

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KINH TẾ - TÀI CHÍNH THÀNH PHỐ HỒ CHÍ MINH



ĐỒ ÁN CHUYÊN NGÀNH

Ngành: Công nghệ thông tin

Chuyên ngành: Trí tuệ nhân tạo

TÊN ĐỀ TÀI

**DỰ BÁO GIÁ VÀNG THEO GIÁ ĐÔ LA MỸ (XAU/USD) SỬ DỤNG MÔ
HÌNH ARIMA VÀ LSTM**

Giảng viên hướng dẫn : TS. Nguyễn Quốc Minh

Sinh viên thực hiện :

Nguyễn Cao Bằng

210051639

21D1TH-NT01

TP. Hồ Chí Minh, thứ sáu, ngày 6 tháng 1, năm 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KINH TẾ - TÀI CHÍNH THÀNH PHỐ HỒ CHÍ MINH

ĐỒ ÁN CHUYÊN NGÀNH

Ngành: Công nghệ thông tin

Chuyên ngành: Trí tuệ nhân tạo

TÊN ĐỀ TÀI

**DỰ BÁO GIÁ VÀNG THEO GIÁ ĐÔ LA MỸ (XAU/USD) SỬ DỤNG MÔ
HÌNH ARIMA VÀ LSTM**

Giảng viên hướng dẫn : TS. Nguyễn Quốc Minh

Sinh viên thực hiện :

Nguyễn Cao Bằng

210051639

21D1TH-NT01

TP. Hồ Chí Minh, thứ sáu, ngày 6 tháng 1, năm 2025

MỤC LỤC

MỤC LỤC.....	1
DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT.....	3
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ	4
CHƯƠNG 1: TỔNG QUAN.....	7
1.1. Tóm tắt nghiên cứu	7
1.2. Đồ án	7
1.2.1. Nhiệm vụ đồ án	7
1.2.2. Cấu trúc đồ án	9
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	10
2.1. Giới thiệu về chuỗi thời gian.....	10
2.2. Mô hình ARIMA.....	13
2.2.1. Các bước để xây dựng một ARIMA model	17
2.3. Mạng LSTM.....	18
2.3.1. Ý tưởng cốt lõi của LSTM	19
2.3.2. Bên trong LSTM	21
2.3.3. Các biến thể của bộ nhớ dài hạn	23
2.3.4. Kiến trúc LSTM	24
2.3.5. Ứng dụng của LSTM	26
2.4. Lý thuyết về Vàng và đô la Mỹ.....	27
2.4.1. Khái niệm vàng và Đô la Mỹ	27
2.4.2. Vai trò của vàng và Đô la Mỹ trong nền kinh tế hiện nay	27
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ	29
3.1. Lý thuyết về ngôn ngữ lập trình.....	29
3.1.1. Ngôn ngữ R	29
3.1.2. Python	29
3.2. Giới thiệu về dữ liệu.....	30

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM.....	31
4.1. Dự đoán bằng mô hình ARIMA.....	31
4.1.1. Bước 1: Phân tích, kiểm tra tính mùa vụ của dữ liệu và khử mùa.....	31
4.1.2. Bước 2: Sai phân, kiểm tra tính dừng, mô hình ACF – PACF và chọn các giá trị P, D, Q (dữ liệu chưa khử mùa).....	36
4.1.3. Bước 3: Sai phân, kiểm tra tính dừng, mô hình ACF – PACF và chọn các giá trị p, d, q (dữ liệu đã khử mùa).....	40
4.1.4. Bước 4: Lựa chọn mô hình ARIMA	44
4.1.5. Bước 5: Kiểm định mô hình đã chọn	54
4.2. Dự đoán bằng mô hình LSTM	60
4.2.1. Bước 1: Tiền xử lý dữ liệu	60
4.2.2. Bước 2: Xây dựng mô hình.....	65
4.2.3. Bước 3: Tái sử dụng mô hình.....	68
4.2.4. Bước 4: Kiểm tra độ chính xác của mô hình	69
4.2.5. Dự đoán giá tương lai.....	73
4.3. So sánh dự đoán của 2 mô hình ARIMA và LSTM.....	76
CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ	80
5.1. Kết luận	80
5.2. Kiến nghị.....	80
5.2.1. Các nhà đầu tư ngắn hạn (khoảng 1-6 tháng)	80
5.2.2. Nhà đầu tư dài hạn (6-12 tháng)	81
5.2.3. Nhà đầu tư trung tính (kết hợp cả hai mô hình).....	81
TÀI LIỆU THAM KHẢO.....	82

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

STT	Ký hiệu/Viết tắt	Viết đầy đủ	Nội dung/Ý nghĩa
1	ARIMA	Autoregressive Integrated Moving Average	Trung bình động tích hợp tự hồi quy
2	XAU/USD	Gold/Dollar	Tỷ giá của vàng so với USD
3	AR	AutoRegressive	Tự động hồi quy
4	MA	Moving Average	Đường trung bình động
5	ARMA	AutoRegressive Moving Average	Đường trung bình động tự hồi quy
6	GARCH	Generalized Autoregressive Conditional Heteroskedasticity	Mô hình tổng quát tự điều chỉnh phương sai có điều kiện khác nhau
7	ACF	AutoCorrelation Function	Hàm tự tương quan
8	PACF	Partial AutoCorrelation Function	Hàm tự tương quan từng phần
9	AIC	Akaike's Information Criterion	Tiêu chí thông tin của Akaike
10	BIC	Bayesian Information Criterion	Tiêu chí thông tin Bayesian
11	MSE	Mean Squared Error	Sai số trung bình bình phương

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 2-1: Đồ thị về chuỗi nhiệt độ trung bình theo tháng thể hiện yếu tố mùa vụ.	10
Hình 2-2: Đồ thị về yếu tố xu hướng trong chuỗi thời gian của chuỗi giá.	11
Hình 2-3: Mô-đun lặp lại trong RNN tiêu chuẩn chứa một lớp duy nhất.	19
Hình 2-4: Mô-đun lặp lại trong LSTM chứa bốn lớp tương tác.	19
Hình 2-5: Thanh trạng thái.	20
Hình 2-6: Hàm sigmoid.	20
Hình 4-1: Dữ liệu giá XAU/USD dạng time series.	31
Hình 4-2: Xu hướng giá XAU/USD.	32
Hình 4-3: Giá trị thành phần mùa vụ của giá XAU/USD.	33
Hình 4-4: Đồ thị đồ thị xu hướng ước tính của giá XAU/USD.	34
Hình 4-5: xu hướng dữ liệu giá XAU/USD đã loại bỏ mùa vụ.	35
Hình 4-6: giá trị dữ liệu XAU/USD sau khi loại bỏ mùa vụ.	35
Hình 4-7: Biểu đồ dữ liệu chưa khử mùa sau khi sai phân lần 1 của giá XAU/USD.	37
Hình 4-8: Giá trị tương quan từng phần ở 60 độ trễ của giá XAU/USD.	38
Hình 4-9: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD.	38
Hình 4-10: Giá trị tương quan ở 60 độ trễ của giá XAU/USD.	39
Hình 4-11: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD.	39
Hình 4-12: Biểu đồ dữ liệu đã khử mùa sau khi sai phân lần 1 của giá XAU/USD.	41
Hình 4-13: Giá trị tương quan từng phần ở 60 độ trễ của giá XAU/USD (đã khử mùa)	42
Hình 4-14: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD (đã khử mùa)	42
Hình 4-15: Giá trị tương quan ở 60 độ trễ của giá XAU/USD (đã khử mùa).	43

Hình 4-16: Mô hình tương quan ở 60 độ trễ của giá XAU/USD (đã khử mùa)	43
Hình 4-17: Giá trị tương quan từng phần ở 60 độ trễ của giá XAU/USD	46
Hình 4-18: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD.....	47
Hình 4-19: Giá trị tương quan ở 60 độ trễ của giá XAU/USD	48
Hình 4-20: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD.....	48
Hình 4-21: Giá trị tương quan từng phần ở 60 độ trễ của giá XAU/USD (đã khử mùa)	49
Hình 4-22: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD (đã khử mùa)	50
Hình 4-23: Giá trị tương quan ở 60 độ trễ của giá XAU/USD (đã khử mùa)	50
Hình 4-24: Mô hình tương quan ở 60 độ trễ của giá XAU/USD (đã khử mùa)	51
Hình 4-25: Kết quả khởi chạy autoarima cho dữ liệu chưa khử mùa	52
Hình 4-26: Kết quả khởi chạy autoarima cho dữ liệu đã khử mùa	52
Hình 4-27: Giá trị AIC cho hai mô hình arima đã chọn	54
Hình 4-28: Giá trị BIC cho hai mô hình arima đã chọn.....	54
Hình 4-29: Mô hình các nghiệm nghịch đảo của giá XAU/USD	55
Hình 4-30: Kết quả kiểm định Ljung-Bổ cho mô hình modelA	56
Hình 4-31: Mô hình đồ thị phần dư cho mô hình modelA	57
Hình 4-32: Kết quả dự đoán của mô hình modelA	58
Hình 4-33: Đồ thị dự đoán của mô hình modelA	58
Hình 4-34: Biểu đồ phần dự đoán của modelA.....	59
Hình 4-35: Giá trị chỉ số vàng thực tế ba tháng cuối năm 2022	59
Hình 4-36: Giá XAU/USD sau khi đọc file	60
Hình 4-37: Dữ liệu sau khi đã được định dạng lại cột ngày	61

Hình 4-38; Thông tin của dữ liệu XAU/USD	61
Hình 4-39: Biểu đồ của dữ liệu XAU/USD sau khi đã xử lý.....	62
Hình 4-40: Hai cột giá trị chính dùng để huấn luyện mô hình	63
Hình 4-41: Biểu đồ so sánh giá dự báo và giá thực tế của XAU/USD	70
Hình 4-42: Dữ liệu giá dự báo huấn luyện.....	71
Hình 4-43: Dữ liệu giá dự báo kiểm tra	72
Hình 4-44: Biểu đồ giá trị phần dư của tập huấn luyện và tập kiểm tra	73
Hình 4-45: Dữ liệu dự đoán giá XAU/USD trong 12 tháng	74
Hình 4-46: Biểu đồ dự đoán giá XAU/USD trong 12 tháng.....	75
Hình 4-47: Biểu đồ dự đoán giá vàng 12 tháng tới.....	75
Hình 4-48: Biểu đồ xu hướng dự đoán giá XAU/USD bằng mô hình ARIMA	77
Hình 4-49: Biểu đồ xu hướng dự đoán giá XAU/USD bằng mô hình LSTM	77
Hình 4-50: Biểu đồ dự đoán giá XAU/USD bằng mô hình ARIMA.....	78
Hình 4-51; Biểu đồ dự đoán giá XAU/USD bằng mô hình LSTM	78

CHƯƠNG 1: TỔNG QUAN

1.1. Tóm tắt nghiên cứu

Giá vàng (XAU/USD) là một trong những chỉ số quan trọng trong thị trường tài chính toàn cầu, phản ánh giá trị của vàng so với đồng đô la Mỹ. Vàng không chỉ được xem xét như một tài sản đầu tư mà còn là một công cụ phòng ngừa rủi ro trong bối cảnh biến động kinh tế và chính trị. Có rất nhiều yếu tố khách quan ảnh hưởng đến giá vàng như: tỷ lệ lạm phát, lãi suất, biến động thị trường chứng khoán và các sự kiện chính trị.

Nghiên cứu về giá vàng thường tập trung vào việc phân tích các xu hướng lịch sử, các yếu tố tác động và dự đoán giá trị trong tương lai. Việc hiểu rõ xu hướng của giá vàng không chỉ giúp các nhà đầu tư đưa ra quyết định thông minh mà còn hỗ trợ các nhà hoạch định chính sách trong việc điều chỉnh chiến lược tài chính.

Với sự phát triển của công nghệ và các công cụ phân tích dữ liệu, việc dự đoán giá vàng ngày càng trở nên chính xác hơn, mở ra cơ hội mới cho các nhà đầu tư và các tổ chức tài chính.

Đó là lý do mà chúng tôi đã chọn “**Sử dụng phân tích chuỗi thời gian cho việc dự báo giá XAU/USD (giá vàng theo giá đô la Mỹ)**” làm đề tài cho đề án này với mong muốn có thể dự báo được chỉ số của giá vàng theo giá đô la Mỹ trong khoảng thời gian từ vài tháng từ các số liệu chi tiết đã có trong hơn 50 năm qua (1970 – 2024) để một phần giúp cho các nhà đầu tư có thể tham khảo và đưa ra được những lựa chọn đúng đắn để đầu tư cho vàng theo giá đồng đô la Mỹ.

1.2. Đề án

1.2.1. Nhiệm vụ đề án

1. Tính cấp thiết và lý do hình thành đề tài

Tính cấp thiết: Dự đoán giá vàng/đô la Mỹ (XAU/USD) là một đề tài quan trọng và cấp thiết trong lĩnh vực tài chính và đầu tư. Các nhà đầu tư, các ngân hàng, và các tổ chức tài chính sẽ rất quan tâm đến việc dự đoán trước những sự biến đổi của giá vàng để

có thể giúp họ đưa ra quyết định đầu tư thông minh và hiệu quả nhất. Mặc dù chắc chắn sẽ có những sai số tương đối nhưng nó cũng sẽ giúp được cho các nhà đầu tư giảm thiểu được các rủi ro không mong muốn.

Lý do hình thành đề tài: giá vàng/đô la Mỹ (XAU/USD) có những tác động và ảnh hưởng lớn đến nền kinh tế toàn cầu và trong thị trường tài chính. Việc có thể dự đoán chính xác xu hướng và biến động của của chỉ số này có tác dụng lớn trong việc giúp các nhà đầu tư, doanh nghiệp và chính phủ đưa ra những quyết định chiến lược và phòng ngừa các rủi ro lớn liên quan đến kinh tế và tài chính.

2. Ý nghĩa khoa học và thực tiễn

Ý nghĩa khoa học: Nghiên cứu và dự đoán giá vàng/đô la Mỹ (XAU/USD) bằng cách sử dụng hai mô hình ARIMA và LSTM có thể mang lại những kiến thức mới về sự biến đổi và tương tác giữa các chỉ số này với các yếu tố kinh tế, chính trị và xã hội. Nó cung cấp cho chúng ta cái nhìn sâu sắc hơn về các yếu tố đã đang và sẽ ảnh hưởng đến giá vàng.

Ý nghĩa thực tiễn: Áp dụng mô hình ARIMA và LSTM để dự đoán giá vàng mang lại lợi ích thực tiễn cho nhiều nhóm đối tượng khác nhau, bao gồm:

- + Nhà đầu tư: Dự đoán chính xác giúp cho các nhà đầu tư nói chung và bản thân nói riêng đưa ra quyết định mua bán và định hình chiến lược đầu tư một cách cẩn trọng nhất trên các sàn chứng khoán cũng như tích trữ tài sản.

- + Ngân hàng - tổ chức tài chính: Dự đoán này sẽ giúp họ quản lý rủi ro và đưa ra quyết định về việc bảo vệ tài sản và quản lý vốn.

- + Doanh nghiệp: Dự đoán giúp họ định hình chiến lược về nhập khẩu, xuất khẩu và quản lý giá thành.

- + Chính phủ: Dự đoán giúp chính phủ đưa ra các biện pháp kinh tế và chính sách tiền tệ hợp lý để ổn định thị trường và tăng trưởng kinh tế.

3. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng hai mô hình ARIMA và LSTM, sau đó so sánh, đánh giá, nhận xét cả hai mô hình, cuối cùng kết luận mô hình nào có khả năng dự đoán giá vàng/đô la Mỹ một cách chính xác và tin cậy hơn trong khoảng thời gian ngắn. Mô hình sẽ được sử dụng để dự đoán các xu hướng và những sự biến động trong tương lai, đồng thời cung cấp thông tin hữu ích cho các nhà đầu tư, các ngân hàng, tổ chức tài chính, doanh nghiệp và chính phủ để họ có thể đưa ra quyết định thông minh và hiệu quả trong lĩnh vực tài chính và đầu tư.

1.2.2. Cấu trúc đề án

CHƯƠNG 1: TỔNG QUAN

Tóm tắt nghiên, trình bày về những nghiên cứu tìm hiểu, nêu những nhiệm vụ của đề án và cấu trúc của đề án ngắn gọn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Trình bày các cụm từ, những khái niệm khoa học về những từ ngữ sẽ nghiên cứu trong đề tài cũng như giới thiệu sơ lược về bộ dữ liệu được sử dụng.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

Giới thiệu về ngôn ngữ lập trình đã sử dụng, mô hình phân tích chuỗi thời gian sử dụng trong đề tài này và các lý thuyết cơ bản về những mô hình trên.

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

Trình bày các bước thực hiện mô hình đã lý giải trong chương 3, khởi chạy chi tiết từng bước rõ ràng, cho thấy kết quả, giải thích và nhận xét chung về các kết quả đó.

CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ

Rút ra các kết luận cuối cùng sau khi chạy mô hình và dự đoán ở chương 4. Sau đó sẽ đưa ra những kiến nghị về kết quả dự báo cũng như sẽ kiến nghị về các chiến lược đầu tư khả thi nhất dựa theo kết quả đã dự đoán được.

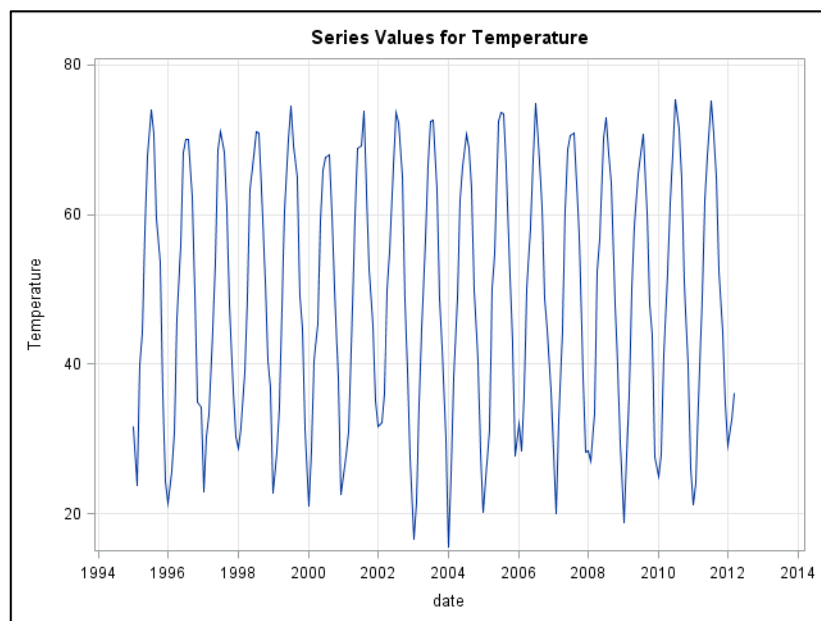
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu về chuỗi thời gian

Dự báo chuỗi thời gian là một lớp mô hình quan trọng trong thống kê, kinh tế lượng và machine learning. Sở dĩ chúng ta gọi lớp mô hình này là chuỗi thời gian (time series) là vì mô hình được áp dụng trên các chuỗi đặc thù có yếu tố thời gian. Một mô hình chuỗi thời gian thường dự báo dựa trên giả định rằng các quy luật trong quá khứ sẽ lặp lại ở tương lai. Do đó xây dựng mô hình chuỗi thời gian là chúng ta đang mô hình hóa mối quan hệ trong quá khứ giữa biến độc lập (biến đầu vào) và biến phụ thuộc (biến mục tiêu). Dựa vào mối quan hệ này để dự đoán giá trị trong tương lai của biến phụ thuộc.

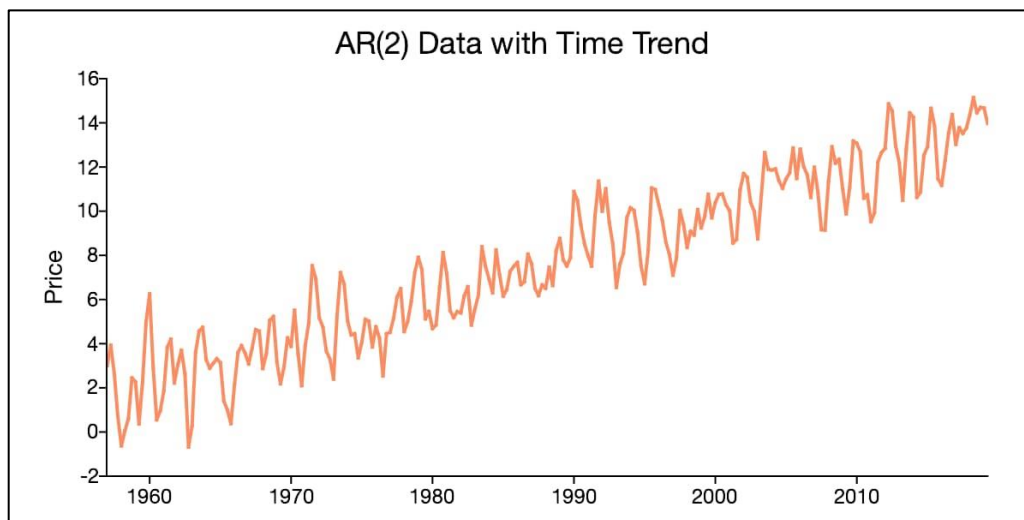
Do là dữ liệu chịu ảnh hưởng bởi tính chất thời gian nên chuỗi thời gian thường xuất hiện những quy luật đặc trưng như: yếu tố chu kỳ, mùa vụ và yếu tố xu hướng. Đây là những đặc trưng thường thấy và xuất hiện ở hầu hết các chuỗi thời gian.

Yếu tố chu kỳ, mùa vụ là những đặc tính lặp lại theo chu kỳ. Ví dụ như nhiệt độ trung bình các tháng trong năm sẽ chịu ảnh hưởng bởi các mùa xuân, hạ, thu, đông. Hay xuất nhập khẩu của một quốc gia thường có chu kỳ theo các quý.



Hình 2-1: Đồ thị về chuỗi nhiệt độ trung bình theo tháng thể hiện yếu tố mùa vụ.

Yếu tố xu hướng (trend) thể hiện đà tăng hoặc giảm của chuỗi trong tương lai. Chẳng hạn như lạm phát là xu hướng chung của các nền kinh tế, do đó giá cả trung bình của giỏ hàng hóa cơ sở hay còn gọi là chỉ số CPI luôn có xu hướng tăng và xu hướng tăng này đại diện cho sự mất giá của đồng tiền.



Hình 2-2: Đồ thị về yếu tố xu hướng trong chuỗi thời gian của chuỗi giá.

Các dự báo chuỗi thời gian có tính ứng dụng cao và được sử dụng rất nhiều lĩnh vực như tài chính ngân hàng, chứng khoán, bảo hiểm, thương mại điện tử, marketing, quản lý chính sách. Bên dưới là một số ứng dụng của dự báo chuỗi thời gian:

Dự báo nhu cầu thị trường để lập kế hoạch sản xuất kinh doanh cho hãng.

Dự báo lợi suất tài sản tài chính, tỷ giá, giá cả hàng hóa phái sinh để thực hiện trading hiệu quả trong market risk.

Dự báo giá chứng khoán, các chuỗi lợi suất danh mục để quản trị danh mục đầu tư.

Dự báo giá bitcoin, giá dầu mỏ, giá gas, ...

Dự báo nhiệt độ, lượng mưa để lập kế hoạch sản xuất nông, lâm, ngư nghiệp.

Dự báo tác động của các nhân tố vĩ mô như lãi suất, cung tiền, đầu tư trực tiếp nước ngoài, chi tiêu chính phủ, lạm phát... tác động lên tăng trưởng GDP để điều hành nền kinh tế.

Vai trò của chuỗi thời gian rất quan trọng đối với nền kinh tế và hoạt động của doanh nghiệp nên trong machine learning và thống kê có những ngành học nghiên cứu chuyên sâu về chuỗi thời gian như kinh tế lượng, định giá tài sản tài chính.

Khác với các mô hình dự báo thông thường trong machine learning, các mô hình trong dự báo chuỗi thời gian trong kinh tế lượng có những đặc trưng rất riêng. Đòi hỏi phải tuân thủ nghiêm ngặt các điều kiện về chuỗi dừng, nhiễu trắng và tự tương quan. Có rất nhiều lớp mô hình chuỗi thời gian khác nhau và mỗi một lớp mô hình sẽ có một tiêu chuẩn áp dụng cụ thể. Chúng ta có thể liệt kê một số mô hình phổ biến:

Mô hình ARIMA: Dựa trên giả thuyết chuỗi dừng và phương sai sai số không đổi. Mô hình sử dụng đầu vào chính là những tín hiệu quá khứ của chuỗi được dự báo để dự báo nó. Các tín hiệu đó bao gồm: chuỗi tự hồi qui AR (auto regression) và chuỗi trung bình trượt MA (moving average). Hầu hết các chuỗi thời gian sẽ có xu hướng tăng hoặc giảm theo thời gian, do đó yếu tố chuỗi dừng thường không đạt được. Trong trường hợp chuỗi không dừng thì ta sẽ cần biến đổi sang chuỗi dừng bằng sai phân. Khi đó tham số đặc trưng của mô hình sẽ có thêm thành phần bậc của sai phân d và mô hình được đặc tả bởi 3 tham số ARIMA (p, d, q).

