

Tóm tắt nội dung—Trong nhiều thập kỷ gần đây, kim loại quý đã và đang duy trì được sức hút của mình trong giới đầu tư, đặc biệt là vàng, bạch kim và bạc. Cùng với đó, những bài toán và mô hình dự đoán giá kim loại không ngừng được sinh ra và cải tiến với độ chính xác ngày một cao hơn. Tuy vậy, tùy theo bối cảnh kinh tế thế giới và sự tiến bộ của các kỹ thuật phân tích dữ liệu mà những mô hình dự đoán sẽ cho ra những kết quả khác nhau. Để giải quyết vấn đề này, nhóm chúng em sẽ thực hiện xây dựng và đánh giá song song các mô hình dự báo giá theo chuỗi thời gian với 10 thuật toán sau: Linear Regression, ARIMA, RNN, GRU, LSTM, FFT, TBATS, SES, N-HiTS, PatchTST; cùng với đó là 3 phương pháp kiểm định mô hình: MAPE, RMSE, MAE.

Index Terms—Linear Regression – Hồi quy tuyến tính, ARIMA – Autoregressive Integrated Moving Average, RNN – Recurrent Neural Network, GRU – Gated Recurrent Unit, LSTM – Long Short Term Memory, FFT – Fast Fourier Transform, TBATS – (Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend, Seasonal components), SES – Simple Exponential Smoothing, N-HiTS – Neural Hierarchical Interpolation for Time Series, PatchTST – Patch Time Series Transformer, MAPE – Mean Absolute Percentage Error, RMSE – Root Mean Squared Error, MAE – Mean Absolute Error.

I. GIỚI THIỆU CHUNG

Sự biến động của giá vàng(Au), bạc(Ag) và bạch kim(Pt) luôn là điểm nóng trong lĩnh vực đầu tư và tài chính. Việc dự đoán giá của các loại kim loại quý này không chỉ có ý nghĩa quan trọng trong việc hỗ trợ quyết định đầu tư của các nhà đầu tư và phân tích thị trường, mà còn trong việc cung cấp thông tin chi tiết và chính xác về biến động giá, giúp cải thiện đáng kể hiệu suất kinh doanh và quản lý rủi ro trong các lĩnh vực liên quan như chứng khoán và tiền tệ. Dự đoán giá kim loại quý như vàng không chỉ tập trung vào dự đoán giá toàn cầu của 1 ngày hôm sau mà còn dự đoán giá trong tương lai xa hơn như 3 đến 5 tháng sau, dự đoán khoảng biến động giá, xu hướng biến động, mối liên hệ của giá các kim loại,... Do đó, nhiều trang điện tử dự báo chuyên nghiệp đã được thiết lập như Bloomberg, Kitco, LBmMA,... Bằng cách so sánh và kiểm định nhiều phương pháp và mô hình dự báo từ kinh điển cho đến mới nhất, nhóm nghiên cứu kỳ vọng sẽ tìm ra được mô hình, thuật toán với kết quả khả quan nhất nhằm linh hoạt đáp ứng được nhu cầu ngày một cao của thị trường. Bài nghiên cứu của chúng em sử dụng bộ dữ liệu từ trang thông tin điện tử MetalPriceAPI để lấy giá vàng, bạch kim, bạc được ghi nhận trong quá khứ (lấy theo mệnh giá USD).

II. CÁC NGHIÊN CỨU GẦN ĐÂY

A. Nghiên cứu của nhóm tác giả Tawum Juvert Mbah, Haiwang Ye, Jianhua Zhang Mei Long

Nghiên cứu này đã chọn ra sáu yếu tố ảnh hưởng đến thị trường đá vôi để mô phỏng và dự đoán xu hướng giá trong tương lai, bao gồm xi măng, vàng, than đá, năng lượng, lãi suất và giá đá vôi. Nghiên cứu sử dụng hai mô hình mạng nơ-ron sâu tiên tiến là RNN và ARIMA để mô phỏng và dự đoán giá đá vôi. Kết quả cho thấy, mô hình ARIMA đã có dự đoán tốt hơn so với mô hình RNN về xu hướng và biến động giá của đá vôi. Sự khác biệt chính giữa hai mô hình này nằm ở việc mô hình ARIMA có khả năng tạo ra kết quả chính xác hơn và thời gian huấn luyện ít hơn so với mô hình RNN. Do đó, mạng ARIMA

được chứng minh là một phương pháp tính vi và hiệu quả trong việc mô hình, phân tích và dự đoán giá của thị trường đá vôi.

