

ĐỀ XUẤT TRONG TƯƠNG LAI

Trong nghiên cứu này, các mô hình dự báo chuỗi thời gian như ARIMA, SARIMA, SimpleRNN, LSTM và GRU đã được thử nghiệm trên dữ liệu giá cổ phiếu của MWG, ACB, và CMC. Mặc dù đã đạt được những kết quả đáng khích lệ, đặc biệt với các mô hình học sâu như SimpleRNN, LSTM và GRU, vẫn còn nhiều hướng đi mới mà nghiên cứu trong tương lai có thể khám phá để cải thiện độ chính xác và độ tin cậy của các dự báo.

Trước hết, nghiên cứu trong tương lai có thể mở rộng bằng cách thử nghiệm các mô hình học sâu tiên tiến hơn như Transformer hoặc các mô hình dựa trên Attention. Các mô hình này đã cho thấy tiềm năng lớn trong việc nắm bắt các mối quan hệ phức tạp và dài hạn trong dữ liệu chuỗi thời gian, từ đó có thể mang lại kết quả dự báo tốt hơn. Bên cạnh đó, một hướng nghiên cứu quan trọng khác là kết hợp các mô hình (ensemble models) để tận dụng điểm mạnh của từng mô hình riêng lẻ. Việc kết hợp mô hình học sâu với các mô hình truyền thống như ARIMA hay SARIMA trong một mô hình ensemble có thể giúp cải thiện hiệu suất và giảm thiểu các điểm yếu của từng mô hình đơn lẻ.

Ngoài ra, xử lý trước dữ liệu là một khía cạnh quan trọng trong việc nâng cao hiệu quả của các mô hình dự báo. Nghiên cứu trong tương lai có thể tập trung vào việc khám phá các kỹ thuật xử lý dữ liệu tiên tiến hơn như kỹ thuật chọn lọc đặc trưng (feature engineering), giảm chiều dữ liệu (dimensionality reduction) hoặc phát hiện dị thường (anomaly detection) để loại bỏ những yếu tố gây nhiễu và tăng độ chính xác của mô hình. Bên cạnh đó, hiệu suất của các mô hình học sâu phụ thuộc nhiều vào các tham số như kích thước mạng, số lượng lớp và tốc độ học. Tối ưu hóa siêu tham số (hyperparameter optimization) bằng các thuật toán như Bayesian Optimization hoặc Genetic Algorithms có thể là một bước quan trọng để đạt được kết quả tốt hơn.

Hơn nữa, nghiên cứu hiện tại chỉ giới hạn trong ba mã cổ phiếu MWG, ACB, và CMC. Do đó, nghiên cứu trong tương lai có thể mở rộng bằng cách thử nghiệm trên nhiều mã cổ phiếu khác nhau hoặc trên các thị trường tài chính khác để đánh giá tính tổng quát của các mô hình đã phát triển. Đồng thời, một yếu tố quan trọng khác cần được xem xét trong tương lai là tích hợp các yếu tố kinh tế vĩ mô, chính sách tài chính, và các biến số khác có thể ảnh hưởng đến giá cổ phiếu. Sử dụng mô hình đa biến (multivariate models) kết hợp với dữ liệu kinh tế vĩ mô có thể giúp mô hình dự báo nắm bắt tốt hơn các yếu tố tác động đến thị trường.

Cuối cùng, việc triển khai các mô hình đã phát triển vào các hệ thống giao dịch thực tế để đánh giá hiệu suất trong môi trường sống động và có yếu tố bất ngờ là một bước tiến quan trọng. Kết quả từ việc này có thể cung cấp thông tin giá trị cho việc cải thiện mô hình và điều chỉnh các chiến lược giao dịch. Nhìn chung, những hướng phát triển trong tương lai này sẽ không chỉ giúp tăng cường hiệu quả của các mô hình dự báo mà còn mở rộng khả năng áp dụng chúng trong thực tiễn.

REFERENCES

Ananda Chatterjee, et al. “Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models.” 2021.

I, Vaia. “A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks.” *MDPI*, https://www.mdpi.com/1999-5903/15/8/255.

Siami, Sima, and Akbar Siami. “[1803.06386] Forecasting Economics and Financial Time Series: ARIMA vs. LSTM.” *arXiv*, 16 March 2018, https://arxiv.org/abs/1803.06386.

Sima Siami-Namini, et al. “A Comparison of ARIMA and LSTM in Forecasting Time Series.” *IEEE International Conference on Machine Learning and Applications*, 2018.