Mô hình SARIMA: Về bản chất đây là mô hình ARIMA nhưng được điều chỉnh đặc biệt để áp dụng cho những chuỗi thời gian có yếu tố mùa vụ. Như chúng ta đã biết về bản chất ARIMA chính là mô hình hồi qui tuyến tính nhưng mối quan hệ tuyến tính thường không giải thích tốt chuỗi trong trường hợp chuỗi xuất hiện yếu tố mùa vụ. Chính vì thế, bằng cách tìm ra chu kỳ của qui luật mùa vụ và loại bỏ nó khỏi chuỗi ta sẽ dễ dàng hồi qui mô hình theo phương pháp ARIMA.

Mô hình ARIMAX: Là một dạng mở rộng của model ARIMA. Mô hình cũng dựa trên giải định về mối quan hệ tuyến tính giữa giá trị và phương sai trong quá khứ với giá trị hiện tại và sử dụng phương trình hồi qui tuyến tính được suy ra từ mối quan hệ trong quá khứ nhằm dự báo tương lai. Mô hình sẽ có thêm một vài biến độc lập khác và cũng được xem như một mô hình hồi qui động (hoặc một

số tài liệu tiếng việt gọi là mô hình hồi qui động thái). Về bản chất ARIMAX tương ứng với một mô hình hồi qui đa biến nhưng chiếm lợi thế trong dự báo nhờ xem xét đến yếu tố tự tương quan được biểu diễn trong phần dư của mô hình. Nhờ đó cải thiện độ chính xác.

Mô hình GARCH: Các giả thuyết về chuỗi dừng và phương sai sai số không đổi đều không dễ đạt được trong thực tế. Trái lại phương sai sai số biến đổi rất dễ xảy ra đối với các chuỗi tài chính, kinh tế bởi thường có những sự kiện không mong đợi và cú sốc kinh tế không lường trước khiến biến động phương sai của chuỗi thay đổi. Trong trường hợp đó nếu áp dụng ARIMA thì thường không mang lại hiệu quả cao cho mô hình. Các nhà kinh tế lượng và thống kê học đã nghĩ đến một lớp mô hình mà có thể dự báo được phương sai để kiểm soát các thay đổi không mong đợi. Dựa trên qui luật của phương sai, kết quả dự báo chuỗi sẽ tốt hơn so với trước đó.

2.2. Mô hình ARIMA

Mô hình ARIMA chỉ mới được khởi xướng từ vài năm trở lại đây, nhưng với sự phát triển của khoa học công nghệ, thế giới đã và đang nghiên cứu mô hình ARIMA ở mức độ ngày càng sâu bởi sự thành công của nó trong dự báo. Trong một số trường hợp dự báo thu được từ phương pháp này có tính tin cậy cao hơn so với các dự báo thu được từ các phương pháp mô hình kinh tế lượng truyền thống khác, đặc biệt là trong dự báo ngắn hạn. Tại Việt Nam cũng có một số công trình nghiên cứu vận dụng mô hình ARIMA để dự báo sản lượng lúa, lượng khách du lịch, nhưng rất ít nghiên cứu liên quan đến chứng khoán được trình bày một cách hệ thống và cập nhật. Tuy nhiên, nếu ứng dụng tốt thì ARIMA sẽ là công cụ hữu hiệu đối với nhà đầu tư chứng khoán Việt Nam. Dựa trên mô hình tự hồi quy AR và mô hình trung bình động MA. ARIMA là mô hình dự báo định lượng theo thời gian, giá trị tương lai của biến số dự báo sẽ phụ thuộc vào xu thế vận động của đối tượng đó trong quá khứ. George Box và Gwilym Jenkins đã nghiên cứu mô hình ARIMA (Autoregressive Integrated Moving Average - Tự hồi qui tích hợp Trung bình trượt), và tên của họ được dùng để gọi tên các quá trình ARIMA tổng quát,

áp dụng vào việc phân tích và dự báo các chuỗi thời gian. Bản chất số liệu chuỗi thời gian là thể hiện tâm lý của nhà đầu tư và thông qua diễn biến hành vi và hướng đi của dữ liệu. Vì vậy sử dụng mô hình ARIMA để dự báo Vn-Index là rất phù hợp. Thực hiện phương pháp Box-Jenkins gồm bốn bước: nhận dạng mô hình thử nghiệm, ước lượng, kiểm định bằng chuẩn đoán và dự báo. Đầu tiên, mô hình ARIMA phân tích tính tương quan giữa các dữ liệu quan sát để đưa ra mô hình dự báo thông qua các giai đoạn nhận dạng mô hình, ước lượng các tham số từ dữ liệu quan sát và kiểm tra các tham số ước lượng để tìm ra mô hình thích hợp. Mô hình kết quả của quá trình trên gồm các tham số thể hiện mức độ tương quan trên dữ liệu và được chọn để dự báo giá trị tương lai. Giới hạn độ tin cậy của dự báo được tính dựa trên phương sai của sai số dự báo.

ARIMA model là viết tắt của cụm từ Autoregressive Integrated Moving Average. Mô hình ARIMA (p, d, q) là sự kết hợp của 3 thành phần chính là: tự hồi quy tự động (AR), tích phân (I) và trung bình động (MA).

Đây là một phương pháp định lượng để phát hiện hướng di chuyển của quan sát trong tương lai dựa trên dữ liệu trong quá khứ mà không sử dụng các biến ngoại sinh độc lập. ARIMA không giả định bất kỳ mô hình cụ thể nào, nhưng việc xác định mô hình dựa trên phân tích dữ liệu theo trường hợp cụ thể. Do đó, mô hình này linh hoạt và kinh tế hơn các phương pháp khác và đã chứng minh được hiệu quả trong dự báo trong thế giới thực. Nếu một chuỗi thời gian dừng ở sai phân bậc d, ta nói chuỗi liên kết bậc d, ký hiệu Kết hợp với quá trình ARMA ta có được mô hình trình trung bình trượt, đồng liên kết, tự hồi quy với p số hạng tự hồi quy và q số hạng trung bình trượt, và cần lấy sai phân bậc d để chuỗi dừng. Mô hình sẽ biểu diễn phương trình hồi quy tuyến tính đa biến (multiple linear regression) của các biến đầu vào (còn gọi là biến phụ thuộc trong thống kê) là 2 thành phần:

Thành phần 1: AR (Autoregressive):

Tự hồi quy tự động đề cập đến sự phụ thuộc tuyến tính của giá trị hiện tại trong chuỗi thời gian vào các giá trị trước đó. Thành phần này được đánh giá bằng cách sử dụng các giá trị lag (trễ) của chuỗi thời gian. Độ trễ dài hoặc ngắn trong quá trình AR

phụ thuộc vào tham số trễ p . Cụ thể, quá trình AR (p) của chuỗi p được biểu diễn như bên dưới:

$$AR(p): Y_t = \phi_0 + \phi_1 * Y_{t-1} + \phi_2 * Y_{t-2} + \dots + \phi_p * Y_{t-p} + \epsilon_t$$

Trong đó:

Y_t là giá trị của chuỗi thời gian tại thời điểm t .

ϕ_0 là hằng số.

$\phi_1, \phi_2, \dots, \phi_p$ là các hệ số autoregressive (AR).

ϵ_t là thành phần ngẫu nhiên tại thời điểm t .

Các mô hình chuỗi thời gian được dựa vào giả thiết là các chuỗi thời gian nghiêm cứu có tính dừng yếu. Nói ngắn gọn, giá trị trung bình và phương sai của chuỗi thời gian có tính dừng yếu là hằng số và đồng phương sai của nó không đổi theo thời gian. Nhưng ta biết rằng nhiều chuỗi thời gian kinh tế không có tính dừng, tức là chúng kết hợp.

Thành phần 2: Moving average (MA)

Quá trình trung bình trượt được hiểu là quá trình dịch chuyển hoặc thay đổi giá trị trung bình của chuỗi theo thời gian. Do chuỗi của chúng ta được giả định là dừng nên quá trình thay đổi trung bình dường như là một chuỗi nhiễu trắng. Trung bình động đề cập đến một mô hình tuyến tính của sự phụ thuộc giữa giá trị hiện tại trong chuỗi thời gian và các giá trị trước đó của một thành phần ngẫu nhiên. Thành phần này được đánh giá bằng cách sử dụng các giá trị trung bình của chuỗi thời gian. Quá trình moving average sẽ tìm mối liên hệ về mặt tuyến tính giữa các phần tử ngẫu nhiên (stochastic term). Chuỗi này phải là một chuỗi nhiễu trắng thỏa mãn các tính chất:

$$\begin{cases} E(\epsilon_t) &= 0 & (1) \\ \sigma(\epsilon_t) &= \alpha & (2) \\ \rho(\epsilon_t, \epsilon_{t-s}) &= 0, \forall s \leq t & (3) \end{cases}$$

Giải thích phương trình:

Vế (1) có nghĩa rằng kì vọng của chuỗi bằng 0 để đảm bảo chuỗi dừng không có sự thay đổi về trung bình theo thời gian.

Vế (2) là phương sai của chuỗi không đổi. Do kỳ vọng và phương sai không đổi nên chúng ta gọi phân phối của chuỗi nhiễu trắng là phân phối xác định và được kí hiệu là:

$$MA(q) = \mu + \sum_{i=1}^q \theta_i \epsilon_{t-i}$$

Thành phần MA (Moving Average): MA thường được viết dưới dạng:

$$MA(q): X(t) = \mu + \theta_1 * \epsilon_{t-1} + \theta_2 * \epsilon_{t-2} + \dots + \theta_q * \epsilon_{t-q} + \epsilon_t$$

Trong đó:

$X(t)$ là giá trị hiện tại

μ là trung bình của chuỗi

ϵ_t là nhiễu ngẫu nhiên tại thời điểm t

$\theta_1, \theta_2, \dots, \theta_p$ là hệ số moving average (MA)

Thông thường chuỗi sẽ dừng sau quá trình đồng tích hợp $I(0)$ hoặc $I(1)$. Phương trình hồi qui ARIMA (p, d, q) có thể được biểu diễn dưới dạng:

$$Y_t = \phi_0 + \phi_1 * Y_{t-1} + \phi_2 * Y_{t-2} + \dots + \phi_p * Y_{t-p} + \theta_1 * \epsilon_{t-1} + \theta_2 * \epsilon_{t-2} + \dots + \theta_q * \epsilon_{t-q} + \epsilon_t$$

Y_t là giá trị của chuỗi thời gian tại thời điểm t .

ϕ_0 là hằng số.

$\phi_1, \phi_2, \dots, \phi_p$ là các hệ số autoregressive (AR).

$\theta_1, \theta_2, \dots, \theta_q$ là các hệ số moving average (MA).

ϵ_t là thành phần nhiễu ngẫu nhiên tại thời điểm t .

Integrated: Là quá trình đồng tích hợp hoặc lấy sai phân. Khi biến đổi sang chuỗi dừng, các nhân tố ảnh hưởng thời gian được loại bỏ và chuỗi sẽ dễ dự báo hơn. Để tạo thành chuỗi dừng, một phương pháp đơn giản nhất là chúng ta sẽ lấy sai phân. Một số chuỗi tài chính còn quy đổi sang logarit hoặc lợi suất. Tích phân thể hiện quá trình chuyển đổi chuỗi thời gian thành một chuỗi thời gian mới bằng cách lấy hiệu suất giữa các giá trị tại các thời điểm liên tiếp. Thành phần này giúp ổn định và làm phẳng chuỗi thời gian.

Quá trình sai phân bậc d của chuỗi được thực hiện như sau:

- Sai phân bậc 1: $I(1) = \Delta(x_t) = x_t - x_{t-1}$
- Sai phân bậc d : $I(d) = \Delta^d(x_t) = \underbrace{\Delta(\Delta(\dots \Delta(x_t)))}_{d \text{ times}}$

2.2.1. Các bước để xây dựng một ARIMA model

Bước 1: Kiểm tra tính dừng của Time series.

Time series phải đảm bảo tính dừng. Tính dừng ở đây có nghĩa là nó không có trend tăng dần hoặc giảm dần theo thời gian. Chuỗi được dự báo có giá định là trung bình và phương sai không đổi bởi theo tư tưởng của Box-Jenkins điều này sẽ làm cho chuỗi được dự báo dễ hơn.

Để kiểm tra tính dừng chúng ta sẽ sử dụng chỉ số Augmented Dickey-Fuller unit root test (ADF). Nếu p-value của ADF nhỏ hơn 0.05 thì thỏa mãn tiêu chuẩn dừng. Nếu ADF lớn hơn 0.05 thì chuỗi có nghiệm đơn vị tức là chuỗi là không dừng. ADF về cơ bản là kiểm định hệ số hồi quy của trễ bậc 1 khi hồi một biến theo giá trị trễ bậc 1 của nó. Nếu hệ số này nhỏ hơn 1 thì chuỗi dừng và lớn hơn hoặc bằng 1 thì chuỗi không có tính dừng.

Bước 2: Lấy sai phân.

Mục đích của bước này là chuyển từ một chuỗi không dừng về chuỗi dừng. Chuỗi sai phân của biến được dự báo có thể được kiểm tra tính tự tương quan và tính dừng và tính phân phối chuẩn.

Thông thường chúng ta sẽ lấy sai phân bậc 1 hoặc bậc 2. Sau khi xác định giá trị bậc của sai phân thích hợp chúng ta sẽ chuyển sang bước 3.

Bước 3: Xác định bậc p và q

Trong bước này chúng ta phải xác định bậc phù hợp của 2 phương trình AR và MA. Để tìm được bậc phù hợp chúng ta sẽ cần sử dụng đến đồ thị tự tương quan (ACF) và tự tương quan riêng phần (PACF). Về thế nào là tự tương quan và tự tương quan riêng phần sẽ được trình bày ở bài viết khác.

Xác định bậc của AR model.

Đồ thị PACF sẽ được sử dụng để xác định bậc p của quá trình AR. Dựa vào đỉnh của đồ thị PACF với các mức trễ liên tiếp nhau. Nếu độ trễ lớn nhất có đỉnh nằm ngoài khoảng tin cậy 5% thì giá trị bậc của AR sẽ được quyết định bằng chính độ trễ này.

Xác định bậc của MA model.

Đồ thị ACF sẽ được sử dụng để xác định bậc q của quá trình MA. Cách xác định cũng tương tự như xác định AR là căn cứ vào độ trễ lớn nhất nằm ngoài khoảng tin cậy 5%.

Bước 4: Xây dựng model dự báo.

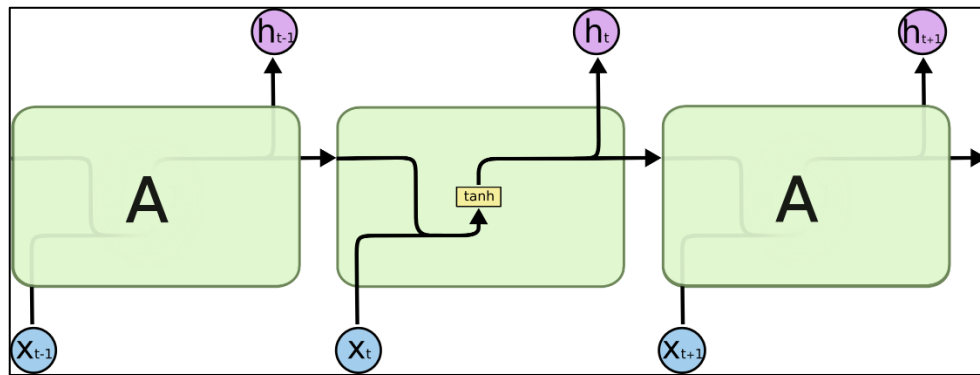
Từ các bước xác định p, d, q ta sẽ có được một quá trình ARIMA(p,d,q) và thực hiện hồi qui trên tập train dataset. Sau khi thu được model dự báo chúng ta sẽ áp dụng trên tập test dataset và cross check xem giá trị dự báo có phù hợp với giá trị thực tế không.

2.3. Mạng LSTM

Mạng bộ nhớ dài-ngắn (Long Short-Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

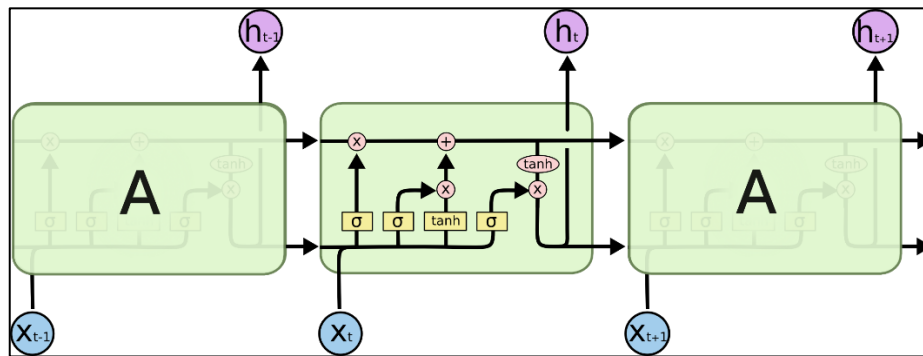
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-đun này có cấu trúc rất đơn giản, thường là một tầng Tanh.



Hình 2-3: Mô-đun lặp lại trong RNN tiêu chuẩn chứa một lớp duy nhất.

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.

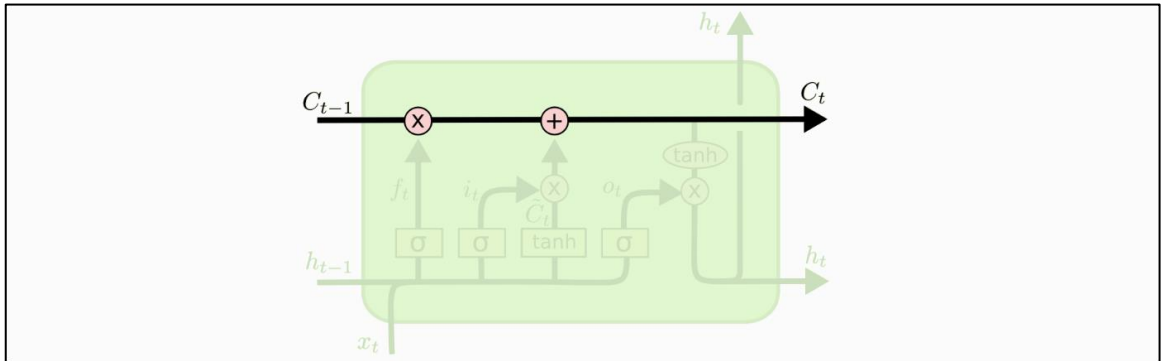


Hình 2-4: Mô-đun lặp lại trong LSTM chứa bốn lớp tương tác.

2.3.1. Ý tưởng cốt lõi của LSTM

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ.

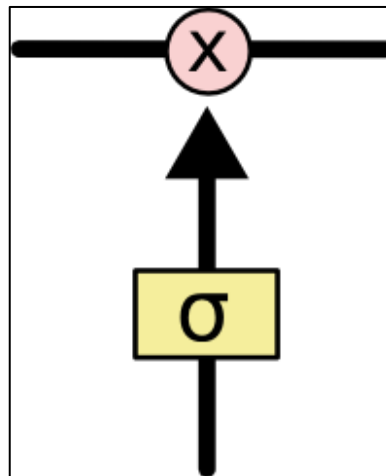
Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.



Hình 2-5: Thanh trạng thái

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế bào, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.



Hình 2-6: Hàm sigmoid

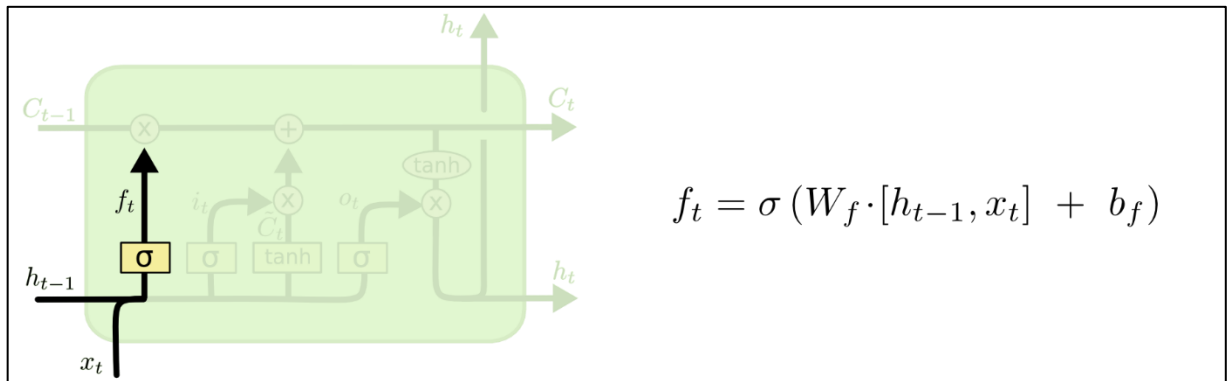
Tầng sigmoid sẽ cho đầu ra là một số trong khoản $[0,1][0,1]$, mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 00 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 11 thì có nghĩa là cho tất cả các thông tin đi qua nó.

Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

2.3.2. Bên trong LSTM

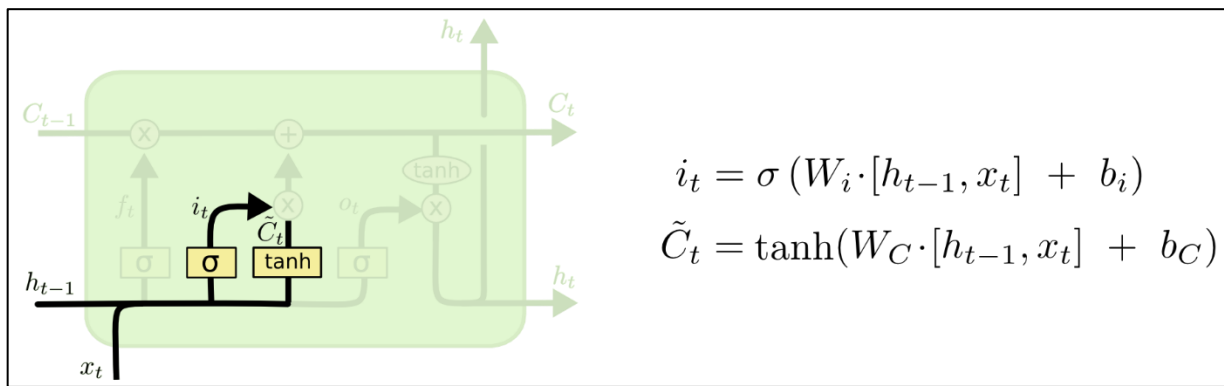
Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là “tầng cổng quên” (forget gate layer). Nó sẽ lấy đầu vào là h_{t-1} và x_t rồi đưa ra kết quả là một số trong khoảng $[0,1]$ cho mỗi số trong trạng thái tế bào C_{t-1}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

Quay trở lại với ví dụ mô hình ngôn ngữ dự đoán từ tiếp theo dựa trên tất cả các từ trước đó, với những bài toán như vậy, thì trạng thái tế bào có thể sẽ mang thông tin về giới tính của một nhân vật nào đó giúp ta sử dụng được đại từ nhân xưng chuẩn xác. Tuy nhiên, khi đề cập tới một người khác thì ta sẽ không muốn nhớ tới giới tính của nhân vật nữa, vì nó không còn tác dụng gì với chủ thể mới này.



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

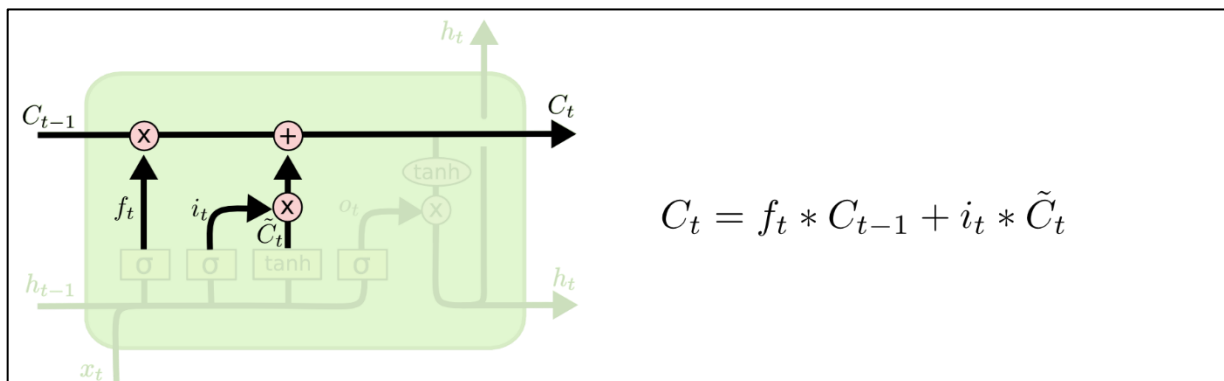
Chẳng hạn với ví dụ mô hình ngôn ngữ của ta, ta sẽ muốn thêm giới tính của nhân vật mới này vào trạng thái tế bào và thay thế giới tính của nhân vật trước đó.