B. Nghiên cứu của Bojun Yin, Renguang Zuo, Yihui Xiong

Các nhà nghiên cứu sử dụng mô hình GRU để tạo bản đồ tiềm năng khoáng sản (Mineral Prospectivity Mapping - MPM) , sử dụng dữ liệu từ quận Baguio, Philipines. Các kết quả thu được nhấn mạnh tính hiệu quả của mô hình GRU trong MPM. Các khu vực cao điểm được phân biệt thể hiện mối quan hệ không gian chặt chẽ với các mỏ khoáng sản đã biết, cung cấp thông tin quan trọng cho các hoạt động khai thác khoáng sản tiếp theo trong khu vực nghiên cứu.

C. Nghiên cứu của F.Javier Galán-Sales, Pablo Reina-Jiménez, Manuel Carranza-García, José María Luna-Romera

Nghiên cứu tiềm năng của việc sử dụng FFT như một công cụ biến đổi đặc trưng để cải thiện độ chính xác và hiệu quả của các mô hình dự báo chuỗi thời gian. Kết quả của nhóm nghiên cứu cho thấy rằng việc sử dụng FFT như một công cụ biến đổi đặc trưng vượt trội hơn các phương pháp biến đổi đặc trưng truyền thống trong việc dự báo độ chính xác và hiệu quả tính toán.

III. TÀI NGUYÊN

A. Các tập dữ liệu sử dụng

Bộ ba dataset thể hiện giá ba loại kim loại quý: vàng, bạc và bạch kim trong khoảng thời gian từ 1/1/2018 đến 1/3/2024 được lấy nguồn từ các API do <https://metalpriceapi.com/> cung cấp.

Sau khi gọi API để lấy dữ liệu, nhóm tiến hành chuyển đổi từ dạng json thành csv và thu được 3 file csv gồm:

- Giá vàng: gold_price_2018_2024.csv
- Giá bạc: silver_price_2018_2024.csv
- Giá bạch kim: platinum_price_2018_2024.csv

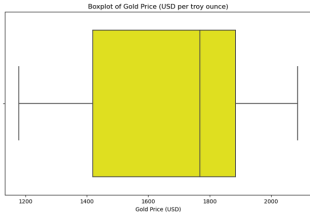
Trong mỗi tập dữ liệu gồm hai cột:

- Date:** ngày thu nhập dữ liệu (định dạng YYYY-MM-DD).
- Value (USD per troy ounce):** giá của kim loại quý tương ứng với cột Date (mệnh giá USD).

B. Thống kê mô tả

Bảng I
THỐNG KÊ MÔ TẢ GIÁ VÀNG, GIÁ BẠC VÀ GIÁ BẠCH KIM

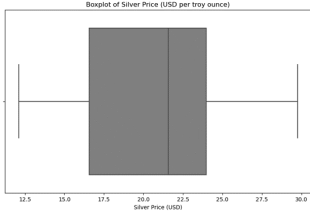
	Giá vàng	Giá bạc	Giá bạch kim
red!20red!20Count	2252	2251	2252
red!20Mean	1673.567	20.531	940.485
red!20Std	262.337	4.131	106.855
red!20Min	1178.57	12.112	591.46
red!2025%	1419.73	16.572	864.475
red!2050%	1768.317	21.584	930.843
red!2075%	1884.517	23.975	999.513
red!20Max	2085.54	29.748	1306.684