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

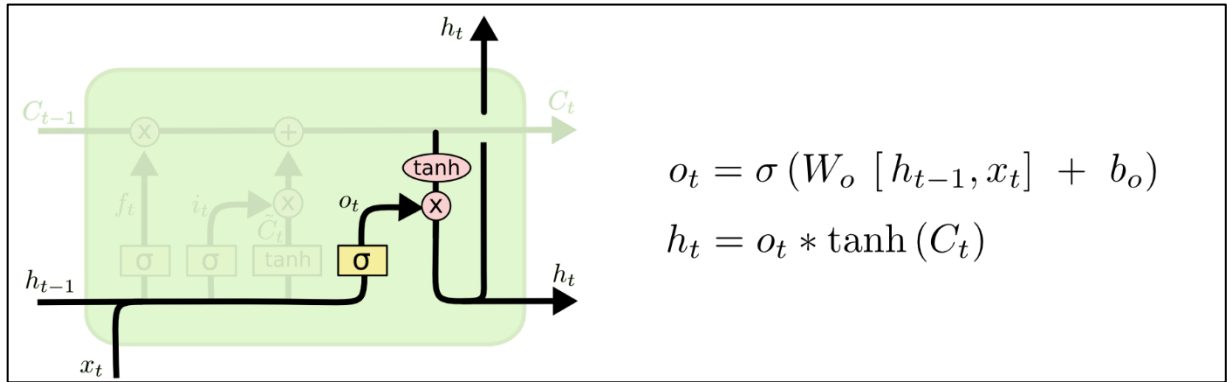
Ta sẽ nhân trạng thái cũ với f_t để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm $i_t * \tilde{C}_t$. Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao.

Với bài toán mô hình ngôn ngữ, chính là việc ta bỏ đi thông tin về giới tính của nhân vật cũ, và thêm thông tin về giới tính của nhân vật mới như ta đã quyết định ở các bước trước đó.



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

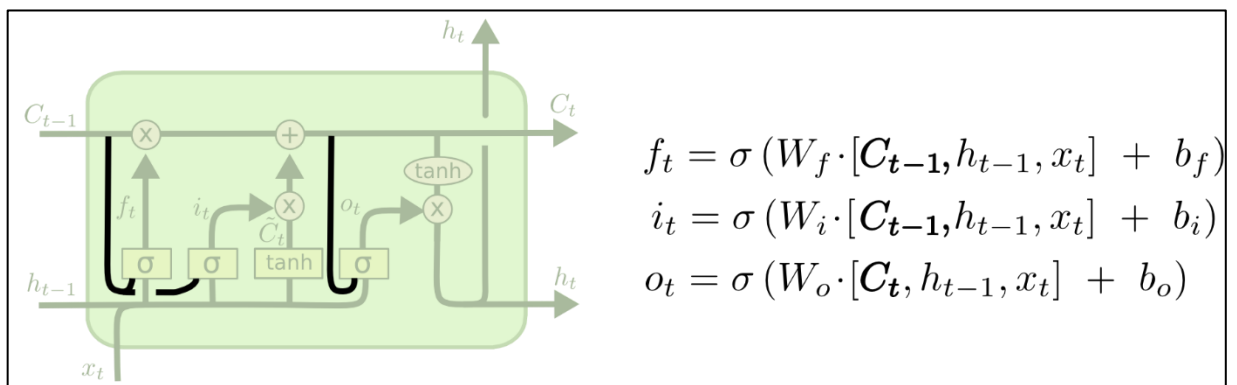
Với ví dụ về mô hình ngôn ngữ, chỉ cần xem chủ thể mà ta có thể đưa ra thông tin về một trạng từ đi sau đó. Ví dụ, nếu đầu ra của chủ thể là số ít hoặc số nhiều thì ta có thể biết được dạng của trạng từ đi theo sau nó phải như thế nào.



2.3.3. Các biến thể của bộ nhớ dài hạn

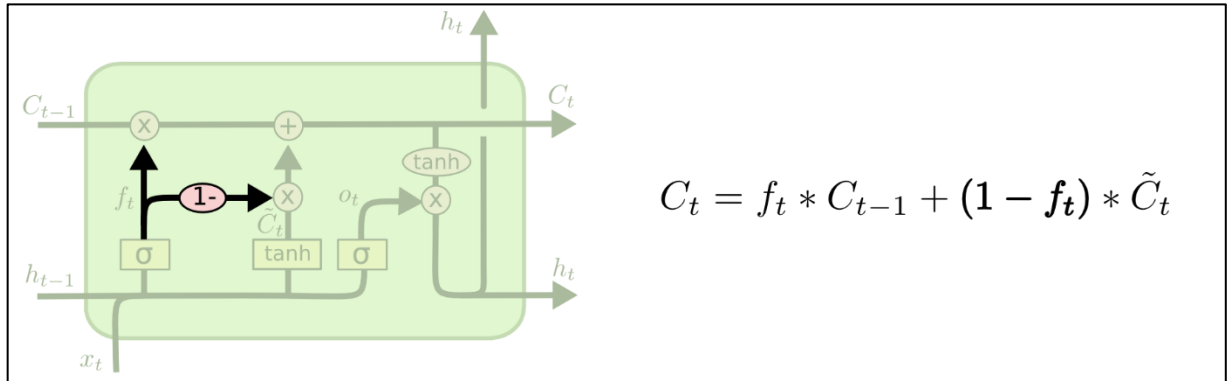
Những thứ ta vừa mô tả ở trên là một LSTM khá bình thường. Nhưng không phải tất cả các LTSM đều giống như vậy. Thực tế, các bài báo về LTSM đều sử dụng một phiên bản hơi khác so với mô hình LTSM chuẩn. Sự khác nhau không lớn, nhưng chúng giúp giải quyết phần nào đó trong cấu trúc của LTSM.

Một dạng LTSM phổ biến được giới thiệu bởi Gers & Schmidhuber (2000) được thêm các đường kết nối “peephole connections”, làm cho các tầng công nhận được giá trị đầu vào là trạng thái tế bào.

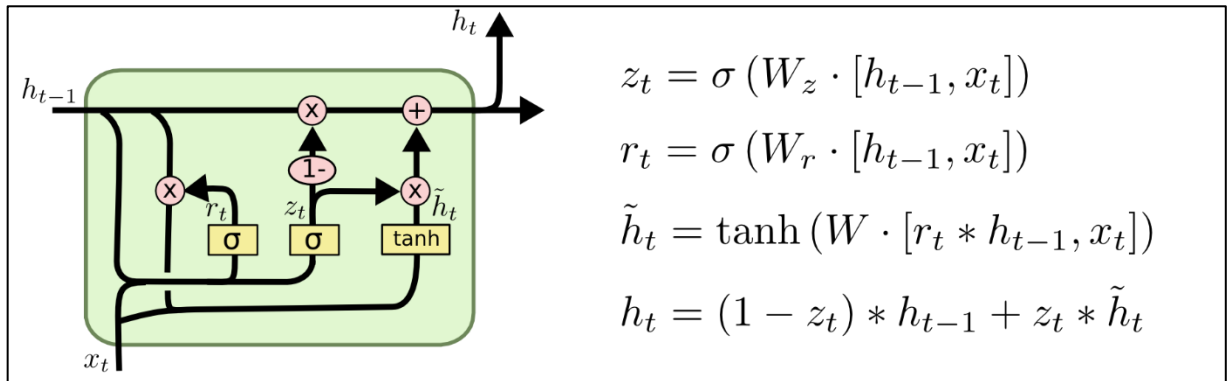


Hình trên mô tả các đường được thêm vào mọi công, nhưng cũng có những bài báo chỉ thêm cho một vài công mà thôi.

Một biến thể khác là nối 2 cổng loại trừ và đầu vào với nhau. Thay vì phân tách các quyết định thông tin loại trừ và thông tin mới thêm vào, ta sẽ quyết định chúng cùng với nhau luôn. Ta chỉ bỏ đi thông tin khi mà ta thay thế nó bằng thông tin mới đưa vào. Ta chỉ đưa thông tin mới vào khi ta bỏ thông tin cũ nào đó đi.



Một biến thể khá thú vị khác của LSTM là Gated Recurrent Unit, hay GRU được giới thiệu bởi Cho, et al. (2014). Nó kết hợp các cổng loại trừ và đầu vào thành một cổng “cổng cập nhật” (update gate). Nó cũng hợp trạng thái tế bào và trạng thái ẩn với nhau tạo ra một thay đổi khác. Kết quả là mô hình của ta sẽ đơn giản hơn mô hình LSTM chuẩn và ngày càng trở nên phổ biến.



2.3.4. Kiến trúc LSTM

Kiến trúc LSTM liên quan đến ô nhớ được điều khiển bởi ba cổng: cổng vào, cổng quên và cổng ra. Các cổng này quyết định thông tin nào sẽ được thêm vào, xóa khỏi và xuất ra khỏi ô nhớ.

Cổng đầu vào kiểm soát thông tin nào được thêm vào ô nhớ.

Cổng quên kiểm soát thông tin nào được xóa khỏi ô nhớ.

Cổng ra điều khiển thông tin nào được đưa ra từ ô nhớ.

Điều này cho phép mạng LSTM giữ lại hoặc loại bỏ thông tin một cách có chọn lọc khi thông tin đó truyền qua mạng, cho phép chúng tìm hiểu các mối phụ thuộc lâu dài.

LSTM duy trì trạng thái ẩn, hoạt động như bộ nhớ ngắn hạn của mạng. Trạng thái ẩn được cập nhật dựa trên đầu vào, trạng thái ẩn trước đó và trạng thái hiện tại của ô nhớ.

Mô hình LSTM hai chiều

Bidirectional LSTM (Bi LSTM/ BLSTM) là mạng nơ-ron hồi quy (RNN) có khả năng xử lý dữ liệu tuần tự theo cả hướng thuận và hướng ngược. Điều này cho phép Bi LSTM học các phụ thuộc tầm xa hơn trong dữ liệu tuần tự so với LSTM truyền thống, chỉ có thể xử lý dữ liệu tuần tự theo một hướng.

Bi LSTM được tạo thành từ hai mạng LSTM, một mạng xử lý chuỗi đầu vào theo hướng thuận và một mạng xử lý chuỗi đầu vào theo hướng ngược.

Đầu ra của hai mạng LSTM sau đó được kết hợp để tạo ra đầu ra cuối cùng.

Các mạng trong kiến trúc LSTM có thể được xếp chồng lên nhau để tạo ra các kiến trúc sâu, cho phép học các mẫu và phân cấp phức tạp hơn nữa trong dữ liệu tuần tự. Mỗi lớp LSTM trong cấu hình xếp chồng nắm bắt các mức độ trừu tượng và phụ thuộc thời gian khác nhau trong dữ liệu đầu vào.

2.3.5. Ứng dụng của LSTM

Mô hình hóa ngôn ngữ: LSTM đã được sử dụng cho các tác vụ xử lý ngôn ngữ tự nhiên như mô hình hóa ngôn ngữ, dịch máy và tóm tắt văn bản. Chúng có thể được đào tạo để tạo ra các câu mạch lạc và đúng ngữ pháp bằng cách học các mối quan hệ phụ thuộc giữa các từ trong một câu.

Nhận dạng giọng nói: LSTM đã được sử dụng cho các tác vụ nhận dạng giọng nói như chuyển giọng nói thành văn bản và nhận dạng các lệnh nói. Chúng có thể được đào tạo để nhận dạng các mẫu trong giọng nói và khớp chúng với văn bản tương ứng.

Dự báo chuỗi thời gian: LSTM đã được sử dụng cho các nhiệm vụ dự báo chuỗi thời gian như dự đoán giá cổ phiếu, thời tiết và mức tiêu thụ năng lượng. Chúng có thể học các mẫu trong dữ liệu chuỗi thời gian và sử dụng chúng để đưa ra dự đoán về các sự kiện trong tương lai.

Phát hiện bất thường: LSTM đã được sử dụng cho các nhiệm vụ phát hiện bất thường như phát hiện gian lận và xâm nhập mạng. Chúng có thể được đào tạo để xác định các mẫu trong dữ liệu lệch khỏi chuẩn mực và đánh dấu chúng là bất thường tiềm ẩn.

Hệ thống đề xuất: LSTM đã được sử dụng cho các tác vụ đề xuất như đề xuất phim, nhạc và sách. Chúng có thể học các mẫu hành vi của người dùng và sử dụng chúng để đưa ra các đề xuất được cá nhân hóa.

Phân tích video: LSTM đã được sử dụng cho các tác vụ phân tích video như phát hiện đối tượng, nhận dạng hoạt động và phân loại hành động. Chúng có thể được sử dụng kết hợp với các kiến trúc mạng nơ-ron khác, chẳng hạn như Mạng nơ-ron tích chập (CNN), để phân tích dữ liệu video và trích xuất thông tin hữu ích.

2.4. Lý thuyết về Vàng và đô la Mỹ

2.4.1. Khái niệm vàng và Đô la Mỹ

2.4.1.1 Vàng

Vàng là nguyên tố hóa học có ký hiệu “Au” và số nguyên tử 79, là một nguyên tố quý, làm cho nó trở thành một trong những nguyên tố có số nguyên tử cao tồn tại ngoài tự nhiên. Vàng là kim loại quý đã được sử dụng làm chất phản xạ neutron trong vũ khí hạt nhân (w71), và để đúc tiền, đồ trang sức và nghệ thuật khác trong suốt lịch sử được ghi lại. Trước đây, vàng thường được thực hiện như một chính sách tiền tệ, nhưng tiền vàng đã không còn được coi là một loại tiền tệ lưu hành trong những năm 1930, và tiêu chuẩn vàng thế giới đã bị thay thế bằng một hệ thống tiền tệ định danh sau năm 1971.

2.4.1.2 Đô la Mỹ

Đồng đô la Mỹ hay Mỹ kim, USD (United States dollar), còn được gọi ngắn là "đô la" hay "đô", là đơn vị tiền tệ chính thức của Hoa Kỳ. Nó cũng được dùng để dự trữ ngoài Hoa Kỳ. Hiện nay, việc phát hành tiền được quản lý bởi các hệ thống ngân hàng của Cục Dự trữ Liên bang (Federal Reserve). Ký hiệu phổ biến nhất cho đơn vị này là dấu “\$”.

2.4.2. Vai trò của vàng và Đô la Mỹ trong nền kinh tế hiện nay

2.4.2.1 Vai trò của Vàng

Vàng đã có nhiều vai trò quan trọng trong lịch sử tài chính và tiền tệ. Hiện nay, vai trò chủ yếu của vàng là một tài sản trữ giá và một công cụ đầu tư. Dưới đây là những vai trò cụ thể của vàng:

- Trữ giá: Vàng thường được xem là một cách để bảo vệ giá trị tài sản trong thời gian dài. Trong những thời kỳ không chắc chắn hoặc khủng hoảng kinh tế, người ta thường tìm đến vàng như một giá trị ổn định và an toàn.

- Đầu tư: Vàng cũng là một cách để đầu tư và đa dạng hóa danh mục đầu tư. Giá vàng có thể tăng lên khi có sự không ổn định kinh tế, lạm phát hoặc giảm giá trị đồng tiền.

2.4.2.2 Vai trò của Đô La Mỹ:

USD có vai trò không lồ trong thương mại, dự trữ ngoại hối và trao đổi tiền tệ toàn cầu. Với tầm quan trọng của Mỹ trên thị trường tài chính và nợ toàn cầu, USD tiếp tục giữ vai trò quốc tế lớn.

- Là công cụ an toàn nhất để dự trữ ngoại hối phải hội tụ được các yếu tố: sự vững mạnh của nền kinh tế hỗ trợ, niềm tin vào giá trị của đồng tiền, sự ổn định và tính thanh khoản trên thị trường quốc tế.
- USD được sử dụng rộng rãi trong các giao dịch thương mại quốc tế, nhiều hàng hóa quan trọng như dầu mỏ, vàng và các nguyên liệu thô khác thường được định giá bằng USD, tạo sự ổn định và thuận tiện trong giao dịch quốc tế.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

3.1. Lý thuyết về ngôn ngữ lập trình

Trong đề án lần này, tôi đã sử dụng “ngôn ngữ R” và “Python” làm ngôn ngữ lập trình chính trong việc thực hiện các dự đoán về XAU/USD (giá vàng/đô la Mỹ) trong tương lai gần. Với ngôn ngữ R sẽ là ngôn ngữ lập trình tôi dùng để xây dựng mô hình ARIMA và Python sẽ là ngôn ngữ lập trình tôi dùng để xây dựng mô hình LSTM.

3.1.1. Ngôn ngữ R

Ngôn ngữ R là một ngôn ngữ lập trình và môi trường có năng lực tính toán mạnh mẽ, rất phổ biến và cực kỳ linh hoạt được sử dụng rộng rãi trong các lĩnh vực thống kê và phân tích dữ liệu. R cung cấp một loạt các gói thư viện, phần mềm phong phú và có chức năng thống kê mạnh mẽ, khiến nó trở thành một trong những công cụ yêu thích của nhiều nhà khoa học dữ liệu và những nhà nghiên cứu.

R có một cú pháp dễ hiểu và tương đối linh hoạt, cho phép người dùng có thể tạo, kiểm tra và triển khai các mô hình phân tích mang khuynh hướng phức tạp. Với cú pháp dễ đọc, R giúp người dùng dễ dàng thực hiện các phân tích thống kê phức tạp và tạo ra những đồ thị để trực quan hóa kết quả.

3.1.2. Python

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh, giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.

3.2. Giới thiệu về dữ liệu

Dữ liệu về các giá XAU/USD, tức giá vàng theo giá đô la Mỹ được bài nghiên cứu sử dụng được cập nhật từ trang web “ <https://vn.investing.com/>”. Trang web Investing.com là một nền tảng tài chính nổi tiếng, cung cấp thông tin và công cụ liên quan đến các thị trường tài chính trên toàn cầu. Các nội dung chính mà trang web này cung cấp bao gồm nhiều dữ liệu quan trọng như:

- Dữ liệu thị trường: gồm giá thời gian thực của các loại cổ phiếu, tiền tệ, hàng hóa, tiền điện tử, và các chỉ số chứng khoán.
- Tin tức và phân tích: trang web còn bao gồm nhiều bài viết, tin tức và những phân tích chuyên sâu về kinh tế, thị trường tài chính, và các sự kiện kinh tế toàn cầu.

Đây là lượng dữ liệu được lấy từ trang web trên và nó chứa dữ liệu về các chỉ số vàng và chỉ số giá đô la Mỹ trong khoảng thời gian gần 10 năm (từ 2/1970 đến 12/2024) để sử dụng trong việc dự đoán trong đề tài này.

Ngày	Lần cuối	Mở	Cao	Thấp
1/12/2024	2,623.81	2,653.82	2,726.31	2,583.49
1/11/2024	2,653.55	2,742.50	2,762.30	2,536.90
1/10/2024	2,743.80	2,635.41	2,790.41	2,604.15
1/9/2024	2,634.49	2,502.74	2,685.96	2,471.95
1/8/2024	2,503.03	2,448.10	2,532.05	2,364.40
1/7/2024	2,448.10	2,326.46	2,483.78	2,318.55
1/6/2024	2,325.71	2,329.61	2,387.85	2,286.77
1/5/2024	2,326.97	2,285.91	2,450.13	2,277.47
1/4/2024	2,285.57	2,239.59	2,431.53	2,228.54
1/3/2024	2,232.38	2,043.44	2,235.90	2,038.55

Bảng 3-1: Một phần dữ liệu giá XAU/USD

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

4.1. Dự đoán bằng mô hình ARIMA

4.1.1. Bước 1: Phân tích, kiểm tra tính mùa vụ của dữ liệu và khử mùa

Đầu tiên, gán bộ dữ liệu về giá XAU/USD (giá vàng theo giá đô la Mỹ) vào các biến data. Sau đó dùng hàm as.Date để định dạng lại ngày, tháng và năm. Sau đó xóa các dấu phẩy ở cột “Price” và cuối cùng chuyển dữ liệu đã xử lý về dạng “time series” và gán nó vào biến “gold_ts”.

```
data <- read_csv('/content/XAU_USD (1.2.1970-1.1.2025) - ENG.csv')
data$Date <- as.Date(data$Date, format = "%m/%d/%Y")
data$Price <- as.numeric(gsub(",", "", data$Price))
gold_ts <- ts(data$Price, start = c(1970, 2), frequency = 12)
gold_ts
```

Gọi biến đã được gán dữ liệu chuyển hóa “time series” ra và ta sẽ quan sát được bộ dữ liệu đã rất gọn gàng:

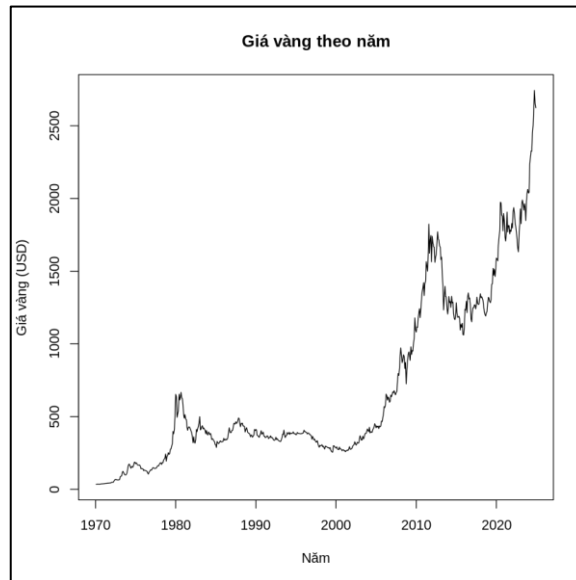
gold_ts

A Time Series: 55 × 12												
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1970		35.20	35.50	36.10	35.85	35.75	35.55	36.10	36.65	37.55	37.80	37.65
1971	38.35	39.10	39.15	39.95	41.15	40.40	42.70	41.00	42.90	42.65	43.95	43.85
1972	47.45	48.50	48.65	49.90	59.75	64.95	68.60	67.15	64.50	64.65	63.90	65.20
1973	66.30	85.30	90.30	91.00	115.05	123.55	115.90	103.80	100.30	98.30	101.25	112.30
1974	132.80	162.80	173.30	169.55	157.25	144.55	156.50	156.30	151.55	167.25	184.55	186.75
1975	176.05	182.00	176.00	167.00	167.20	166.35	166.95	160.00	141.40	143.05	140.70	140.35
1976	128.40	132.55	129.60	128.40	125.50	123.80	112.80	104.35	116.25	123.15	130.25	134.75
1977	132.30	142.75	148.90	147.25	143.30	143.00	144.35	146.00	154.20	161.50	160.00	164.96
1978	176.10	182.25	181.60	170.85	184.15	183.05	200.50	208.70	217.10	242.60	193.40	226.00
1979	233.70	251.55	240.35	245.55	274.85	277.50	296.70	315.35	397.50	382.25	415.90	512.00
1980	653.00	637.00	495.00	518.00	535.50	653.50	614.25	631.25	666.75	629.00	623.00	586.00

Hình 4-1: Dữ liệu giá XAU/USD dạng time series

Tiếp đến, chúng ta sẽ vẽ dữ liệu đồ thị bằng hàm “plot()” để có thể quan sát được xu thế của dữ liệu:

```
plot(gold_ts, main = "Giá vàng theo năm", ylab = "Giá vàng (USD)", xlab = "Năm")
```



Hình 4-2: Xu hướng giá XAU/USD

Biểu đồ cho thấy xu thế biến động liên tục của giá XAU/USD, giá vàng liên tục tăng qua các năm, lên cao nhất qua ba thời điểm bao gồm quanh các năm 1980, 2013 và 2024.

4.1.1.1 Kiểm tra tính mùa vụ

Đối với một chuỗi thời gian có yếu tố mùa vụ (**seasonal time series**) sẽ bao gồm ba thành phần chính: thành phần xu hướng, thành phần theo mùa và thành phần bất thường. Vì vậy, để có thể phân tách một chuỗi thời có yếu tố mùa vụ, chúng ta sẽ tách chuỗi thời gian thành ba thành phần trên.

Để dễ dàng hơn trong việc ước tính các thành phần xu hướng và thành phần theo mùa của chuỗi thời gian có yếu tố mùa vụ dạng mô hình cộng, chúng ta có thể sử dụng hàm “**decompose()**” trong R. Hàm này có chức năng ước tính thành phần xu hướng, thành phần mùa vụ và thành phần bất thường của chuỗi.

Kết quả trả về của hàm “**decompose()**” sẽ trả về cho chúng ta một đối tượng danh sách với kết quả ước tính của các thành phần với các tên gọi tương ứng của chúng là “**Seasonal**”, “**Trend**” và “**Random**”.

Như với dữ liệu chuỗi thời gian của số của đề tài về giá XAU/USD, sử dụng hàm “**decompose()**” để phân rã các chuỗi thời gian có yếu tố mùa vụ và gán nó vào biến “**MUA**”.

```
MUA <- decompose(gold_ts)
```

Các giá trị ước tính của thành phần mùa vụ, xu hướng và bất thường được lưu trong các biến như “**MUA\$seasonal**”, “**MUA\$trend**” và “**MUA\$random**”.

Vì chúng ta đang kiểm tra về tính mùa vụ nên sẽ in ra các giá trị ước tính của thành phần mùa vụ:

```
Mua$seasonal
```

A Time Series: 55 x 12												
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1970		4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1971	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1972	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1973	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1974	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1975	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1976	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1977	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1978	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1979	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629
1980	7.8227202	4.4052819	1.1076507	3.8035072	-0.4861224	-6.9410530	-1.8774526	4.1887773	-2.5563153	-4.1802891	-3.5949419	-1.6917629

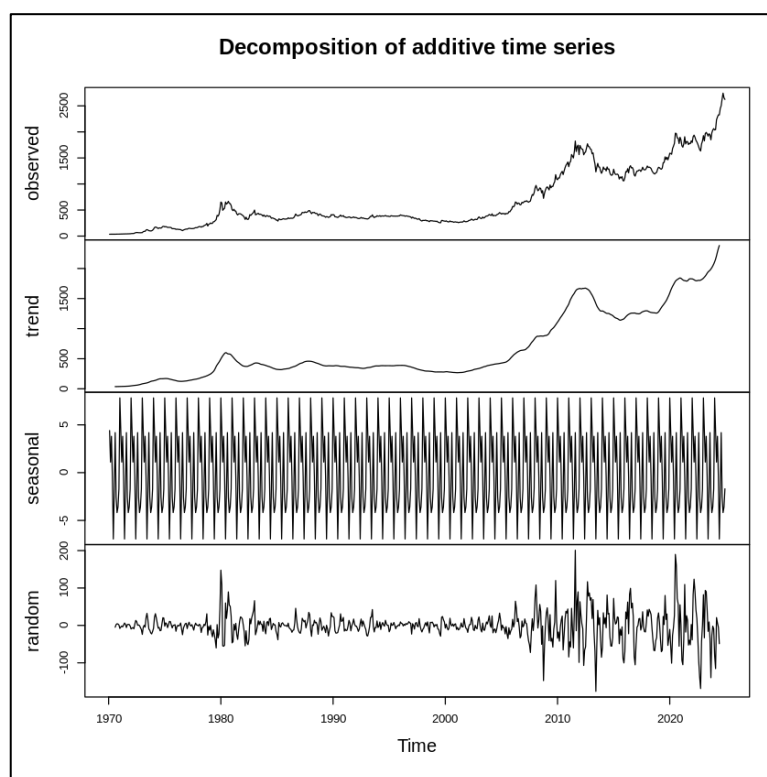
Hình 4-3: Giá trị thành phần mùa vụ của giá XAU/USD

Các yếu tố mùa vụ trên cho thấy được sự ước tính:

Chỉ số mùa vụ cho các tháng từ tháng 1 đến tháng 12 đặc biệt giống nhau cho các năm. Tác động của yếu tố mùa vụ lớn nhất là vào tháng 1 (khoảng 7,82) và thấp nhất là vào tháng 6 (khoảng -6,94). Điều này cho thấy dường như giá XAU/USD có xu thế tăng cao nhất vào tháng 1 và giảm sâu nhất về chỉ số vào tháng 6 hàng năm.

Chúng ta có thể vẽ đồ thị xu hướng ước tính, theo mùa và các thành phần bất thường của chuỗi thời gian bằng cách sử dụng hàm “**plot()**” như sau:

```
plot(MUA)
```



Hình 4-4: Đồ thị đồ thị xu hướng ước tính của giá XAU/USD

Biểu đồ trên hiển thị theo thứ tự:

- + Chuỗi thời gian gốc.
- + Thành phần xu hướng ước tính.
- + Thành phần mùa vụ ước tính.
- + Thành phần bất thường ước tính.

4.1.1.2 Khử yếu tố mùa vụ

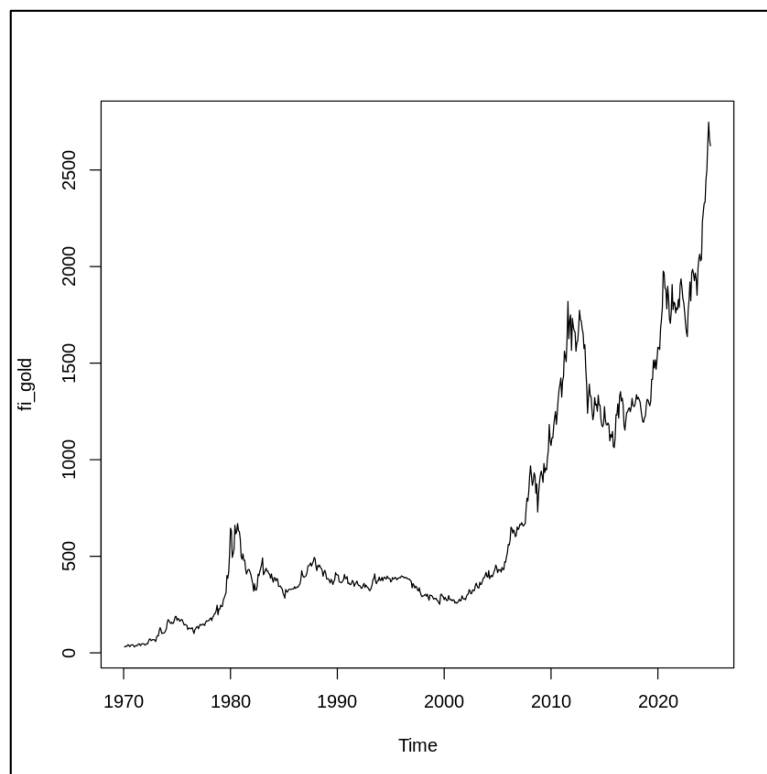
Để khử mùa cũng như là điều chỉnh chuỗi thời gian theo mùa này bằng cách ước tính thành phần mùa vụ và trừ đi thành phần này ước tính này khỏi chuỗi ban đầu.

Theo dữ liệu trên đề tài, ta sẽ điều chỉnh theo yếu tố mùa vụ ở chuỗi thời gian của giá XAU/USD mỗi tháng, chúng ta có thể ước tính thành phần mùa vụ này bằng cách sử dụng hàm “**decompose()**” như đã sử dụng, sau đó trừ thành phần ước tính này khỏi chuỗi thời gian ban đầu và gán vào biến mới “**fi_gold**”.

```
fi_gold <- gold_ts - MUA$seasonal
```

Tiếp tục dùng hàm “**plot()**” hiển thị mô hình dữ liệu và gọi biến “**fi_gold**” để hiển thị chuỗi đã điều chỉnh yếu tố mùa vụ:

```
plot(fi_gold)
fi_gold
```



Hình 4-5: Xu hướng dữ liệu giá XAU/USD đã loại bỏ mùa vụ

	A Time Series: 55 x 12											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1970		30.79472	34.39235	32.29649	36.33612	42.69105	37.42745	31.91122	39.20632	41.73029	41.39494	39.34176
1971	30.52728	34.69472	38.04235	36.14649	41.63612	47.34105	44.57745	36.81122	45.45632	46.83029	47.54494	45.54176
1972	39.62728	44.09472	47.54235	46.09649	60.23612	71.89105	70.47745	62.96122	67.05632	68.83029	67.49494	66.89176
1973	58.47728	80.89472	89.19235	87.19649	115.53612	130.49105	117.77745	99.61122	102.85632	102.48029	104.84494	113.99176
1974	124.97728	158.39472	172.19235	165.74649	157.73612	151.49105	158.37745	152.11122	154.10632	171.43029	188.14494	188.44176
1975	168.22728	177.59472	174.89235	163.19649	167.68612	173.29105	168.82745	155.81122	143.95632	147.23029	144.29494	142.04176
1976	120.57728	128.14472	128.49235	124.59649	125.98612	130.74105	114.67745	100.16122	118.80632	127.33029	133.84494	136.44176
1977	124.47728	138.34472	147.79235	143.44649	143.78612	149.94105	146.22745	141.81122	156.75632	165.68029	163.59494	166.65176
1978	168.27728	177.84472	180.49235	167.04649	184.63612	189.99105	202.37745	204.51122	219.65632	246.78029	196.99494	227.69176
1979	225.87728	247.14472	239.24235	241.74649	275.33612	284.44105	298.57745	311.16122	400.05632	386.43029	419.49494	513.69176
1980	645.17728	632.59472	493.89235	514.19649	535.98612	660.44105	616.12745	627.06122	669.30632	633.18029	626.59494	587.69176

Hình 4-6: Giá trị dữ liệu XAU/USD sau khi loại bỏ mùa vụ

Chúng ta có thể thấy rằng sự biến động theo mùa đã bị loại bỏ khỏi chuỗi thời gian được điều chỉnh theo mùa. Chuỗi thời gian được điều chỉnh theo mùa như trên hiện chỉ chứa 2 thành phần là thành phần xu hướng và thành phần bất thường.

***Lưu ý*:**

Đối với các bộ dữ liệu mang yếu tố mùa vụ thì mô hình ARIMA sẽ có dạng

$$\text{SARIMA}(p,d,q)(P,D,Q)[S]$$

, trong đó:

- + (p,d,q) là ba giá trị ta sẽ xác định được thông qua phân tích dữ liệu đã khử mùa.
- + (P,D,Q) là ba giá trị ta sẽ xác định thông qua phân tích dữ liệu gốc.
- + $[S]$ là biến thể hiện chu kỳ thời gian theo bộ dữ liệu.

4.1.2. Bước 2: Sai phân, kiểm tra tính dừng, mô hình ACF – PACF và chọn các giá trị P, D, Q (dữ liệu chưa khử mùa)

4.1.2.1 Sai phân – kiểm tra tính dừng (dữ liệu chưa khử mùa)

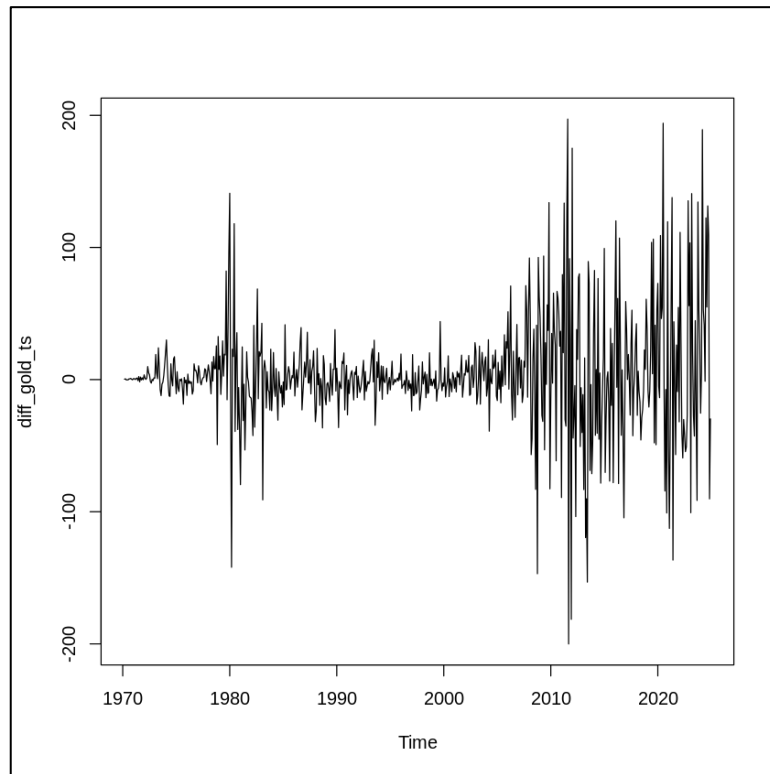
Mô hình SARIMA được xác định cho các chuỗi thời gian dừng, do đó, nếu bắt đầu với một chuỗi không dừng (non-stationary) thì trước tiên sẽ cần lấy sai phân chuỗi cho đến khi có chuỗi dừng. Nếu phải lấy sai phân chuỗi “D” lần để có được chuỗi dừng, thì mô hình SARIMA sẽ có dạng tổng quát là $\text{SARIMA}(p,d,q)(P,D,Q)[S]$. Trong đó, d và D được gọi là bậc tích hợp.

Trên R, Ta có thể sử dụng hàm “**diff()**” để lấy sai phân của một chuỗi.

Trong trường hợp của đề tài này sẽ sai phân 1 lần để xem kết quả.

```
diff_gold_ts <- diff(gold_ts, differences = 1)
plot.ts(diff_gold_ts)
```

Trong đó “gold_ts” là biến chứa dữ liệu chưa khử mùa đã gán ở đầu mục 4.1.1.



Hình 4-7: Biểu đồ dữ liệu chưa khử mùa sau khi sai phân lần 1 của giá XAU/USD

Sau khi sai phân lần 1, chúng ta có thể thấy rằng mô hình đã có tính dừng, vì các dữ liệu chỉ biến thiên xoay quanh một giá trị cố định đó là 0.

=> Giá trị **D = 1**.

4.1.2.2 Mô hình ACF – PACF (dữ liệu chưa khử mùa)

Nếu chuỗi thời gian là chuỗi dừng (bậc gốc hoặc sai phân), thì bước tiếp theo là chọn mô hình SARIMA thích hợp. Điều này nghĩa là tìm giá trị của các giá trị phù hợp nhất của P và Q cho mô hình $ARIMA(p,d,q)(P,D,Q)[S]$. Thông thường, để làm điều này thì phải xem xét biểu đồ tương quan (correlogram) và biểu đồ tương quan từng phần (partial correlogram) của các chuỗi dừng.

Để có thể vẽ được biểu đồ tương quan và biểu đồ tương quan từng phần, chúng ta lần lượt sử dụng các hàm “**acf()**” và hàm “**pacf()**” trong R.

Trở lại ví dụ về các chỉ số, để vẽ biểu đồ tương quan cho các độ trễ đầu tiên của chuỗi sai phân bậc 1, ta thực hiện như sau:

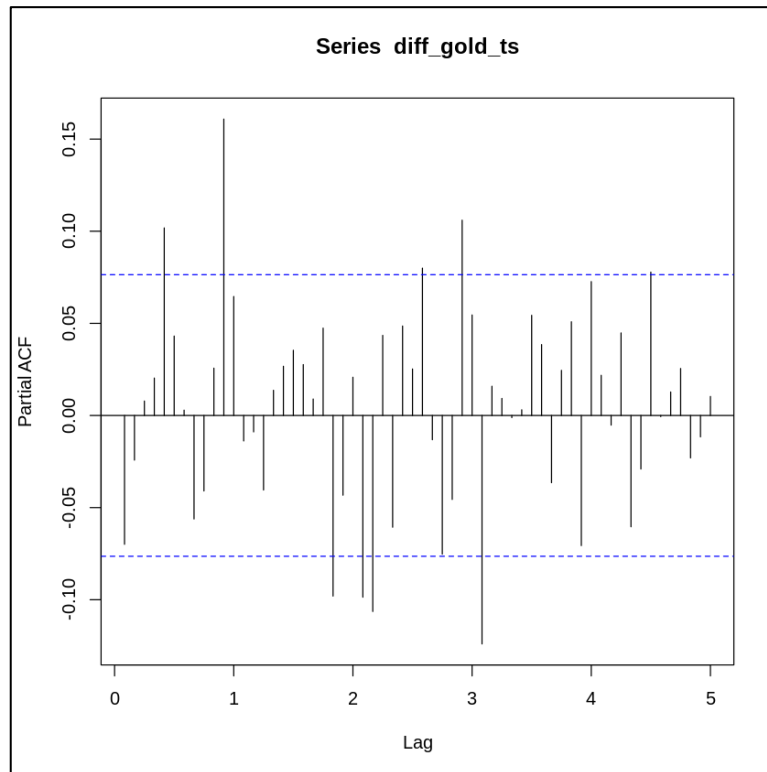
+ Mô hình PACF

```
pacf(diff_gold_ts, lag.max = 60, na.action = na.pass, plot = FALSE)
```

```
Partial autocorrelations of series 'diff_gold_ts', by lag
0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333 0.9167
-0.070 -0.024 0.008 0.020 0.102 0.043 0.003 -0.056 -0.041 0.026 0.161
1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.7500 1.8333
0.065 -0.014 -0.009 -0.040 0.014 0.027 0.035 0.028 0.009 0.047 -0.098
1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.6667 2.7500
-0.043 0.021 -0.099 -0.106 0.043 -0.061 0.049 0.025 0.080 -0.013 -0.075
2.8333 2.9167 3.0000 3.0833 3.1667 3.2500 3.3333 3.4167 3.5000 3.5833 3.6667
-0.046 0.106 0.055 -0.124 0.016 0.009 -0.001 0.003 0.054 0.038 -0.036
3.7500 3.8333 3.9167 4.0000 4.0833 4.1667 4.2500 4.3333 4.4167 4.5000 4.5833
0.024 0.051 -0.071 0.073 0.022 -0.005 0.045 -0.060 -0.029 0.078 -0.001
4.6667 4.7500 4.8333 4.9167 5.0000
0.013 0.026 -0.023 -0.012 0.010
```

Hình 4-8: Giá trị tương quan từng phần ở 60 độ trễ của giá XAU/USD (chưa khử mùa)

```
pacf(diff_gold_ts, lag.max = 60, na.action = na.pass, plot = TRUE)
```



Hình 4-9: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD (chưa khử mùa)

Biểu đồ tương quan từng phần PACF cho thấy về mặt hình thức thì các độ trễ được tính theo con số thập phân nhưng về bản chất thì mỗi giá trị của độ trễ sẽ được tính là một đơn vị số nguyên. Do đó, khi chúng ta nhìn vào biểu đồ cho thấy ở độ trễ 5 và 11 là hai độ trễ có giá trị tương quan tốt và giá trị độ trễ không quá lớn (5: 0,098; 11: 0,158).

=> Giá trị **P = 5 và 11**.

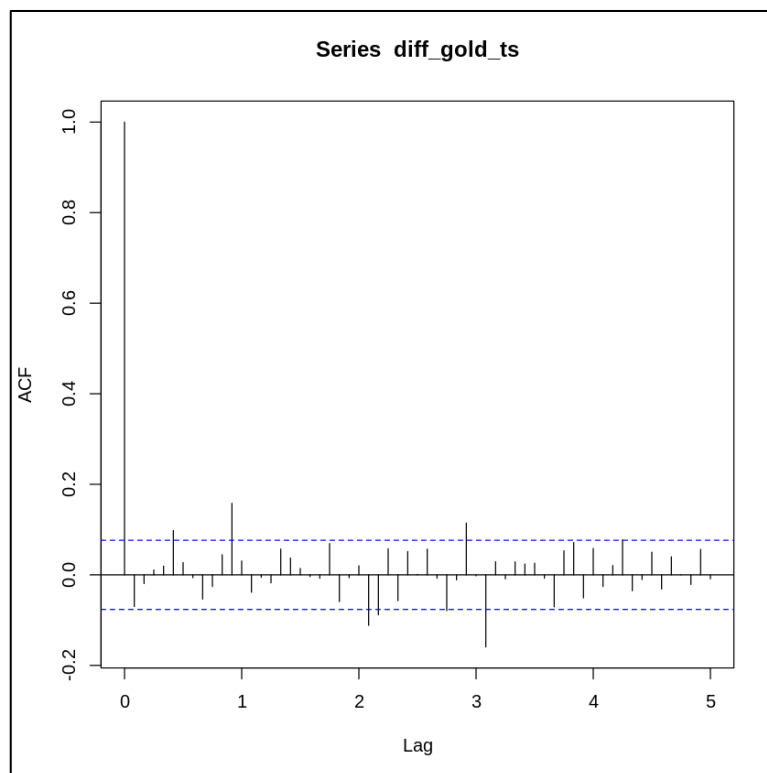
+ Mô hình ACF

```
acf(diff_gold_ts, lag.max = 60, na.action = na.pass, plot = FALSE)
```

```
Autocorrelations of series 'diff_gold_ts', by lag
0.0000 0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333
1.000 -0.070 -0.019 0.011 0.019 0.098 0.027 -0.006 -0.054 -0.026 0.045
0.9167 1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.7500
0.158 0.031 -0.038 -0.005 -0.018 0.057 0.037 0.014 -0.004 -0.008 0.069
1.8333 1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.6667
-0.059 -0.006 0.020 -0.111 -0.088 0.058 -0.057 0.052 0.001 0.057 -0.007
2.7500 2.8333 2.9167 3.0000 3.0833 3.1667 3.2500 3.3333 3.4167 3.5000 3.5833
-0.079 -0.011 0.114 -0.002 -0.159 0.029 -0.009 0.029 0.024 0.026 -0.007
3.6667 3.7500 3.8333 3.9167 4.0000 4.0833 4.1667 4.2500 4.3333 4.4167 4.5000
-0.071 0.053 0.071 -0.051 0.058 -0.026 0.021 0.077 -0.035 -0.010 0.050
4.5833 4.6667 4.7500 4.8333 4.9167 5.0000
-0.031 0.040 0.000 -0.021 0.056 -0.009
```

Hình 4-10: Giá trị tương quan ở 60 độ trễ của giá XAU/USD (chưa khử mùa)

```
acf(diff_gold_ts, lag.max = 60, na.action = na.pass, plot = TRUE)
```



Hình 4-11: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD (chưa khử mùa)

Tương tự với biểu đồ PACF, từ biểu đồ tương quan ACF, chúng ta có thể thấy rằng sự tự tương quan ở độ trễ 5 (0,098) và 11 (0,158) mang giá trị tương quan lớn cắt giá trị tin cậy và giá trị độ trễ cũng không quá cao.

=> Giá trị **Q = 5 và 11**.

+ Lựa chọn mô hình SARIMA

Từ các giá trị đã tìm được **D = 1; P = 1 và 5; Q = 1 và 5**. Chúng ta có thể rút ra được tổng $P*Q = 4$ tổ hợp (P,D,Q) thuộc dữ liệu giá XAU/USD chưa khử mùa:

(5,1,5);

(11,1,5);

(5,1,11);

(11,1,11).

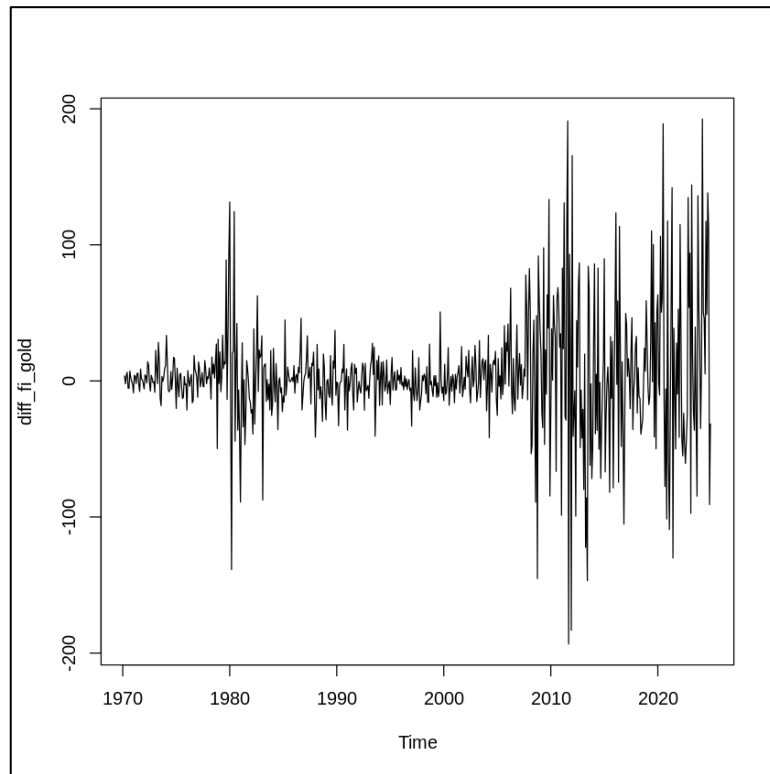
4.1.3. Bước 3: Sai phân, kiểm tra tính dừng, mô hình ACF – PACF và chọn các giá trị p, d, q (dữ liệu đã khử mùa)

4.1.3.1 Sai phân – kiểm tra tính dừng (dữ liệu đã khử mùa)

Tương tự với bộ dữ liệu chưa khử mùa, với dữ liệu khử mùa này vẫn sẽ lấy sai phân bằng hàm “**diff()**” để kiểm tra tính dừng của bộ dữ liệu.

```
diff_fi_gold <- diff(fi_gold, differences = 1)
plot.ts(diff_fi_gold)
```

Trong đoạn code trên, “fi_gold” là biến chứa dữ liệu đã khử mùa mà chúng ta đã gán trong mục 4.1.1.2.



Hình 4-12: Biểu đồ dữ liệu đã khử mùa sau khi sai phân lần 1 của giá XAU/USD

Sau khi sai phân lần 1, tương tự như bộ dữ liệu chưa khử mùa thì chúng ta có thể thấy rằng mô hình đã có tính dừng, vì các dữ liệu chỉ biến thiên xoay quanh một giá trị cố định đó là 0.

=> Giá trị **d** = 1.

4.1.3.2 Mô hình ACF – PACF (dữ liệu đã khử mùa)

Nếu chuỗi thời gian đã là chuỗi dừng thì bước tiếp theo là chọn mô hình SARIMA thích hợp. Điều này nghĩa là tìm giá trị của các giá trị phù hợp nhất của p và q cho mô hình SARIMA(p,d,q)(P,D,Q)[12]. Ở mục hiện tại, ta sẽ tìm p và q cho dữ liệu đã khử mùa.

Để có thể vẽ được biểu đồ tương quan và biểu đồ tương quan từng phần, chúng ta lần lượt sử dụng các hàm “**acf()**” và hàm “**pacf()**” trong R.