Hình 1. Biểu đồ hộp của giá vàng



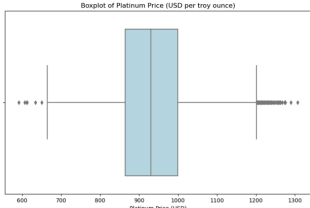
Hình 2. Biểu đồ đường của giá vàng



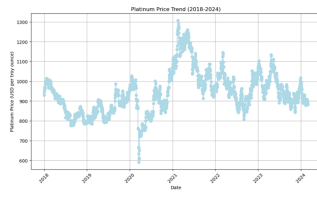
Hình 3. Biểu đồ hộp của giá bạc



Hình 4. Biểu đồ đường của giá bạc



Hình 5. Biểu đồ hộp của giá bạch kim



Hình 6. Biểu đồ đường của giá bạch kim

C. Công cụ sử dụng

Trong quá trình nghiên cứu và phân tích dữ liệu để dự báo giá khoáng sản, nhóm đã sử dụng một số công cụ và thư viện phần mềm phổ biến, chủ yếu được triển khai bằng ngôn ngữ lập trình Python. Danh sách các công cụ và thư viện chính được sử dụng: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn,...

IV. PHƯƠNG PHÁP

A. Linear Regression

Hồi quy tuyến tính (Linear Regression) được giới thiệu lần đầu bởi nhà thống kê Sir Francis Galton vào cuối thế kỷ 19, là một phương pháp thống kê dùng để mô hình hóa mối liên hệ giữa một biến phụ thuộc và một (hoặc nhiều) biến độc lập. Hồi quy tuyến tính đơn thì có 1 biến độc lập, hồi quy tuyến tính bội thì có nhiều hơn 1 biến độc lập.

Nó xây dựng một mô hình tuyến tính bằng cách tìm một đường thẳng (nếu là hồi quy đơn biến) hoặc một siêu phẳng (nếu là hồi quy đa biến) dự đoán tốt nhất dựa trên quan hệ tuyến tính sao cho khác biệt giữa giá trị dự đoán và giá trị thực tế là nhỏ nhất. Hồi quy tuyến tính được ứng dụng rộng rãi để dự đoán giá trị đầu ra, xem xét ảnh hưởng của các biến số trong các lĩnh vực như kinh tế, tài chính, môi trường,...

Phương trình hồi quy tuyến tính đơn biến:

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$

Phương trình hồi quy tuyến tính bội với k biến độc lập:

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k + \varepsilon$$

Trong đó:

- Y là giá trị dự đoán của biến phụ thuộc (Y) ứng với giá trị các biến độc lập X_1, X_2, \dots, X_k
- β_0 là hệ số chặn, là giá trị dự đoán của Y khi giá trị của các biến độc lập bằng 0
- β_1, \dots, β_k là hệ số tương quan hồi quy – cho thấy mức độ mạnh yếu trong mối liên hệ của Y với các biến độc lập.
- X_1, X_2, \dots, X_k là giá trị các biến độc lập để dự đoán Y.
- ε là sai số của dự đoán, là phần biến thiên không thể giải thích bởi mối liên hệ tuyến tính của X và Y.

B. ARIMA

ARIMA, hay mô hình tự hồi quy tích hợp trung bình chuyển động, là một phương pháp phân tích thống kê sử dụng dữ liệu chuỗi thời gian để hiểu tốt hơn một tập dữ liệu hoặc dự đoán xu hướng tương lai. Nó là một mô hình kết hợp của 2 quá trình tự hồi quy và một trung bình trượt. Dữ liệu quá khứ sẽ được sử dụng để dự đoán dữ liệu trong tương lai.

Mô hình ARIMA đại diện cho tự hồi quy tự động (AR), trung bình di chuyển (MA) và tích hợp sai phân (I).

• **Tích hợp (I) d:** Là quá trình sai phân tích hợp, dùng để so sánh sự khác biệt giữa d quan sát.