Trở lại ví dụ về các chỉ số, để vẽ biểu đồ tương quan các độ trễ đầu tiên của chuỗi sai phân bậc 1, ta thực hiện như sau:

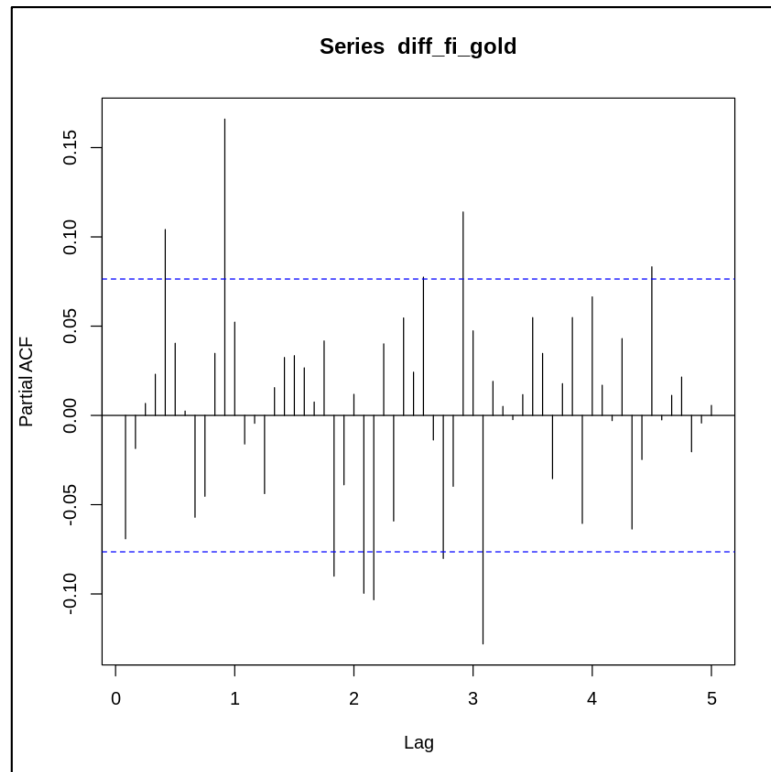
+ Mô hình PACF

```
pacf(diff_fi_gold, lag.max = 60, na.action = na.pass, plot = FALSE)
```

```
Partial autocorrelations of series 'diff_fi_gold', by lag
0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333 0.9167
-0.069 -0.019 0.007 0.023 0.104 0.040 0.002 -0.057 -0.045 0.035 0.166
1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.7500 1.8333
0.052 -0.016 -0.004 -0.044 0.016 0.032 0.033 0.027 0.007 0.042 -0.090
1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.6667 2.7500
-0.039 0.012 -0.100 -0.103 0.040 -0.059 0.055 0.024 0.077 -0.014 -0.080
2.8333 2.9167 3.0000 3.0833 3.1667 3.2500 3.3333 3.4167 3.5000 3.5833 3.6667
-0.040 0.114 0.047 -0.128 0.019 0.005 -0.002 0.012 0.055 0.035 -0.035
3.7500 3.8333 3.9167 4.0000 4.0833 4.1667 4.2500 4.3333 4.4167 4.5000 4.5833
0.018 0.055 -0.060 0.066 0.017 -0.003 0.043 -0.064 -0.025 0.083 -0.002
4.6667 4.7500 4.8333 4.9167 5.0000
0.011 0.021 -0.020 -0.004 0.006
```

Hình 4-13: Giá trị tương quan từng phần ở 60 độ trễ của giá XAU/USD (đã khử mùa)

```
pacf(diff_fi_gold, lag.max = 60, na.action = na.pass, plot = TRUE)
```



Hình 4-14: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD (đã khử mùa)

=> Giá trị **p = 5 và 11**.

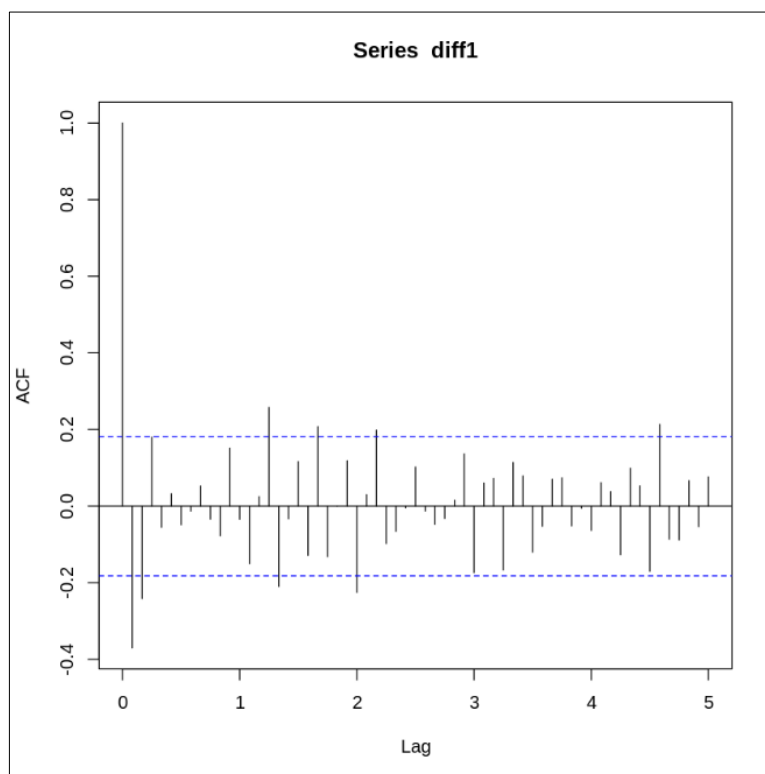
+ Mô hình ACF

```
acf(diff1, lag.max = 60, na.action = na.pass, plot = FALSE)
acf(diff1, lag.max = 60, na.action = na.pass)
```

Autocorrelations of series 'diff1', by lag

0.0000	0.0833	0.1667	0.2500	0.3333	0.4167	0.5000	0.5833	0.6667	0.7500	0.8333
1.000	-0.370	-0.241	0.180	-0.055	0.032	-0.049	-0.013	0.053	-0.034	-0.078
0.9167	1.0000	1.0833	1.1667	1.2500	1.3333	1.4167	1.5000	1.5833	1.6667	1.7500
0.151	-0.034	-0.150	0.025	0.258	-0.210	-0.033	0.116	-0.128	0.208	-0.132
1.8333	1.9167	2.0000	2.0833	2.1667	2.2500	2.3333	2.4167	2.5000	2.5833	2.6667
0.000	0.118	-0.225	0.030	0.198	-0.098	-0.066	-0.004	0.102	-0.013	-0.048
2.7500	2.8333	2.9167	3.0000	3.0833	3.1667	3.2500	3.3333	3.4167	3.5000	3.5833
-0.032	0.016	0.137	-0.173	0.061	0.073	-0.167	0.114	0.079	-0.120	-0.053
3.6667	3.7500	3.8333	3.9167	4.0000	4.0833	4.1667	4.2500	4.3333	4.4167	4.5000
0.070	0.074	-0.052	-0.005	-0.063	0.061	0.038	-0.127	0.099	0.053	-0.170
4.5833	4.6667	4.7500	4.8333	4.9167	5.0000					
0.214	-0.086	-0.088	0.066	-0.054	0.076					

Hình 4-15: Giá trị tương quan ở 60 độ trễ của giá XAU/USD (đã khử mùa)



Hình 4-16: Mô hình tương quan ở 60 độ trễ của giá XAU/USD (đã khử mùa)

=> Giá trị **q = 5 và 11.**

+ Lựa chọn mô hình SARIMA

Từ các giá trị đã tìm được $d = 1$; $p = 5$ và 11 ; $q = 5$ và 11 . Chúng ta có thể rút ra được tổng $p * q = 4$ mô hình thuộc dữ liệu giá XAU/USD đã khử mùa:

(5,1,5);

(11,1,5);

(5,1,11);

(11,1,11).

4.1.4. Bước 4: Lựa chọn mô hình ARIMA

Kết hợp từ các tổ hợp (p,d,q) và (P,D,Q) đã tìm được từ mục 4.2, chúng ta sẽ có được các mô hình ARIMA tổng quát như sau:

(I) 4 Mô hình thuộc dữ liệu chưa khử mùa:

$(5,1,5);$	$(11,1,5);$
$(5,1,11);$	$(11,1,11).$

(II) 4 Mô hình thuộc dữ liệu đã khử mùa:

$(5,1,5);$	$(11,1,5);$
$(5,1,11);$	$(11,1,11).$

Từ (I) và (II) ta sẽ có tổng $(I) * (II) = 16$ mô hình tổng quát và chu kì $[S] = 12$ tháng:

SARIMA(5,1,5)(5,1,5)[12]	SARIMA(11,1,5)(5,1,5)[12]
SARIMA(5,1,5)(5,1,11)[12]	SARIMA(11,1,5)(5,1,11)[12]
SARIMA(5,1,5)(11,1,5)[12]	SARIMA(11,1,5)(11,1,5)[12]
SARIMA(5,1,5)(11,1,11)[12]	SARIMA(11,1,5)(11,1,11)[12]
SARIMA(5,1,11)(5,1,5)[12]	SARIMA(11,1,11)(5,1,5)[12]
SARIMA(5,1,11)(5,1,11)[12]	SARIMA(11,1,11)(5,1,11)[12]
SARIMA(5,1,11)(11,1,5)[12]	SARIMA(11,1,11)(11,1,5)[12]
SARIMA(5,1,11)(11,1,11)[12]	SARIMA(11,1,11)(11,1,11)[12]

Gán lần lượt 16 mô hình vào các biến theo số thứ tự từ model1 đến model16 và chạy chúng.

```
model1 <- arima(gold_ts, order = c(5,1,5), seasonal = list(order =  
c(5,1,5), period = 12))  
model2 <- arima(gold_ts, order = c(5,1,5), seasonal = list(order =  
c(5,1,11), period = 12))
```

```

model3 <- arima(gold_ts, order = c(5,1,5), seasonal = list(order =
c(11,1,5), period = 12))
model4 <- arima(gold_ts, order = c(5,1,5), seasonal = list(order =
c(11,1,11), period = 12))
model5 <- arima(gold_ts, order = c(5,1,11), seasonal = list(order =
c(5,1,5), period = 12))
model6 <- arima(gold_ts, order = c(5,1,11), seasonal = list(order =
c(5,1,11), period = 12))
model7 <- arima(gold_ts, order = c(5,1,11), seasonal = list(order =
c(11,1,5), period = 12))
model8 <- arima(gold_ts, order = c(5,1,11), seasonal = list(order =
c(11,1,11), period = 12))
model9 <- arima(gold_ts, order = c(11,1,5), seasonal = list(order =
c(5,1,5), period = 12))
model10 <- arima(gold_ts, order = c(11,1,5), seasonal = list(order =
c(5,1,11), period = 12))
model11 <- arima(gold_ts, order = c(11,1,5), seasonal = list(order =
c(11,1,5), period = 12))
model12 <- arima(gold_ts, order = c(11,1,5), seasonal = list(order =
c(11,1,11), period = 12))
model13 <- arima(gold_ts, order = c(11,1,11), seasonal = list(order =
c(5,1,5), period = 12))
model14 <- arima(gold_ts, order = c(11,1,11), seasonal = list(order =
c(5,1,11), period = 12))
model15 <- arima(gold_ts, order = c(11,1,11), seasonal = list(order =
c(11,1,5), period = 12))
model16 <- arima(gold_ts, order = c(11,1,11), seasonal = list(order =
c(11,1,11), period = 12))

```

Kết quả sau khi chạy chương trình cho thấy rằng cả 16 mô hình trên đều không thể dùng được khi xuất hiện lỗi:

```

Error in optim(init[mask], armafn, method = optim.method, hessian = TRUE, :
non-finite finite-difference value [3]
Traceback:

```

```

1. optim(init[mask], armafn, method = optim.method, hessian = TRUE,
.   control = optim.control, trans = as.logical(transform.pars))
2. .handleSimpleError(function (cnd)
.   {
.     watcher$capture_plot_and_output()
.     cnd <- sanitize_call(cnd)
.     watcher$push(cnd)
.     switch(on_error, continue = invokeRestart("eval_continue"),
.           stop = invokeRestart("eval_stop"), error =
invokeRestart("eval_error",
.             cnd))
.   }, "non-finite finite-difference value [3]",
base::quote(optim(init[mask],

```

```
.      armafn, method = optim.method, hessian = TRUE, control =
optim.control,
.      trans = as.logical(transform.pars)))
```

"non-finite finite-difference value [3]": lỗi này xảy ra khi trong hàm optim khi khớp mô hình ARIMA cho thấy sự bất ổn định về số trong quá trình tối ưu hóa. Nguyên nhân chính dẫn đến lỗi này là do độ phức tạp của mô hình quá lớn. Mô hình ARIMA ta đang cố gắng khớp (VD: model 1: SARIMA(5,1,5)(5,1,5)[12] và 15 model còn lại) khá phức tạp với số bậc đang xét quá cao cho các thành phần AR, I và MA cho cả hai thành phần mùa vụ và không theo mùa. Độ phức tạp này đã khiến quá trình tối ưu hóa gặp khó khăn, dẫn đến các vấn đề về hội tụ, tức là thuật toán tối ưu hóa không thể tìm thấy được một bộ tham số mô hình tối ưu thỏa đáng.

Do đó chúng ta cần phải lựa chọn lại các tham số AR, I và MA cho mô hình.

Chúng ta sẽ quay trở về từ bước chọn tham số AR và MA cho hai thành phần mùa vụ và không theo mùa.

4.1.4.1 Mô hình ACF – PACF (dữ liệu chưa khử mùa)

+ Mô hình PACF

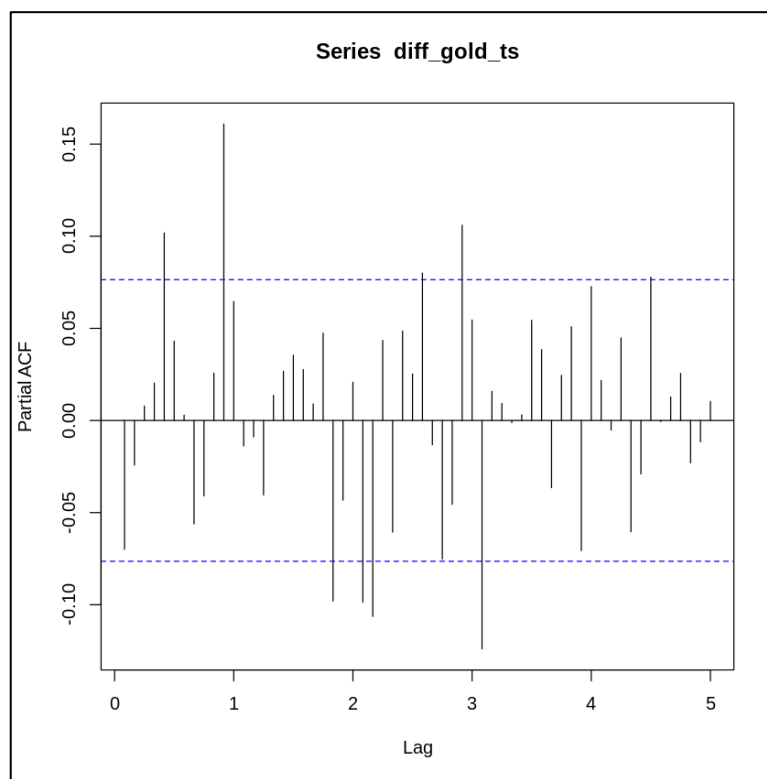
```
pacf(diff_gold_ts, lag.max = 60, na.action = na.pass, plot = FALSE)
```

```
Partial autocorrelations of series 'diff_gold_ts', by lag

0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333 0.9167
-0.070 -0.024 0.008 0.020 0.102 0.043 0.003 -0.056 -0.041 0.026 0.161
1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.7500 1.8333
0.065 -0.014 -0.009 -0.040 0.014 0.027 0.035 0.028 0.009 0.047 -0.098
1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.6667 2.7500
-0.043 0.021 -0.099 -0.106 0.043 -0.061 0.049 0.025 0.080 -0.013 -0.075
2.8333 2.9167 3.0000 3.0833 3.1667 3.2500 3.3333 3.4167 3.5000 3.5833 3.6667
-0.046 0.106 0.055 -0.124 0.016 0.009 -0.001 0.003 0.054 0.038 -0.036
3.7500 3.8333 3.9167 4.0000 4.0833 4.1667 4.2500 4.3333 4.4167 4.5000 4.5833
0.024 0.051 -0.071 0.073 0.022 -0.005 0.045 -0.060 -0.029 0.078 -0.001
4.6667 4.7500 4.8333 4.9167 5.0000
0.013 0.026 -0.023 -0.012 0.010
```

Hình 4-17: Giá trị tương quan từng phần ở 60 độ trễ của giá XAU/USD

```
pacf(diff_gold_ts, lag.max = 60, na.action = na.pass, plot = TRUE)
```

Hình 4-18: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD

Vì độ trễ 5 là quá lớn để có thể train được mô hình, nên chúng ta sẽ chọn độ trễ bé hơn 5 và có giá trị tương quan lớn nhất. Theo điều kiện đó chúng ta có thể thấy được độ trễ số 1 là độ trễ có giá trị tương quan cao nhất (0,070). Do đó ta sẽ chọn 1 làm số đại diện cho P.

=> Giá trị **P = 1**.

+ Mô hình ACF

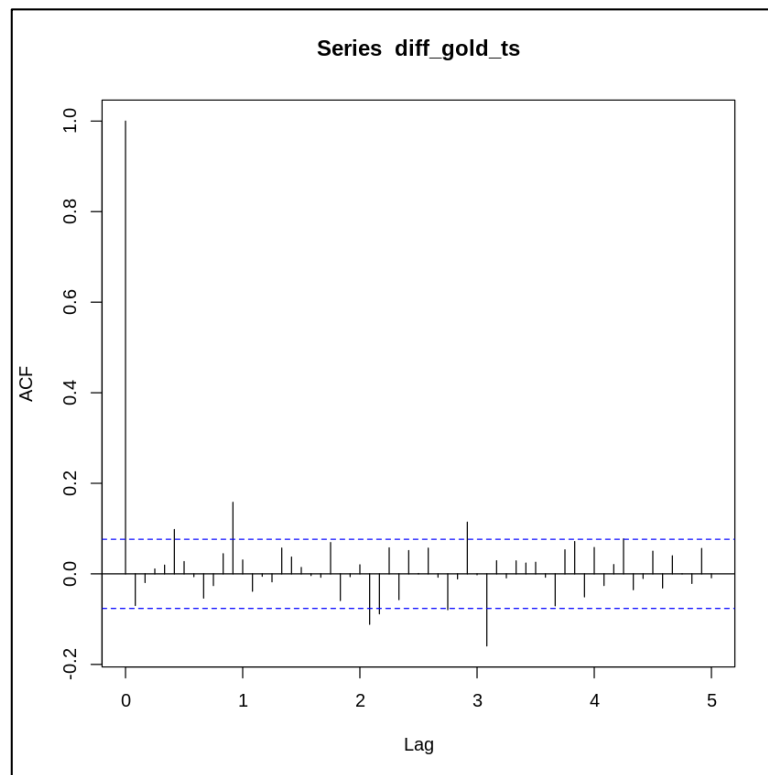
```
acf(diff_gold_ts, lag.max = 60, na.action = na.pass, plot = FALSE)
```

Autocorrelations of series 'diff_gold_ts', by lag

```
0.0000 0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333
1.000 -0.070 -0.019 0.011 0.019 0.098 0.027 -0.006 -0.054 -0.026 0.045
0.9167 1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.7500
0.158 0.031 -0.038 -0.005 -0.018 0.057 0.037 0.014 -0.004 -0.008 0.069
1.8333 1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.6667
-0.059 -0.006 0.020 -0.111 -0.088 0.058 -0.057 0.052 0.001 0.057 -0.007
2.7500 2.8333 2.9167 3.0000 3.0833 3.1667 3.2500 3.3333 3.4167 3.5000 3.5833
-0.079 -0.011 0.114 -0.002 -0.159 0.029 -0.009 0.029 0.024 0.026 -0.007
3.6667 3.7500 3.8333 3.9167 4.0000 4.0833 4.1667 4.2500 4.3333 4.4167 4.5000
-0.071 0.053 0.071 -0.051 0.058 -0.026 0.021 0.077 -0.035 -0.010 0.050
4.5833 4.6667 4.7500 4.8333 4.9167 5.0000
-0.031 0.040 0.000 -0.021 0.056 -0.009
```

Hình 4-19: Giá trị tương quan ở 60 độ trễ của giá XAU/USD

```
acf(diff_gold_ts, lag.max = 60, na.action = na.pass, plot = TRUE)
```



Hình 4-20: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD

Tương tự với biểu đồ PACF, từ biểu đồ tương quan ACF, chúng ta có thể thấy rằng sự tự tương quan ở độ trễ 1 (-0,070) mang giá trị tương quan lớn nhất và giá trị độ trễ cũng bé hơn 5.

=> Giá trị **Q = 1**.

+ Lựa chọn mô hình ARIMA

Từ các giá trị đã tìm được **D = 1; P = 1; Q = 1**. Chúng ta có thể rút ra được tổng $P*Q = 1$ tổ hợp (P,D,Q) thuộc bộ dữ liệu đô la Mỹ chưa khử mùa:

(1,1,1);

4.1.4.2 Mô hình ACF – PACF (dữ liệu đã khử mùa)

+ Mô hình PACF

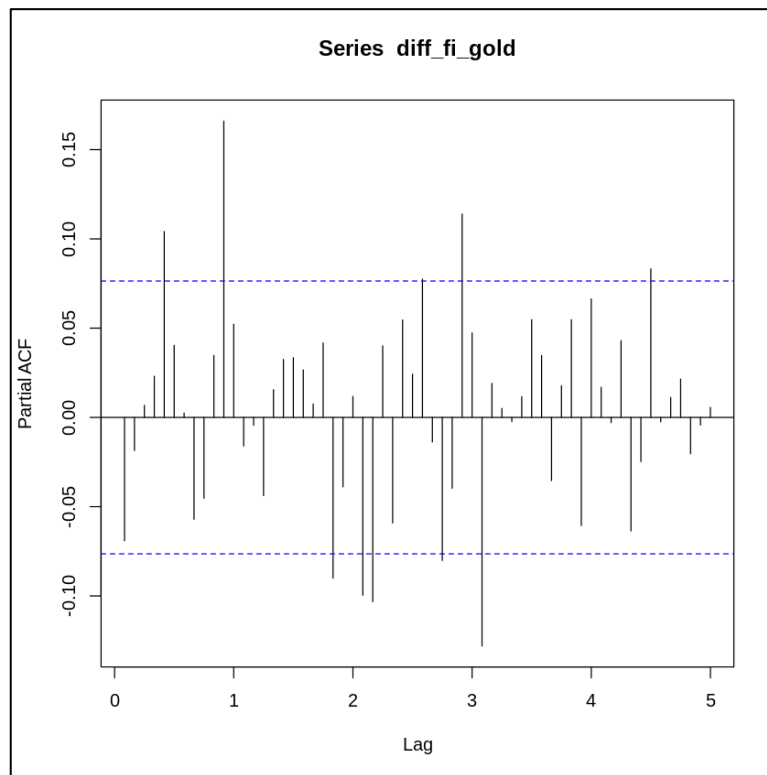
```
pacf(diff_fi_gold, lag.max = 60, na.action = na.pass, plot = FALSE)
```

```
Partial autocorrelations of series 'diff_fi_gold', by lag

0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333 0.9167
-0.069 -0.019 0.007 0.023 0.104 0.040 0.002 -0.057 -0.045 0.035 0.166
1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.7500 1.8333
0.052 -0.016 -0.004 -0.044 0.016 0.032 0.033 0.027 0.007 0.042 -0.090
1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.6667 2.7500
-0.039 0.012 -0.100 -0.103 0.040 -0.059 0.055 0.024 0.077 -0.014 -0.080
2.8333 2.9167 3.0000 3.0833 3.1667 3.2500 3.3333 3.4167 3.5000 3.5833 3.6667
-0.040 0.114 0.047 -0.128 0.019 0.005 -0.002 0.012 0.055 0.035 -0.035
3.7500 3.8333 3.9167 4.0000 4.0833 4.1667 4.2500 4.3333 4.4167 4.5000 4.5833
0.018 0.055 -0.060 0.066 0.017 -0.003 0.043 -0.064 -0.025 0.083 -0.002
4.6667 4.7500 4.8333 4.9167 5.0000
0.011 0.021 -0.020 -0.004 0.006
```

Hình 4-21: Giá trị tương quan từng phần ở 60 độ trễ của giá XAU/USD (đã khử mùa)

```
pacf(diff_fi_gold, lag.max = 60, na.action = na.pass, plot = TRUE)
```



Hình 4-22: Mô hình tương quan từng phần ở 60 độ trễ của giá XAU/USD (đã khử mùa)

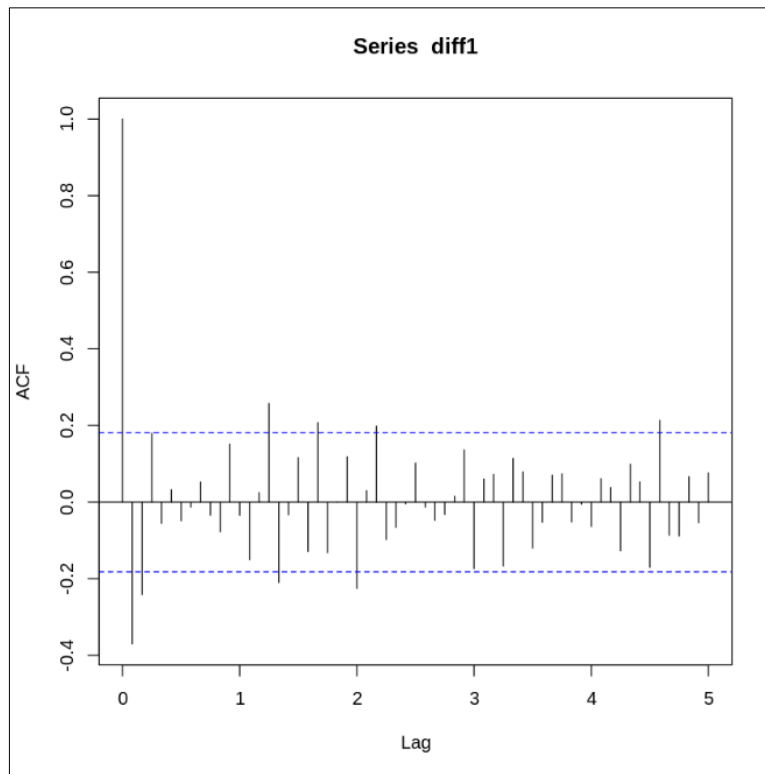
=> Giá trị **p = 1**.

+ Mô hình ACF

```
acf(diff1, lag.max = 60, na.action = na.pass, plot = FALSE)
acf(diff1, lag.max = 60, na.action = na.pass)
```

```
Autocorrelations of series 'diff1', by lag
0.0000 0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333
1.000 -0.370 -0.241 0.180 -0.055 0.032 -0.049 -0.013 0.053 -0.034 -0.078
0.9167 1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.7500
0.151 -0.034 -0.150 0.025 0.258 -0.210 -0.033 0.116 -0.128 0.208 -0.132
1.8333 1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.6667
0.000 0.118 -0.225 0.030 0.198 -0.098 -0.066 -0.004 0.102 -0.013 -0.048
2.7500 2.8333 2.9167 3.0000 3.0833 3.1667 3.2500 3.3333 3.4167 3.5000 3.5833
-0.032 0.016 0.137 -0.173 0.061 0.073 -0.167 0.114 0.079 -0.120 -0.053
3.6667 3.7500 3.8333 3.9167 4.0000 4.0833 4.1667 4.2500 4.3333 4.4167 4.5000
0.070 0.074 -0.052 -0.005 -0.063 0.061 0.038 -0.127 0.099 0.053 -0.170
4.5833 4.6667 4.7500 4.8333 4.9167 5.0000
0.214 -0.086 -0.088 0.066 -0.054 0.076
```

Hình 4-23: Giá trị tương quan ở 60 độ trễ của giá XAU/USD (đã khử mùa)



Hình 4-24: Mô hình tương quan ở 60 độ trễ của giá XAU/USD (đã khử mùa)

=> Giá trị **q = 1**.

+ Lựa chọn mô hình ARIMA

Từ các giá trị đã tìm được $d = 1$; $p = 1$; $q = 1$. Chúng ta có thể rút ra được tổng p * q = 4 mô hình thuộc dữ liệu đã khử mùa:

(1,1,1);

4.1.4.3 Bước 4: Lựa chọn mô hình ARIMA

- **Mô hình của `auto.arima`**

Trước khi chúng ta kết hợp các mô hình theo hướng thủ công thì ta có thể khởi chạy hàm “**auto.arima()**” trong gói thư viện “**forecast**” vì nó có thể được sử dụng để tìm mô hình ARIMA thích hợp một cách máy móc.

- Dữ liệu chưa khử mùa:

```
forecast::auto.arima(diff_gold_ts)
```

```
[74] auto.arima(diff_gold_ts)

Series: diff_gold_ts
ARIMA(0,1,3)

Coefficients:
          ma1      ma2      ma3
        -1.0711  0.0539  0.0344
s.e.      0.0390  0.0567  0.0410

sigma^2 = 1803: log likelihood = -3395.3
AIC=6798.59  AICc=6798.66  BIC=6816.54
```

Hình 4-25: Kết quả khởi chạy autoarima cho dữ liệu chưa khử mùa

Auto arima đề cử 3 chỉ số (P,D,Q) cho SARIMA là (0,1,3)

- Dữ liệu đã khử mùa:

```
forecast::auto.arima(diff_fi_gold)
```

```
auto.arima(diff_fi_gold)

Series: diff_fi_gold
ARIMA(4,1,1)

Coefficients:
          ar1      ar2      ar3      ar4      ma1
        -0.0900 -0.0400 -0.0117  0.0068 -0.9803
s.e.      0.0449  0.0464  0.0460  0.0444  0.0231

sigma^2 = 1787: log likelihood = -3391.46
AIC=6794.93  AICc=6795.06  BIC=6821.86
```

Hình 4-26: Kết quả khởi chạy autoarima cho dữ liệu đã khử mùa

Auto arima đề cử 3 chỉ số (p,d,q) cho SARIMA là (4,1,1)

- Tổng kết:

Kết quả cho thấy hàm `auto.arima()` của cả 2 dữ liệu đã đề cử mô hình “**SARIMA(4,1,1)(0,1,3)[12]**” là mô hình thích hợp nhất cho bộ dữ liệu.

- **Mô hình chọn thủ công**

Kết hợp từ các tổ hợp (p,d,q) và (P,D,Q) đã tìm được từ mục 4.1.4.1 và 4.1.4.2, chúng ta sẽ có được các mô hình ARIMA tổng quát như sau:

(I) 1 Mô hình thuộc dữ liệu chưa khử mùa: $(1,1,1)$

(II) 1 Mô hình thuộc dữ liệu đã khử mùa: $(1,1,1)$

Từ (I) và (II) ta sẽ có tổng $(I) * (II) = 1$ mô hình tổng quát và chu kỳ $[S] = 12$ tháng:

$$\text{SARIMA}(1,1,1)(1,1,1)[12]$$

- **Mô hình AIC, BIC**

Trong phân tích dữ liệu chuỗi thời gian, AIC (Akaike's Information Criterion) và BIC (Bayesian Information Criterion) là hai phương pháp chính dùng để đánh giá mô hình dự báo. Cả hai đều được sử dụng để đo lường sự tối ưu của mô hình dự báo dựa trên dữ liệu quan sát được.

Cả hai mô hình AIC và BIC đều sử dụng nguyên lý đơn giản là cố gắng cân bằng giữa khả năng phù hợp của mô hình và độ phức tạp của mô hình. Tuy nhiên, cách tiếp cận và công thức tính toán của chúng có một số khác biệt.

AIC có mục tiêu là tìm mô hình có AIC score thấp nhất, tức là mô hình có sự phù hợp tốt nhất với dữ liệu với độ phức tạp thấp nhất.

BIC có mục tiêu là tìm mô hình có BIC score thấp nhất, tức là mô hình có sự phù hợp tốt nhất với dữ liệu với độ phức tạp thấp nhất, đồng thời cân nhắc số lượng quan sát.

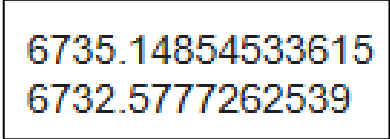
Khi áp dụng ARIMA cho một chuỗi dữ liệu, chúng ta thường cần xác định các siêu tham số như p , d và q . Để đảm bảo độ tin cậy và chất lượng của mô hình ARIMA, chúng ta có thể sử dụng AIC và BIC để kiểm tra sự tối ưu của các mô hình khác nhau, sau đó chọn mô hình có AIC hoặc BIC score thấp nhất. Mô hình có AIC hoặc BIC score thấp nhất được coi là mô hình tốt nhất hoặc tối ưu nhất cho dữ liệu đang xem xét.

Đầu tiên gán 2 mô hình ARIMA đã tìm được vào biến `modelA` và `modelAuto`:

```
modelA <- arima(gold_ts, order = c(1,1,1), seasonal = list(order =
c(1,1,1), period = 12))
modelAuto <- arima(gold_ts, order = c(4,1,1), seasonal = list(order =
c(0,1,3), period = 12))
```

Tiếp đến dùng hàm “AIC()” và “BIC()” để tính toán hai chỉ số AIC và BIC cho hai mô hình.

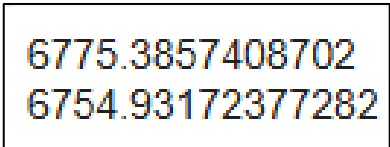
```
AIC(modelAuto)
AIC(modelA)
```



```
6735.14854533615
6732.5777262539
```

Hình 4-27: Giá trị AIC cho hai mô hình arima đã chọn

```
BIC(modelAuto)
BIC(modelA)
```



```
6775.3857408702
6754.93172377282
```

Hình 4-28: Giá trị BIC cho hai mô hình arima đã chọn

Dựa vào hai mô hình AIC và BIC trên, ta có thể thấy được rằng ở cả hai mô hình, modelA là mô hình dự đoán có hai chỉ số AIC và BIC thấp nhất trong tổng 2 mô hình .

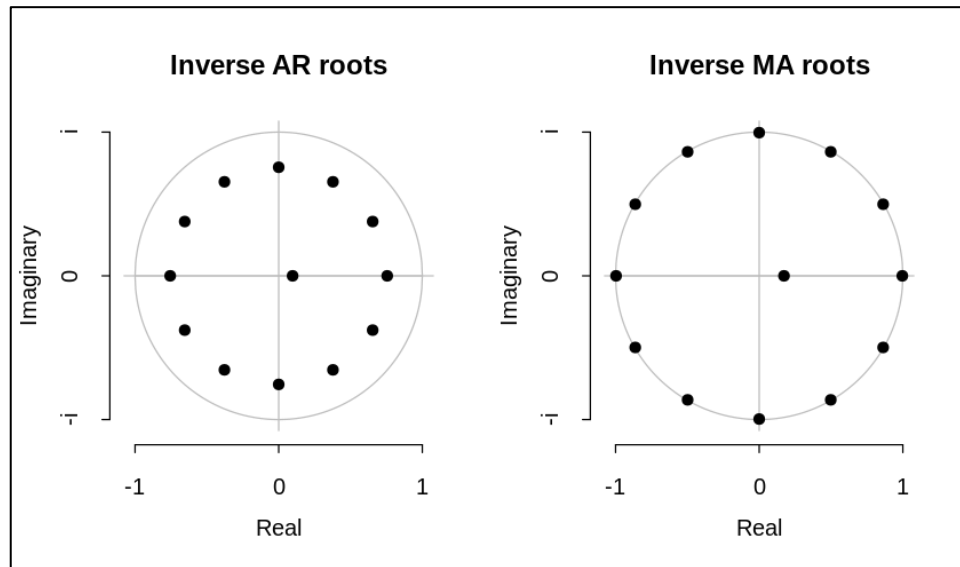
=> Chọn mô hình “arima(2,1,2)(1,1,1)[12]” để dự báo chỉ số giá vàng vì có ý nghĩa thống kê tốt nhất.

4.1.5. Bước 5: Kiểm định mô hình đã chọn

- **Kiểm định nghiệm nghịch đảo AR và MA**

Một mô hình ARIMA chuẩn thì giá trị tuyệt đối của các chỉ số AR và MA đều phải nhỏ hơn 1 và các nghiệm nghịch đảo này buộc phải nằm bên trong đường tròn đơn vị. Do đó chúng ta sẽ sử dụng hàm “plot()” để vẽ ra mô hình các nghiệm nghịch đảo của AR, MA và sau đó quan sát.

```
plot(modelA)
```

Hình 4-29: Mô hình các nghiệm nghịch đảo của giá XAU/USD

Như chúng ta có thể thấy trong mô hình rằng tất cả các nghiệm nghịch đảo AR và MA của mô hình đều nằm ở bên trong đường tròn đơn vị có bán kính bằng 1.

=> Mô hình đã thỏa mãn điều kiện về tính ổn định.

Tiếp đến sẽ kiểm tra những kiểm định liên quan đến sai số để có thể quan sát được các sai số có phụ thuộc vào quá khứ hay không.

- **Kiểm tra Ljung-Box cho phần dư**

Kiểm tra Ljung-Box cho phần dư là một dạng kiểm định thống kê dùng để đánh giá tính ngẫu nhiên của phần dư trong mô hình chuỗi thời gian. Nói một cách đơn giản, kiểm định này sẽ kiểm tra xem liệu phần dư có phải là nhiễu trắng hay không.

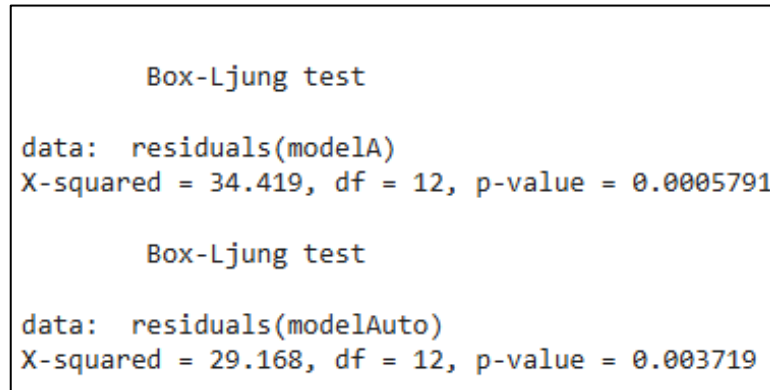
Ta có 2 giả thuyết:

- H_0 : Phần dư không có tương quan tự động.
- H_1 : Phần dư có tương quan tự động.

Nếu giá trị p lớn hơn mức ý nghĩa (0.05) => Không bác bỏ giả thuyết vô hiệu rằng phần dư là nhiễu trắng. Cho thấy mô hình phù hợp với dữ liệu.

Nếu giá trị p nhỏ hơn mức ý nghĩa (0.05) => Bác bỏ giả thuyết vô hiệu và kết luận phần dư không phải là nhiễu trắng. Cho thấy mô hình chưa phù hợp với dữ liệu và cần được cải thiện.

```
Box.test(residuals(modelA), lag = 12, type = "Ljung-Box")
Box.test(residuals(modelAuto), lag = 12, type = "Ljung-Box")
```



```
Box-Ljung test

data: residuals(modelA)
X-squared = 34.419, df = 12, p-value = 0.0005791

Box-Ljung test

data: residuals(modelAuto)
X-squared = 29.168, df = 12, p-value = 0.003719
```

Hình 4-30: Kết quả kiểm định Ljung-Bô cho mô hình modelA

Kết quả:

- **modelA:** p-value = 0.0005791 < 0.05

=> Bác bỏ giả thuyết H0.

Mô hình modelA có p-value nhỏ hơn mức ý nghĩa 0.05. Điều này cho thấy rằng phần dư của cả hai mô hình **đều có tương quan tự động**.

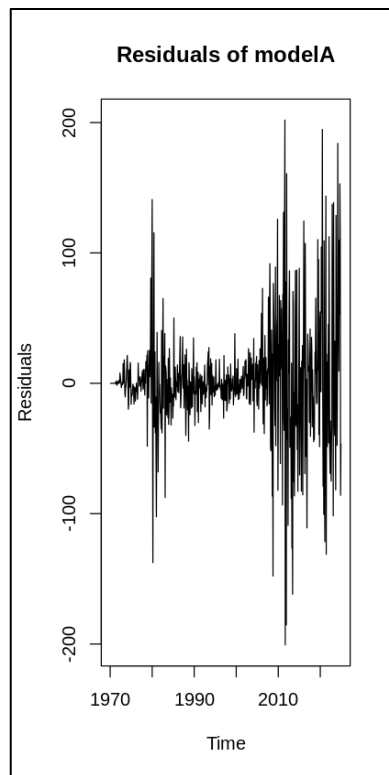
=> Các mô hình này chưa nắm bắt hết được thông tin trong dữ liệu và có thể cần được cải thiện.

- **Kiểm tra đồ thị phần dư**

Đồ thị phần dư là biểu đồ thể hiện sự khác biệt giữa giá trị thực tế của chuỗi thời gian và giá trị dự đoán bởi mô hình, nó biểu diễn phần dư của mô hình, tức là phần mà mô hình không thể giải thích được. Đồ thị phần dư giúp = đánh giá chất lượng của mô hình. Nếu mô hình phù hợp với dữ liệu, phần dư sẽ phân bố ngẫu nhiên xung quanh đường 0 và không có xu hướng rõ ràng. Ngược lại, nếu phần dư có xu hướng hoặc mô hình nào đó, điều này cho thấy mô hình chưa phù hợp với dữ liệu và cần được cải thiện.

```
par(mfrow = c(1, 2))
```

```
plot(residuals(modelA), main = "Residuals of modelA", ylab =  
"Residuals")
```



Hình 4-31: Mô hình đồ thị phần dư cho mô hình modelA

Phần dư của mô hình đều quá tương đồng và phân bố ngẫu nhiên xung quanh đường 0, điều này cho thấy mô hình đã giải thích được phần lớn biến động trong dữ liệu và là hai mô hình tốt.

4.1.5.1 Dự báo sử dụng mô hình modelA

Chúng ta có thể sử dụng mô hình ARIMA để thực hiện dự báo các giá trị tương lai của các chuỗi bằng cách sử dụng hàm “**forecast.Arima()**” với thiết lập số kỳ dự báo ở tùy chọn “h” (ở hai bộ dữ liệu này mang ý nghĩa tháng) như sau.

Dự đoán giá XAU/USD sẽ thay đổi trong vòng 12 tháng “h = 12” bằng mô hình modelA đã được kiểm định:

Mô hình modelA

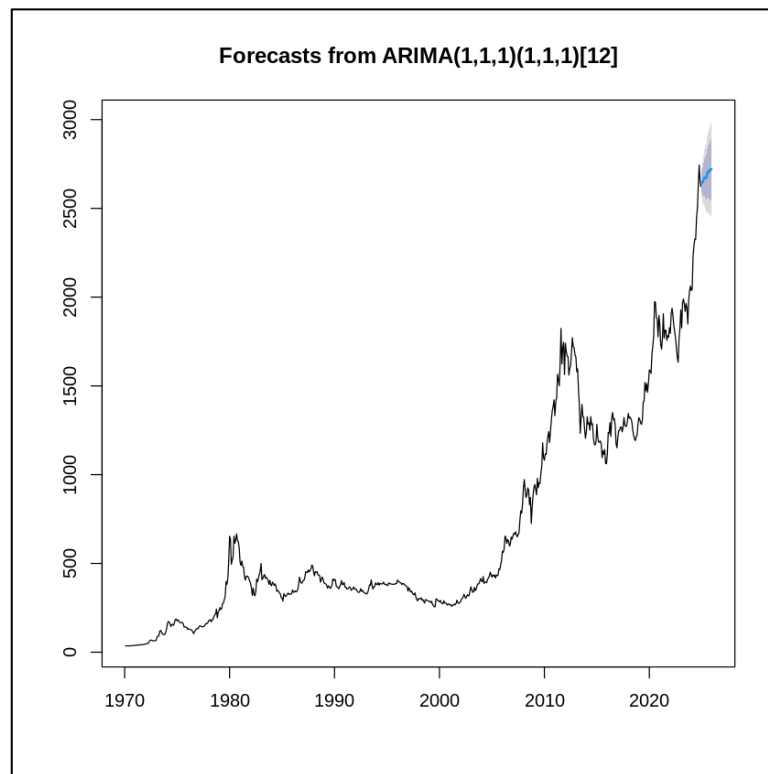
```
predictA <- forecast::forecast.Arima(modelA, h = 12)
```

```
predictA
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2025	2645.575	2590.412	2700.738	2561.210	2729.940
Feb 2025	2646.553	2571.438	2721.669	2531.674	2761.432
Mar 2025	2661.338	2570.782	2751.895	2522.844	2799.833
Apr 2025	2674.264	2570.560	2777.969	2515.662	2832.867
May 2025	2675.443	2560.081	2790.805	2499.012	2851.874
Jun 2025	2670.846	2544.901	2796.791	2478.229	2863.462
Jul 2025	2691.824	2556.119	2827.529	2484.281	2899.367
Aug 2025	2706.929	2562.120	2851.737	2485.462	2928.395
Sep 2025	2704.224	2550.851	2857.597	2469.660	2938.788
Oct 2025	2718.128	2556.644	2879.612	2471.160	2965.097
Nov 2025	2714.342	2545.136	2883.549	2455.563	2973.121
Dec 2025	2721.700	2545.109	2898.292	2451.627	2991.774

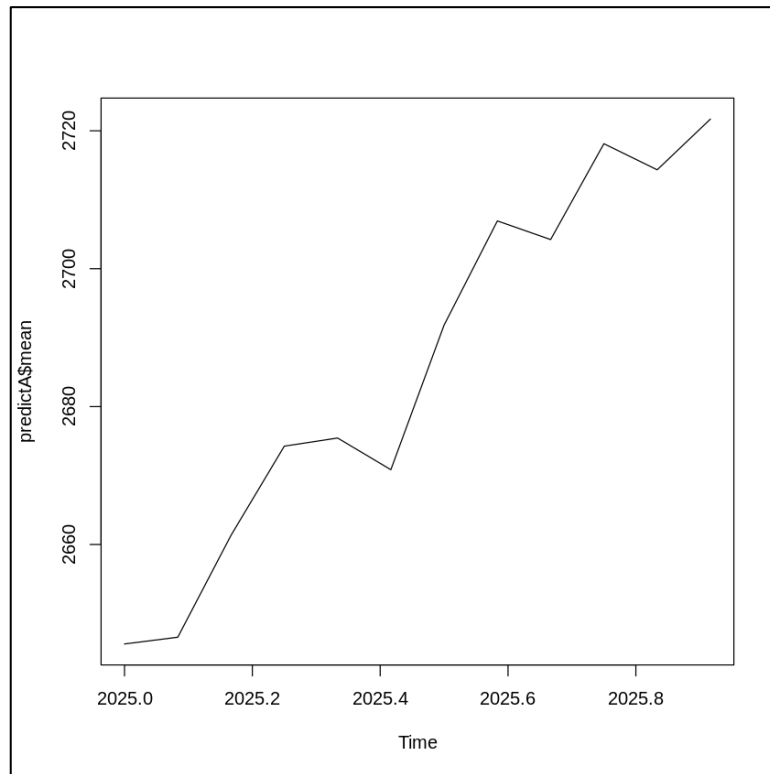
Hình 4-32: Kết quả dự đoán của mô hình modelA

```
forecast:::plot.forecast(predictA)
```



Hình 4-33: Đồ thị dự đoán của mô hình modelA

```
plot(predictA$mean)
```



Hình 4-34: Biểu đồ phân dự đoán của modelA

Hàm “**forecast.Arima()**” cho ta dự báo chỉ số vàng trong 12 tháng tiếp theo, cũng như tính toán khoảng dự đoán 80% và 95% cho những dự đoán đó.

So sánh kết quả dự đoán với thực tế 1 tháng đầu năm 2025

Thời gian	1/1/2025
Thực tế	2,639.49
ModelA	2,645.58

Hình 4-35: Giá trị chỉ số vàng thực tế ba tháng cuối năm 2022

Chúng ta có thể thấy rằng sự dự báo từ đến từ mô hình modelA ở khoảng tin cậy 95% đều khá chính xác chỉ cách nhau khoảng 6 USD so với thực tế.

4.2. Dự đoán bằng mô hình LSTM

4.2.1. Bước 1: Tiền xử lý dữ liệu

Vì mô hình LSTM được viết bằng ngôn ngữ Python nên sẽ có đôi chút khác biệt về mặt ngữ nghĩa và chữ viết của code.

Đầu tiên sẽ đọc dữ liệu từ file .csv chứa dữ liệu của giá XAU/USD.

```
df = pd.read_csv('/content/drive/MyDrive/Test for LSTM/XAU_USD  
(1.2.1970-1.1.2025) - VI.csv')  
df = df.drop(columns=["KL", "% Thay đổi"])  
print(df)
```

	Ngày	Lần cuối	Mở	Cao	Thấp
0	01/12/2024	2,623.81	2,653.82	2,726.31	2,583.49
1	01/11/2024	2,653.55	2,742.50	2,762.30	2,536.90
2	01/10/2024	2,743.80	2,635.41	2,790.41	2,604.15
3	01/09/2024	2,634.49	2,502.74	2,685.96	2,471.95
4	01/08/2024	2,503.03	2,448.10	2,532.05	2,364.40
..
654	01/06/1970	35.75	35.60	35.75	35.75
655	01/05/1970	35.85	36.15	35.85	35.85
656	01/04/1970	36.10	35.55	36.10	36.10
657	01/03/1970	35.50	35.15	35.50	35.50
658	01/02/1970	35.20	35.25	35.20	35.20

[659 rows x 5 columns]

Hình 4-36: Giá XAU/USD sau khi đọc file

Sau đó định dạng lại cột ngày theo công thức dd/mm/yyyy để dễ dàng xử lý được dữ liệu cho sau này.