• **Tự hồi quy (AR) p:** Đây là thành phần tự hồi quy bao gồm tập hợp các trễ của biến hiện tại. Độ trễ bậc p là giá trị quay lại thời điểm p trong quá khứ của chuỗi. Quá trình AR có độ trễ dài hay ngắn phụ thuộc vào tham số trễ p.

• **Trung bình di chuyển (MA) q:** Quá trình trung bình di chuyển (MA) q được hiểu là quá trình thay đổi giá trị trung bình của chuỗi theo thời gian. Quá trình trung bình di chuyển sẽ tìm ra mối liên hệ tuyến tính giữa dữ liệu hiện tại và q lỗi trước đó.

Phương trình hồi quy ARIMA (p, d, q) có thể được biểu diễn như sau: Phương trình hồi quy tuyến tính đơn giản:

$$\Delta x_t = \sum_{i=1}^p \alpha_i \Delta(x_{t-i}) + \sum_{i=1}^q \beta_i \varepsilon_{t-i}$$

C. RNN

Mạng hồi quy RNN (Recurrent Neural Network) là một loại Mạng nơ ron (Neural Network) thường được sử dụng cho dữ liệu tuần tự như chuỗi thời gian. RNN sử dụng đầu ra của bước trước làm đầu vào cho bước hiện tại trong khi Mạng Nơ ron truyền thống tất cả đầu vào và đầu ra đều độc lập với nhau.

Đặc điểm chính và quan trọng nhất của RNN là trạng thái ẩn (Hidden state). Trạng thái này nhớ đầu vào trước đó của mạng. Nó sử dụng cùng trọng số cho đầu vào của mỗi lớp trong mạng.

RNN tính toán trạng thái ẩn ký hiệu là h_t dựa trên đầu vào hiện tại x_t và trạng thái ẩn trước đó h_{t-1} :

$$h_t = \sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

Đầu ra y_t của RNN được tính toán dựa trên trạng thái ẩn hiện tại:

$$y_t = \sigma_y(W_{hy}h_t + b_o)$$

Trong đó:

- σ_h, σ_y : Hàm kích hoạt của lớp ẩn và lớp đầu ra.
- W_{xh} : Là ma trận trọng số kết nối giữa lớp đầu vào và lớp ẩn.
- W_{hh} : Là ma trận trọng số kết nối giữa lớp ẩn và chính nó.
- W_{hy} : Là ma trận trọng số kết nối giữa lớp ẩn và lớp đầu ra.

Các tham số này được cập nhật bằng cách sử dụng Lan truyền ngược (Backpropagation). Tuy nhiên, do RNN hoạt động trên dữ liệu tuần tự nên nó sẽ sử dụng Backpropagation through Time (BPTT).

Khác với lan truyền ngược truyền thống chỉ tính toán lỗi cho bước đầu ra cuối cùng và cập nhật tham số của mỗi lớp độc lập, BPTT tính toán lỗi cho mỗi bước thời gian trong chuỗi đầu vào, cộng dồn chúng lại và cập nhật tham số của toàn bộ mạng dựa trên lỗi tổng hợp.

D. SES

Mô hình Smoothing Exponential đơn giản (SES) là một phương pháp dự báo thời gian đơn giản và hiệu quả được sử dụng để làm mịn dữ liệu chuỗi thời gian và dự đoán giá trị trong tương lai. SES đặc biệt hữu ích khi cần dự báo ngắn hạn trên dữ liệu không có xu hướng hoặc mùa vụ rõ rệt. Phương pháp này hoạt động bằng cách áp dụng trọng số giảm một cách mũ màu mực cho các quan sát trước đó, từ đó làm mịn các biến động ngẫu nhiên trong dữ liệu.