```
df["Ngày"] = pd.to_datetime(df.Ngày, format="%d/%m/%Y")  
df.head()
```

	Ngày	Lần cuối	Mở	Cao	Thấp
0	2024-12-01	2,623.81	2,653.82	2,726.31	2,583.49
1	2024-11-01	2,653.55	2,742.50	2,762.30	2,536.90
2	2024-10-01	2,743.80	2,635.41	2,790.41	2,604.15
3	2024-09-01	2,634.49	2,502.74	2,685.96	2,471.95
4	2024-08-01	2,503.03	2,448.10	2,532.05	2,364.40

Hình 4-37: Dữ liệu sau khi đã được định dạng lại cột ngày

Tiếp đến dùng hàm **“`.info()`”** để có thể xem được thông tin của dữ liệu.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 659 entries, 0 to 658
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0   Ngày        659 non-null    datetime64[ns]
 1   Lần cuối    659 non-null    object
 2   Mở          659 non-null    object
 3   Cao         659 non-null    object
 4   Thấp        659 non-null    object
dtypes: datetime64[ns](1), object(4)
memory usage: 25.9+ KB
```

Hình 4-38: Thông tin của dữ liệu XAU/USD

Theo thông tin, ta có thể thấy tất cả các cột dữ liệu đều không có giá trị rỗng, cột ngày có định dạng hiện tại là datetime và các cột giá như Lần Cuối, Mở, Cao, Thấp đều có kiểu dữ liệu là object. Và toàn bộ file dữ liệu có tổng cộng 659 dòng dữ liệu và 5 cột nội dung dữ liệu.

Bởi vì các cột giá đều có kiểu dữ liệu là “object”, không thể sử dụng trong việc dự đoán. Vì vậy chúng ta cần chuyển đổi kiểu dữ liệu của chúng. lấy cột giá “Lần Cuối” làm cột thông tin chính để xây dựng mô hình.

```
df['Ngày'] = pd.to_datetime(df['Ngày'], format='%d/%m/%Y')
df = df.sort_values(by='Ngày')
df['Lần cuối'] = df['Lần cuối'].astype(str).str.replace(',', ' ').astype(float)
df['Mở'] = df['Mở'].astype(str).str.replace(',', ' ').astype(float)
df['Cao'] = df['Cao'].astype(str).str.replace(',', ' ').astype(float)
df['Thấp'] = df['Thấp'].astype(str).str.replace(',', ' ').astype(float)

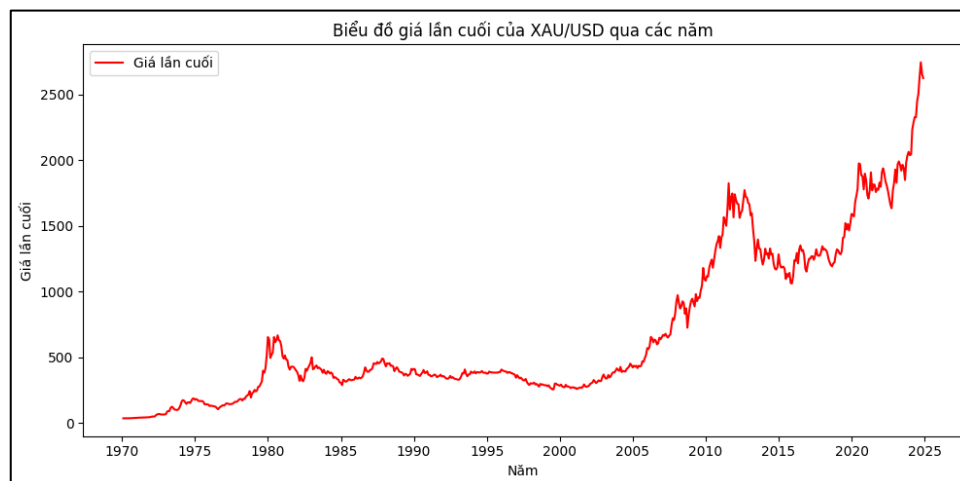
df['Năm'] = df['Ngày'].dt.year

plt.figure(figsize=(10, 5))
plt.plot(df['Ngày'], df['Lần cuối'], label='Giá lần cuối', color='red')
plt.xlabel('Năm')
plt.ylabel('Giá lần cuối')
plt.title('Biểu đồ giá lần cuối của XAU/USD qua các năm')
plt.legend(loc='best')

years = YearLocator(base=5)

plt.gca().xaxis.set_major_locator(years)
plt.gca().xaxis.set_major_formatter(DateFormatter('%Y'))

plt.tight_layout()
plt.show()
```



Hình 4-39: Biểu đồ của dữ liệu XAU/USD sau khi đã xử lý

```
df1 = pd.DataFrame(df, columns=['Ngày', 'Lần cuối'])
```



```
df1.index = df1.Ngày
df1.drop('Ngày', axis=1, inplace=True)
df1
```

Lần cuối	
Ngày	
1970-02-01	35.20
1970-03-01	35.50
1970-04-01	36.10
1970-05-01	35.85
1970-06-01	35.75
...	...
2024-08-01	2503.03
2024-09-01	2634.49
2024-10-01	2743.80
2024-11-01	2653.55
2024-12-01	2623.81
659 rows × 1 columns	

Hình 4-40: Hai cột giá trị chính dùng để huấn luyện mô hình

Tiếp theo, để có thể huấn luyện mô hình, chúng ta cần phải chia tập mô hình ra làm 2 phần theo tỷ lệ phổ biến nhất là [8:2], tức là 8 phần đầu dữ liệu sẽ dùng để huấn luyện mô hình và 2 phần còn lại dùng để kiểm tra mô hình đã được huấn luyện.

Ở đây, vì dữ liệu có đến 659 dòng dữ liệu nên ta sẽ lấy 550 dữ liệu đầu để huấn luyện và số lượng dữ liệu còn lại dùng để kiểm tra.

```
data = df1.values
train_data = data[:550]
test_data = data[550:]
```

Bước tiếp theo, ta cần phải chuẩn hóa dữ liệu, việc chuẩn hóa dữ liệu là một bước quan trọng trước khi đưa vào huấn luyện mô hình LSTM. Cụ thể, nó thực hiện việc co giãn tất cả các giá trị của dữ liệu về trong khoảng từ 0 đến 1. Điều này sẽ giúp nâng cao hiệu suất của mô hình, tăng tốc độ huấn luyện và giúp cho việc so sánh các biến trở nên dễ dàng hơn. Ta sẽ chuẩn hóa dữ liệu bằng hàm “MinMaxScaler”.

```
sc = MinMaxScaler(feature_range=(0,1))
sc_train = sc.fit_transform(data)
```

Tiếp đến sẽ tạo vòng lặp các giá trị bao gồm 1 vòng lặp sẽ gồm 50 giá trị tính từ giá trị ban đầu đến giá trị thứ 50, 1 vòng lặp như vậy dùng để mô hình LSTM học tập từ dữ liệu cũ và sau đó dự đoán ra giá trị mới thứ 51. sau đó tiếp tục lấy giá trị thứ 51 cùng với 49 điểm dữ liệu liền kề trước đó để học và dự đoán ra giá trị thứ 52. Tiếp diễn cho đến giới hạn của mỗi tập huấn luyện.

```
x_train,y_train=[],[]
for i in range(50,len(train_data)):
    x_train.append(sc_train[i-50:i,0]) #lấy 50 giá đóng cửa liên tục
    y_train.append(sc_train[i,0]) #lấy ra giá đóng cửa ngày hôm sau
```

Sau khi chạy vòng lặp các giá trị thì ta sẽ có tập huấn luyện x_train gồm:

```
x_train
[array([0.    , 0.00011076, 0.00033227, 0.00023998, 0.00020306,
        0.00012922, 0.00033227, 0.00053533, 0.00086761, 0.00095991,
        0.00090453, 0.00116296, 0.00143986, 0.00145832, 0.00175367,
        0.00219671, 0.00191981, 0.00276896, 0.00214133, 0.0028428 ,
        0.0027505 , 0.00323045, 0.00319353, 0.00452263, 0.00491029,
        0.00496566, 0.00542716, 0.00906372, 0.01098353, 0.01233109,
        0.01179576, 0.0108174 , 0.01087278, 0.01059588, 0.01107583,
        0.01148195, 0.01849664, 0.02034261, 0.02060105, 0.02948017,
        0.03261833, 0.02979399, 0.02532674, 0.02403456, 0.02329617,
        0.02438529, 0.02846489, 0.03603338, 0.04710921, 0.05098575]),
 array([0.00011076, 0.00033227, 0.00023998, 0.00020306, 0.00012922,
        0.00033227, 0.00053533, 0.00086761, 0.00095991, 0.00090453,
        0.00116296, 0.00143986, 0.00145832, 0.00175367, 0.00219671,
        0.00191981, 0.00276896, 0.00214133, 0.0028428 , 0.0027505 ,
        0.00323045, 0.00319353, 0.00452263, 0.00491029, 0.00496566,
        0.00542716, 0.00906372, 0.01098353, 0.01233109, 0.01179576,
        0.0108174 , 0.01087278, 0.01059588, 0.01107583, 0.01148195,
        0.01849664, 0.02034261, 0.02060105, 0.02948017, 0.03261833,
        0.02979399, 0.02532674, 0.02403456, 0.02329617, 0.02438529,
        0.02846489, 0.03603338, 0.04710921, 0.05098575, 0.04960127])]
```

Và `y_train`: Chứa giá trị đóng cửa của ngày hôm sau, tương ứng với mỗi chuỗi trong `x_train`. Đây là giá trị mà mô hình cần học để dự đoán.

```
y_train
[0.04960127002879715,
 0.04506017869009819,
 0.040371409584287074,
 0.04478328287676289,
 0.04470944399320681,
 0.04295577050874991,
 0.04875212286790223,
 0.0551391862955032,
 0.05595141401462009,
 0.052001033744369776, ...]
```

Đến hai bước cuối cùng của xử lý dữ liệu là xếp dữ liệu thành mảng 2 chiều và 3 chiều.

Xếp dữ liệu thành mảng 2 chiều là việc chuyển đổi dữ liệu từ dạng danh sách sang dạng mảng NumPy để có thể sử dụng các hàm và phương thức của NumPy cho việc tính toán và xử lý dữ liệu hiệu quả hơn.

Ban đầu, `x_train` và `y_train` được tạo ra dưới dạng danh sách, mà danh sách trong Python là một kiểu dữ liệu linh hoạt nhưng không hiệu quả cho các phép toán số học. Hơn hết, mảng NumPy được thiết kế để xử lý dữ liệu số một cách nhanh chóng và hiệu quả nên việc chuyển đổi sang mảng NumPy, tức mảng hai chiều là cần thiết.

```
x_train = np.array(x_train)
y_train = np.array(y_train)
```

Đối với mảng 3 chiều thì mục đích chính là định dạng lại dữ liệu đầu vào cho phù hợp với yêu cầu của mô hình LSTM trong Keras. Vì mô hình LSTM là một loại mạng nơ-ron hồi quy yêu cầu dữ liệu đầu vào có dạng 3 chiều: (Số lượng mẫu dữ liệu, Số bước thời gian trong mỗi mẫu, Số lượng đặc trưng của mỗi bước thời gian). Do đó phải xếp dữ liệu thêm một lần thành mảng 3 chiều mới có thể train mô hình LSTM.

```
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
y_train = np.reshape(y_train, (y_train.shape[0], 1))
```

4.2.2. Bước 2: Xây dựng mô hình

Đầu tiên, tạo một mô hình Sequential, là một dạng mô hình tuyến tính trong Keras, nơi các lớp được thêm vào theo thứ tự. Tiếp đến sẽ thêm một lớp LSTM với 128 nút, xác định hình dạng đầu vào của mô hình, số lượng timesteps (theo mục 4.2.1 là 50, tương ứng với 50 ngày dữ liệu trước đó) và số lượng features (trong trường hợp này chỉ có giá đóng cửa).

Sau đó thêm một lớp LSTM thứ hai với 64 nút. Lớp này nhận đầu ra chuỗi từ lớp LSTM trước đó.

Thêm một lớp Dropout với tỷ lệ 0.2. Lớp này giúp ngăn chặn việc mô hình bị overfitting (học tủ) bằng cách ngẫu nhiên bỏ qua 20% các units trong quá trình huấn luyện.

Thêm một lớp Dense với 1 nút. Đây là lớp đầu ra của mô hình, dùng để dự đoán giá vàng.

Cuối cùng là biên dịch mô hình với hàm loss là 'mean_absolute_error' (sai số tuyệt đối trung bình) và trình tối ưu hóa là 'adam'. Hàm loss đo lường sự khác biệt giữa giá trị dự đoán - giá trị thực tế. Trình tối ưu hóa được sử dụng để điều chỉnh các trọng số của mô hình nhằm giảm thiểu hàm loss.

```
model = Sequential()
model.add(LSTM(units=128, input_shape=(x_train.shape[1], 1), return_sequences=True))
model.add(LSTM(units=64))
model.add(Dropout(0.5))
model.add(Dense(1))
model.compile(loss='mean_absolute_error', optimizer='adam')
```

Đến bước bắt đầu huấn luyện mô hình. Đầu tiên lưu đường dẫn để lưu mô hình đã huấn luyện tốt nhất vào file save_model.keras trong thư mục Google Drive.

```
save_model = "/content/drive/MyDrive/Test for LSTM/save_model.keras" #
```

Tiếp đến sử dụng hàm ModelCheckpoint để lưu lại mô hình tốt nhất trong quá trình huấn luyện. Trong đó:

save_model: Đường dẫn để lưu mô hình.

monitor='loss': Theo dõi giá trị loss function để xác định mô hình tốt nhất.

`verbose=2`: In ra thông tin chi tiết trong quá trình huấn luyện.

`save_best_only=True`: Chỉ lưu mô hình tốt nhất (loss function thấp nhất).

`mode='auto'`: Tự động xác định lưu (min/max) dựa trên giá trị monitor.

```
best_model =  
ModelCheckpoint(save_model,monitor='loss',verbose=2,save_best_only=True,  
mode='auto')
```

Bắt đầu quá trình huấn luyện mô hình:

`x_train`: Dữ liệu đầu vào cho huấn luyện.

`y_train`: Dữ liệu đầu ra tương ứng với `x_train`.

`epochs=100`: Số lần lặp lại toàn bộ dữ liệu huấn luyện.

`batch_size=50`: Số lượng mẫu dữ liệu được sử dụng trong mỗi lần cập nhật trọng số của mô hình.

`verbose=2`: In ra thông tin chi tiết trong quá trình huấn luyện.

`callbacks=[best_model]`: Sử dụng callback `best_model` để lưu lại mô hình tốt nhất.

```
model.fit(x_train,y_train,epochs=100,batch_size=50,verbose=2,callbacks=[  
best_model])
```

Sau khi chạy các code trên thì đây là một phần kết quả chạy:

Epoch 1/100

```
Epoch 1: loss improved from inf to 0.08040, saving model to  
/content/drive/MyDrive/Test for LSTM/save_model.keras  
10/10 - 4s - 392ms/step - loss: 0.0804  
Epoch 2/100
```

```
Epoch 2: loss improved from 0.08040 to 0.04525, saving model to  
/content/drive/MyDrive/Test for LSTM/save_model.keras  
10/10 - 1s - 117ms/step - loss: 0.0453  
Epoch 3/100
```

```
Epoch 3: loss improved from 0.04525 to 0.03371, saving model to  
/content/drive/MyDrive/Test for LSTM/save_model.keras  
10/10 - 2s - 231ms/step - loss: 0.0337  
Epoch 4/100
```

```
Epoch 4: loss improved from 0.03371 to 0.03066, saving model to
/content/drive/MyDrive/Test for LSTM/save_model.keras
10/10 - 3s - 261ms/step - loss: 0.0307
Epoch 5/100
```

```
Epoch 5: loss improved from 0.03066 to 0.02967, saving model to
/content/drive/MyDrive/Test for LSTM/save_model.keras
10/10 - 1s - 137ms/step - loss: 0.0297
Epoch 6/100
```

Ta có thể thấy khi huấn luyện mô hình thì sẽ tự động lưu mô hình nào có hàm mất mát bé nhất và sẽ bỏ qua những model có hàm mất mát lớn hơn.

Bước cuối cùng của bước huấn luyện này sẽ là lấy giá trị thực tế của giá đóng cửa trên tập train (`y_train`). Sử dụng mô hình đã huấn luyện để dự đoán giá đóng cửa trên tập train và chuyển đổi kết quả dự đoán về giá trị thực tế (`y_train_predict`).

```
y_train = sc.inverse_transform(y_train)
final_model = load_model("/content/drive/MyDrive/Test for
LSTM/save_model.keras")
y_train_predict = final_model.predict(x_train)
y_train_predict = sc.inverse_transform(y_train_predict)
```

4.2.3. Bước 3: Tái sử dụng mô hình

Việc tái sử dụng mô hình trên tập `test_data` là bước quan trọng để đánh giá hiệu suất và khả năng dự đoán của mô hình trên dữ liệu mới, chưa từng được sử dụng trong quá trình huấn luyện.

Xử lý dữ liệu test: Đoạn code chuẩn bị dữ liệu test (`x_test`) tương tự như cách đã làm với dữ liệu huấn luyện. Điều này đảm bảo dữ liệu test có định dạng phù hợp để đưa vào mô hình.

```
test = df1[len(train_data)-50:].values
test = test.reshape(-1,1)
sc_test = sc.transform(test)

x_test = []
for i in range(50,test.shape[0]):
    x_test.append(sc_test[i-50:i,0])
x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0],x_test.shape[1],1))
```

Dự đoán: Sử dụng `final_model.predict(x_test)` để dự đoán giá trị trên tập `test_data`.

Đảo ngược chuẩn hóa: Sử dụng `sc.inverse_transform` để chuyển giá trị dự đoán về dạng ban đầu.

```
y_test = data[550:]
y_test_predict = final_model.predict(x_test)
y_test_predict = sc.inverse_transform(y_test_predict)
```

4.2.4. Bước 4: Kiểm tra độ chính xác của mô hình

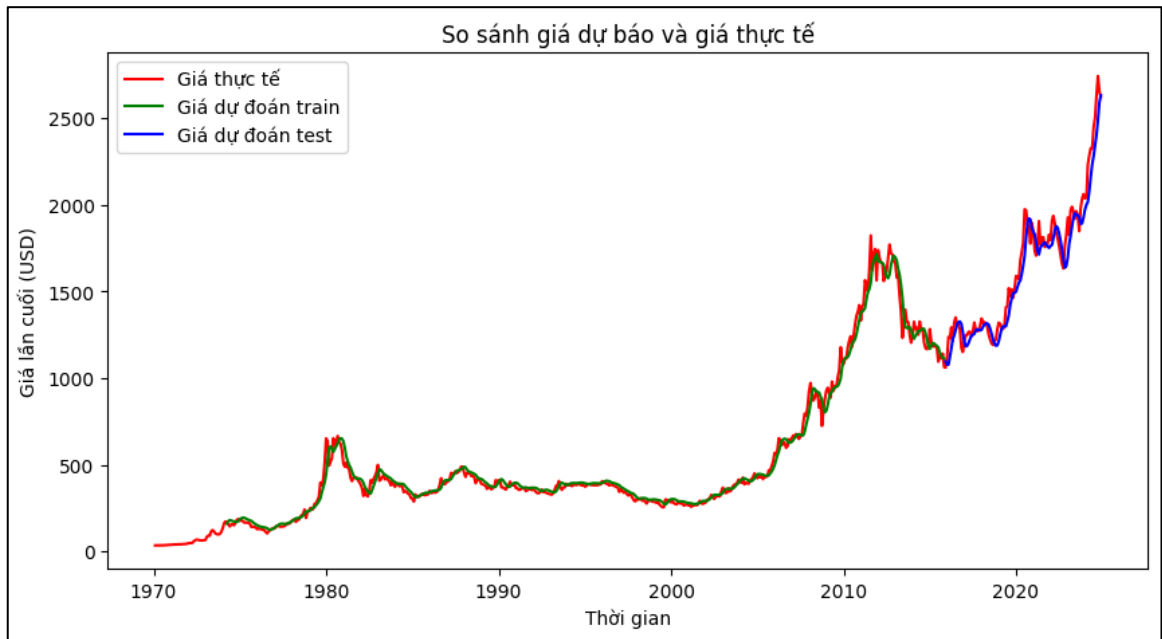
Đầu tiên, kiểm tra độ chính xác của mô hình bằng việc lập biểu đồ so sánh giữa 3 dữ liệu bao gồm: thực tế, tập huấn luyện và tập kiểm tra.

```
train_data1 = df1[50:550]
test_data1 = df1[550:]

plt.figure(figsize=(10,5))
plt.plot(df1,label='Giá thực tế',color='red')
train_data1['Dự đoán'] = y_train_predict
plt.plot(train_data1['Dự đoán'],label='Giá dự đoán train',color='green')

test_data1['Dự đoán'] = y_test_predict
plt.plot(test_data1['Dự đoán'],label='Giá dự đoán test',color='blue')

plt.title('So sánh giá dự báo và giá thực tế')
plt.xlabel('Thời gian')
plt.ylabel('Giá lần cuối (USD)')
plt.legend()
plt.show()
```



Hình 4-41: Biểu đồ so sánh giá dự báo và giá thực tế của XAU/USD

Biểu đồ cho thấy giá dự đoán huấn luyện (màu xanh lá) có xu hướng rất khớp so với xu hướng giá thực tế. Nhưng so với giá dự đoán thì giá dự đoán kiểm tra (xanh dương) lại có xu hướng lệch hơn so với xu hướng giá thực tế. Vì vậy ta nên kiểm tra thêm một vài chỉ số để hiểu rõ mô hình hơn.

```
#r2
print('Độ phù hợp tập train:', r2_score(y_train, y_train_predict))
#mae
print('Sai số tuyệt đối trung bình trên tập train (USD):', mean_absolute_error(y_train, y_train_predict))
#mae
print('Phần trăm sai số tuyệt đối trung bình tập train:', mean_absolute_percentage_error(y_train, y_train_predict))
Độ phù hợp tập train: 0.9878578202905779
Sai số tuyệt đối trung bình trên tập train (USD): 27.76585087158203
Phần trăm sai số tuyệt đối trung bình tập train: 0.056643879059475606
```

Đảo ngược chuẩn hóa: Sử dụng `sc.inverse_transform` để chuyển giá trị dự đoán về dạng ban đầu.

	Lần cuối	Dự đoán
Ngày		
1974-04-01	169.55	160.024399
1974-05-01	157.25	175.080933
1974-06-01	144.55	181.377197
1974-07-01	156.50	178.754639
1974-08-01	156.30	175.413940
...
2015-07-01	1095.40	1177.382080
2015-08-01	1134.10	1148.094849
2015-09-01	1114.60	1129.392700
2015-10-01	1142.00	1114.638184
2015-11-01	1064.02	1113.686523
500 rows x 2 columns		

Hình 4-42: Dữ liệu giá dự báo huấn luyện

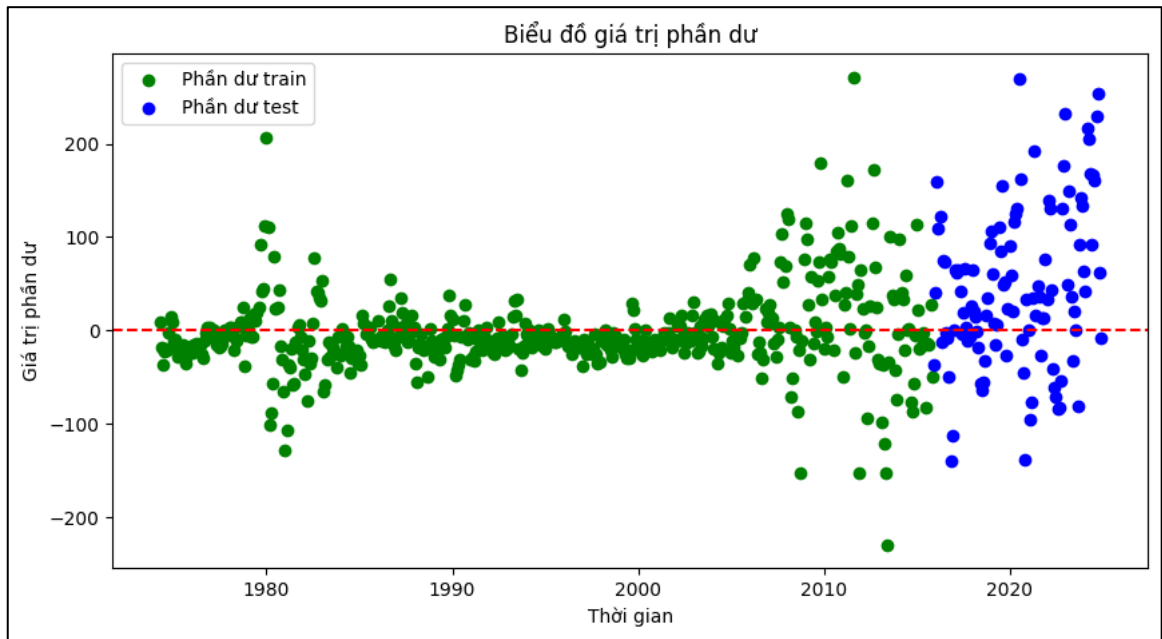
```
#r2
print('Độ phù hợp tập test:', r2_score(y_test, y_test_predict))
#mae
print('Sai số tuyệt đối trung bình trên tập test
(USD):', mean_absolute_error(y_test, y_test_predict))
#mae
print('Phần trăm sai số tuyệt đối trung bình tập
test:', mean_absolute_percentage_error(y_test, y_test_predict))
Độ phù hợp tập test: 0.937552476855157
Sai số tuyệt đối trung bình trên tập test (USD): 74.34689874211584
Phần trăm sai số tuyệt đối trung bình tập test: 0.043636437498968124
```

	Lần cuối	Dự đoán
Ngày		
2015-12-01	1060.91	1097.031738
2016-01-01	1117.64	1077.758179
2016-02-01	1237.76	1078.576294
2016-03-01	1231.95	1122.057495
2016-04-01	1293.36	1170.739258
...
2024-08-01	2503.03	2342.240234
2024-09-01	2634.49	2404.844971
2024-10-01	2743.80	2490.280518
2024-11-01	2653.55	2591.175049
2024-12-01	2623.81	2632.364258
109 rows × 2 columns		

Hình 4-43: Dữ liệu giá dự báo kiểm tra

```
# Tính giá trị phần dư cho tập train
train_data1['Phần dư'] = train_data1['Lần cuối'] - train_data1['Dự
đoán']

# Tính giá trị phần dư cho tập test
test_data1['Phần dư'] = test_data1['Lần cuối'] - test_data1['Dự đoán']
plt.figure(figsize=(10, 5))
plt.scatter(train_data1.index, train_data1['Phần dư'], label='Phần dư
train', color='green')
plt.scatter(test_data1.index, test_data1['Phần dư'], label='Phần dư
test', color='blue')
plt.axhline(y=0, color='red', linestyle='--') # Đường 0 để dễ quan sát
plt.xlabel('Thời gian')
plt.ylabel('Giá trị phần dư')
plt.title('Biểu đồ giá trị phần dư')
plt.legend()
plt.show()
```



Hình 4-44: Biểu đồ giá trị phần dư của tập huấn luyện và tập kiểm tra

```
# Lấy giá trị phần dư từ test_data1
residuals = test_data1['Phần dư'].values

# Tính toán RMSE
rmse = np.sqrt(np.mean(residuals**2))

# In ra kết quả
print('RMSE trên tập kiểm tra:', rmse)

RMSE trên tập kiểm tra: 96.92349972528075
```

4.2.5. Dự đoán giá tương lai

```
def predict_multiple_months(model, sc, data, num_months):
    predictions = []
    current_data = data[-50:].reshape(-1, 1)

    for _ in range(num_months):
        current_data_normalized = sc.transform(current_data)
        x_input = current_data_normalized[-50:].reshape(1, 50, 1)
        y_pred = model.predict(x_input)
        y_pred_inverse = sc.inverse_transform(y_pred)
        predictions.append(y_pred_inverse[0][0])
        current_data = np.append(current_data, y_pred_inverse)
        current_data = current_data.reshape(-1, 1)
    last_date = pd.to_datetime(df['Ngày'].iloc[-1])
```

```

    future_dates = pd.date_range(last_date + pd.DateOffset(months=1),
periods=num_months, freq='MS')
    result_df = pd.DataFrame({'Ngày': future_dates, 'Dự đoán':
predictions})

    return result_df

future_predictions = predict_multiple_months(final_model, sc, data,
num_months=12)
future_predictions

```

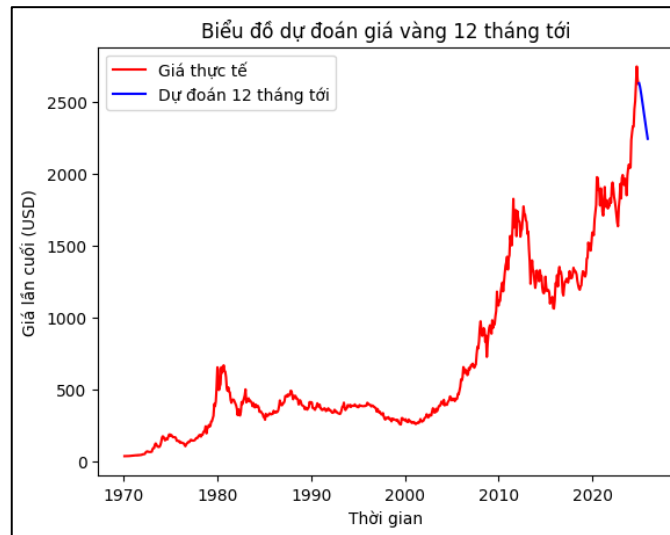
	Ngày	Dự đoán
0	2025-01-01	2593.853271
1	2025-02-01	2560.121582
2	2025-03-01	2512.789062
3	2025-04-01	2458.811035
4	2025-05-01	2401.749756
5	2025-06-01	2343.528076
6	2025-07-01	2285.264160
7	2025-08-01	2227.661377
8	2025-09-01	2171.188232
9	2025-10-01	2116.166992
10	2025-11-01	2062.815674
11	2025-12-01	2011.274170

Hình 4-45: Dữ liệu dự đoán giá XAU/USD trong 12 tháng

```

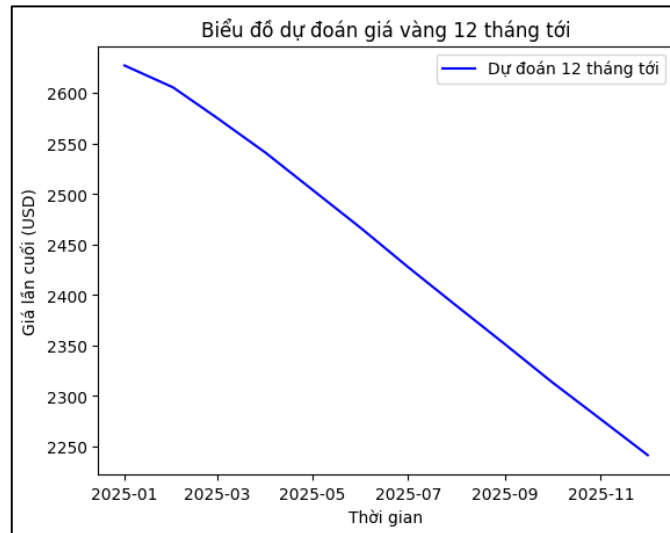
plt.plot(df['Ngày'], df['Lần cuối'], label='Giá thực tế', color='red')
plt.plot(future_predictions['Ngày'], future_predictions['Dự đoán'],
label='Dự đoán 12 tháng tới', color='blue')
plt.xlabel('Thời gian')
plt.ylabel('Giá lần cuối (USD)')
plt.title('Biểu đồ dự đoán giá vàng 12 tháng tới')
plt.legend()
plt.show()

```



Hình 4-46: Biểu đồ dự đoán giá XAU/USD trong 12 tháng

```
plt.plot(future_predictions['Ngày'], future_predictions['Dự đoán'],
label='Dự đoán 12 tháng tới', color='blue')
plt.xlabel('Thời gian')
plt.ylabel('Giá lần cuối (USD)')
plt.title('Biểu đồ dự đoán giá vàng 12 tháng tới')
plt.legend()
plt.show()
```

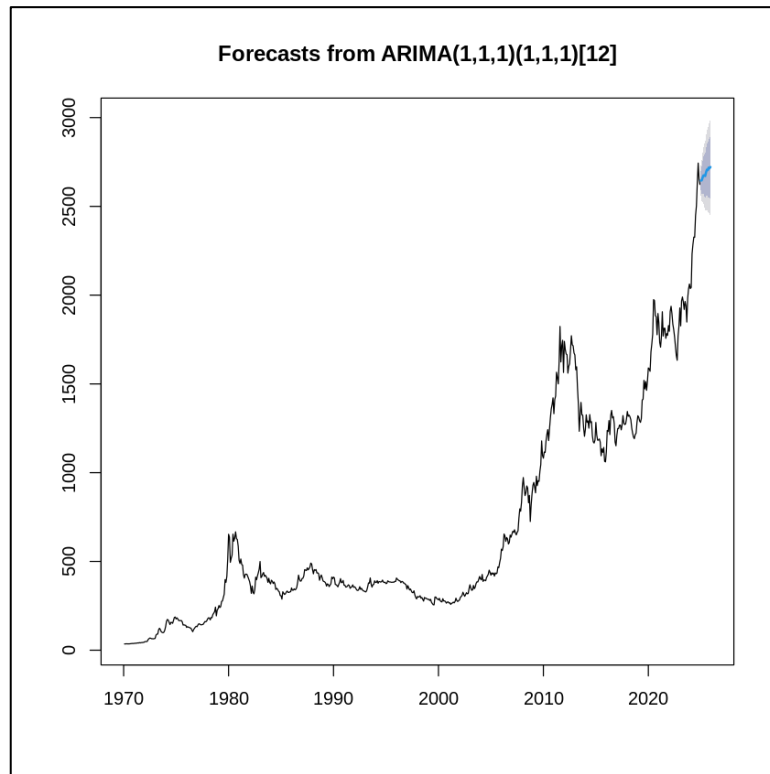


Hình 4-47: Biểu đồ dự đoán giá vàng 12 tháng tới

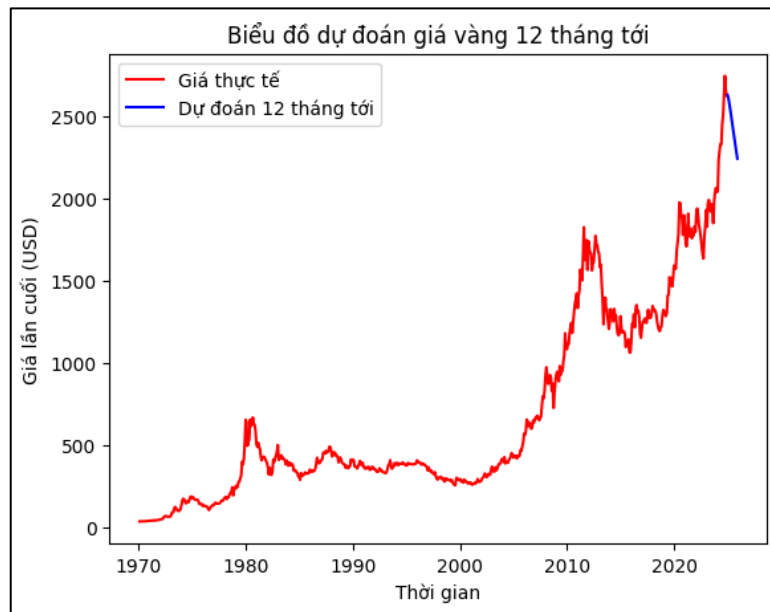
4.3. So sánh dự đoán của 2 mô hình ARIMA và LSTM

DỰ ĐOÁN GIÁ XAU/USD TRONG 12 THÁNG TỚI		
Ngày	ARIMA	LSTM
1/1/2025	2645.575	2627.252441
1/2/2025	2646.553	2605.559326
1/3/2025	2661.338	2575.418457
1/4/2025	2674.264	2540.921387
1/5/2025	2675.443	2504.111816
1/6/2025	2670.846	2466.114990
1/7/2025	2691.824	2427.632324
1/8/2025	2706.929	2389.151611
1/9/2025	2704.224	2351.035400
1/10/2025	2718.128	2313.561279
1/11/2025	2714.342	2276.939453
1/12/2025	2721.700	2241.325195

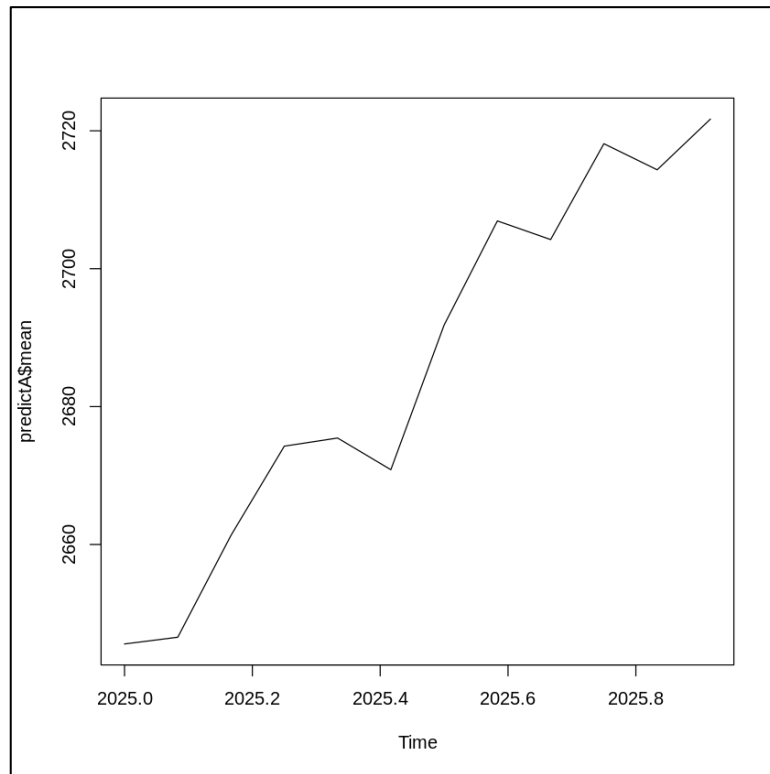
Bảng 4-1: Dữ liệu dự đoán của hai mô hình ARIMA và LSTM



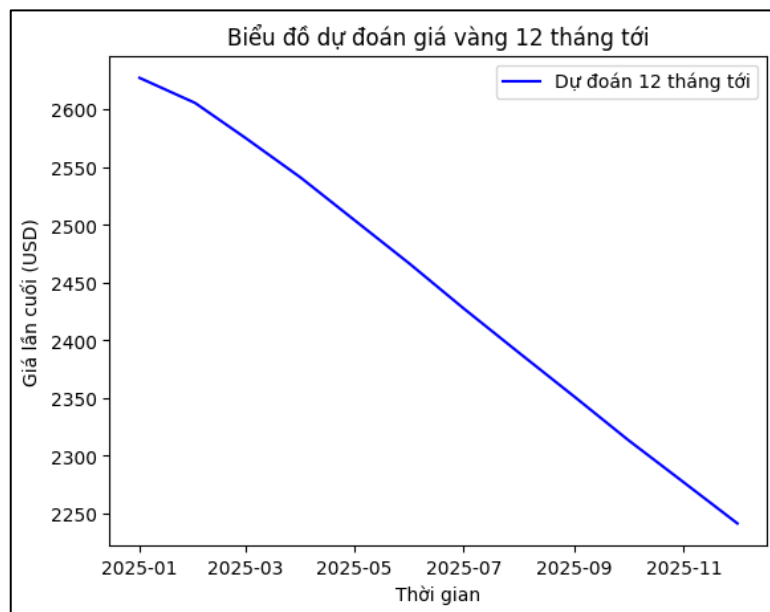
Hình 4-48: Biểu đồ xu hướng dự đoán giá XAU/USD bằng mô hình ARIMA



Hình 4-49: Biểu đồ xu hướng dự đoán giá XAU/USD bằng mô hình LSTM



Hình 4-50: Biểu đồ dự đoán giá XAU/USD bằng mô hình ARIMA



Hình 4-51; Biểu đồ dự đoán giá XAU/USD bằng mô hình LSTM

Xu hướng dự đoán của ARIMA:

ARIMA dự báo giá vàng (XAU/USD) sẽ tăng dần qua 12 tháng trong năm 2025. Giá bắt đầu từ 2645.575 (1/2025) và kết thúc ở mức 2721.7 (12/2025). Mức và xu hướng

tăng tương đối đồng đều và ổn định, với biên độ tăng trung bình mỗi tháng khoảng 6 - 20 USD. Biên độ biến động của ARIMA còn khá nhỏ (chênh lệch khoảng 76 USD giữa giá thấp nhất và cao nhất).

Xu hướng dự đoán của LSTM:

Mô hình LSTM lại dự báo giá vàng sẽ giảm liên tục qua các tháng. Giá bắt đầu ở mức 2627.252 (1/2025) và giảm xuống 2241.325 (12/2025). Mức độ và xu hướng giảm khá rõ rệt, với mức giảm trung bình khoảng 30 - 40 USD mỗi tháng. Biên độ biến động của LSTM lớn hơn nhiều so với ARIMA (chênh lệch khoảng 386 USD giữa giá thấp nhất và cao nhất), cho thấy mô hình này có thể nhạy cảm hơn với các thay đổi trong dữ liệu.

So sánh giữa hai mô hình:

Khác biệt về xu hướng của dữ liệu dự báo:

ARIMA và LSTM đưa ra dự đoán với xu hướng hoàn toàn trái ngược: ARIMA dự báo tăng giá, trong khi LSTM dự báo giảm giá cho thấy hai mô hình có cách tiếp cận và xử lý dữ liệu khác nhau. ARIMA dựa vào các yếu tố thời gian và xu hướng dài hạn, trong khi LSTM có khả năng học các mẫu phức tạp trong dữ liệu nhưng có thể bị ảnh hưởng bởi những biến động ngắn hạn.

Trong những tháng đầu năm (1/2025), dự đoán của cả hai mô hình tương đối gần nhau, chỉ chênh lệch khoảng 18 USD. Tuy nhiên, sự khác biệt càng lớn hơn trong các tháng về sau.

CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ

5.1. Kết luận

Về tổng quan, mô hình ARIMA đang chiếm ưu thế hơn hẳn so với LSTM khi kết quả dự đoán có sự biến động nhất định trong khi LSTM lại cho kết quả dự đoán quá một chiều. cả hai đều có điểm chung tích cực là dự đoán gần đúng với dữ liệu thực tế đầu tháng 1/2025.

Về mục tiêu, mục tiêu ban đầu của đề án là dùng hai mô hình ARIMA và LSTM để dự đoán giá XAU/USD và chọn ra mô hình tối ưu nhất. Vì vậy, theo mục tiêu được đề ra, ta có thể kết luận ARIMA sẽ là mô hình tốt và tối ưu hơn để có thể dự báo cho dữ liệu giá vàng theo đô la Mỹ này.

5.2. Kiến nghị

Đương nhiên giá vàng theo giá đô la Mỹ này sẽ bị tác động rất nhiều từ các yếu tố bên ngoài bao gồm các yếu tố kinh tế, thị trường, chính sách tiền tệ, ... Vì vậy các kiến nghị sẽ không xét đến các yếu tố khách quan mà sẽ chỉ dựa vào kết quả dự đoán từ mô hình ARIMA và LSTM để đưa ra các kiến nghị chi tiết cho nhà đầu tư.

5.2.1. Các nhà đầu tư ngắn hạn (khoảng 1-6 tháng)

5.2.1.1 ARIMA:

ARIMA dự đoán giá vàng tăng nhẹ từ tháng 1 đến tháng 6/2025, với mức tăng từ 2645.575 lên 2670.846.

Kiến nghị:

Nhà đầu tư ngắn hạn có thể cân nhắc mua vào ở thời điểm đầu năm 2025 (tháng 1-2) và bán ra sau 3-6 tháng để hưởng lợi từ xu hướng tăng giá nhỏ. Tuy nhiên, do mức tăng giá không lớn, nên cần tính toán kỹ chi phí giao dịch để đảm bảo lợi nhuận thực.

5.2.1.2 LSTM:

LSTM dự đoán giá vàng giảm đáng kể trong cùng giai đoạn, từ 2627.252 (tháng 1/2025) xuống 2466.114 (tháng 6/2025).

Kiến nghị:

Nhà đầu tư ngắn hạn nên hạn chế mua vào trong giai đoạn này, do giá có xu hướng giảm liên tục. Hơn nữa, có thể xem xét cân nhắc bán ra sớm nếu đang sở hữu vàng để tránh thua lỗ.

5.2.2. Nhà đầu tư dài hạn (6-12 tháng)

5.2.2.1 ARIMA:

ARIMA dự đoán giá vàng tiếp tục tăng ổn định từ tháng 7 đến tháng 12/2025, với mức tăng từ 2691.824 (tháng 7/2025) lên 2721.700 (tháng 12/2025).

Kiến nghị:

Nhà đầu tư dài hạn có thể cân nhắc mua vào trong giai đoạn giữa năm (tháng 6-7/2025), khi giá đã tăng lên mức ổn định nhưng chưa đạt đỉnh dự đoán. Nắm giữ lâu dài vì xu hướng tăng nhẹ sẽ đem lại lợi nhuận ổn định.

5.2.2.2 LSTM:

LSTM dự đoán giá vàng giảm mạnh trong giai đoạn này, từ 2427.632 (tháng 7/2025) xuống 2241.325 (tháng 12/2025).

Kiến nghị:

Nhà đầu tư dài hạn nên tránh mua vào trong năm 2025 vì giá có xu hướng giảm liên tục, dẫn đến rủi ro thua lỗ lớn. Nếu đã sở hữu vàng, cân nhắc bán ra trong các tháng đầu năm để tránh mức giảm sâu hơn vào cuối năm.

5.2.3. Nhà đầu tư trung tính (kết hợp cả hai mô hình)

Kết hợp cả hai dự đoán của ARIMA và LSTM để xây dựng chiến lược linh hoạt: Đặt kết quả tăng giá (ARIMA) làm mục tiêu và kết quả giảm giá (LSTM) làm mức dừng lỗ rủi ro. Chỉ mua vào khi giá thực tế vượt qua mức dự đoán giảm của LSTM, để giảm thiểu rủi ro.

TÀI LIỆU THAM KHẢO

- [1] *Cardoso, J. R., Pereira, L. M., Iversen, M. D., & Ramos, A. L. (2014). What is gold standard and what is ground truth? Dental Press Journal of Orthodontics, 19(5), 27–30. <https://doi.org/10.1590/2176-9451.19.5.027-030.ebo>*
- [2] *Cordero, C. A., Danao, B. M., Sy, L. A., & Asian Institute of Management. (n.d.). Predicting the XAU-USD Foreign Exchange Prices using Machine Learning [Journal-article].*
- [3] *Dahl, G. E., Yu, N. D., Deng, N. L., & Acero, A. (2011). Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Transactions on Audio Speech and Language Processing, 20(1), 30–42. <https://doi.org/10.1109/tasl.2011.2134090>*
- [4] *Dung, T. N. T. (2017). DỰ BÁO TIÊU THỤ DẦU MỎ CỦA MỸ BẰNG MÔ HÌNH ARIMA. http://lrc.quangbinhuni.edu.vn:8181/dspace/handle/DHQB_123456789/3654*
- [5] *Econometrics Academy - Time Series ARIMA Models. (n.d.). <https://sites.google.com/site/econometricsacademy/econometrics-models/time-series-arima-models>*
- [6] *Helleiner, E. (2008). Political determinants of international currencies: What future for the US dollar? Review of International Political Economy, 15(3), 354–378. <https://doi.org/10.1080/09692290801928731>*
- [7] *Ho, S., chreiter, Fakult• at f• ur Informatik, Schmidhub, J., er, & IDSIA. (1997). LONG SHORT-TERM MEMORY. Neural Computation, 9(8), 1735–1780. <https://www.bioinf.jku.at/publications/older/2604.pdf>*
- [8] *Hochreiter, S., PhD, Schmidhuber, J., Ronald Williams, Fakultät für Informatik, Technische Universität München, & IDSIA. (1997). Long Short-Term Memory. In*

- Massachusetts Institute of Technology, Neural Computation (Vols. 9–9, pp. 1735–1780). <https://deeplearning.cs.cmu.edu/F23/document/readings/LSTM.pdf>*
- [9] *Lượng, N. V., & Nguyễn, H. T. (2017). ỨNG DỤNG MÔ HÌNH ARIMA ĐỂ DỰ BÁO CHỈ SỐ GIÁ TIÊU DÙNG Ở VIỆT NAM. http://lrc.quangbinhuni.edu.vn:8181/dspace/handle/DHQB_123456789/2829*
- [10] *Newbold, P. (1983). ARIMA model building and the time series analysis approach to forecasting. Journal of Forecasting, 2(1), 23–35. <https://doi.org/10.1002/for.3980020104>*
- [11] *Quý, T. Q., Hải, N. V., Ninh, H. V., & Thúy, N. T. (2024). SỬ DỤNG MÔ HÌNH HỌC SÂU LSTM TRONG DỰ ĐOÁN GIÁ TRỊ CỔ PHIẾU. Quý | TNU Journal of Science and Technology. <https://doi.org/10.34238/tnu-jst.11554>*
- [12] *Quỳnh, G. T. Đ. N. (2015). Ứng dụng mô hình ARIMA để dự báo giá vàng Việt Nam trong ngắn hạn. http://tainguyenso.hce.edu.vn:8080/dspace/handle/TVDHKTH_123456789/3554*
- [13] *Stock Price Prediction Using the ARIMA Model. (2014, March 1). IEEE Conference Publication / IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/7046047>*
- [14] *Trịnh, N. P., Trần, N. a. K., Hà, V. L., Trần, T. H., & Đỗ, T. H. (2022, July 1). Dự đoán giá cổ phiếu sử dụng mô hình chuỗi thời gian với BigDL. <https://elib.vku.udn.vn/handle/123456789/2353>*
- [15] *XAU USD Historical Data - Investing.com. (2001, March 1). Investing.com. <https://www.investing.com/currencies/xau-usd-historical-data>*