Công thức:

$$s_t = \alpha x_t + (1 - \alpha)s_{t-1} + \alpha(x_t - s_{t-1}) \quad (1)$$

- s_t : giá trị dự đoán
- s_{t-1} : giá trị dự đoán trước đó
- α : hệ số làm mịn ($0 < \alpha < 1$)
- t : thời điểm

Khi giá trị của α lớn (gần với 1), mức độ làm mịn giảm. Điều này có nghĩa là thuật toán SES sẽ có ít làm mịn hơn và phản

ứng mạnh mẽ hơn đối với những thay đổi gần đây nhất trong dữ liệu. Các giá trị dự đoán sẽ gần với các giá trị quan sát gần đây nhất.

E. GRU

Mạng Neural hồi tiếp nút có cổng (GRU) là một cơ chế cổng trong các mạng nơ ron hồi quy, là một biến thể đơn giản hơn của mạng Bộ nhớ ngắn hạn dài (LSTM). GRU có thể xử lý dữ liệu tuần tự như văn bản, bài viết và chuỗi thời gian. GRU sử dụng cơ chế cổng để cập nhật có chọn lọc trạng thái ẩn của mạng tại mỗi bước thời gian. Cơ chế cổng được dùng để kiểm soát luồng dữ liệu đi vào và đi ra của mạng. GRU có 2 cơ chế cổng: cổng xóa và cổng cập nhật.

Cổng xóa quyết định bao nhiêu phần mà trạng thái trước đây được giữ lại, còn cổng cập nhật quyết định trạng thái ẩn mới có bao nhiêu phần giống trạng thái ẩn cũ. Đầu ra của GRU là tính toán dựa trên số trạng thái ẩn được cập nhật, giúp mô hình có thể ‘nhớ’ thông tin quan trọng từ quá khứ và ‘nhìn’ vào thông tin mới để điều chỉnh trạng thái hiện tại.

toán cổng xóa, cổng cập nhật, và số trạng thái ẩn của GRU như sau:

Cổng xóa:

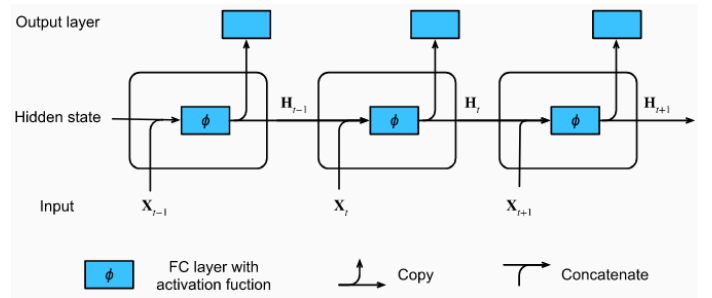
$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r)$$

Cổng cập nhật:

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)$$

Trong đó:

- X_t : đầu vào ở bước thời gian hiện tại.
- H_{t-1} : là trạng thái ẩn ở bước trước đó.
- $W_{xr}, W_{xz} \in \mathbb{R}^{d \times h}$ và $W_{hr}, W_{hz} \in \mathbb{R}^{h \times h}$: là các trọng số.
- $b_r, b_z \in \mathbb{R}^{1 \times h}$: là các tham số độ chênh.
- σ : là hàm sigmoid để 2 giá trị thu được thuộc khoảng (0, 1).



Hình 7. Mô hình GRU

F. FFT

Biến đổi Fourier nhanh (FFT) là một thuật toán cực kỳ hiệu quả để chuyển đổi một tín hiệu rời rạc miền thời gian sang miền tần số dựa trên biến đổi Fourier rời rạc (DFT). Phép biến đổi DFT phân tích một dãy các số thành các thành phần ở các tần số khác nhau. Phép biến đổi này giúp xác định các thành phần tần số chính của chuỗi thời gian từ đó dự báo các giá trị tương lai dựa trên các thành phần này.

Phép biến đổi Fourier rời rạc - DFT của một tín hiệu rời rạc cho trước $x_n = [x_0, x_1, \dots, x_{N-1}]$ được cho bởi biểu thức sau:

$$X_k = \begin{cases} \sum_{n=0}^{N-1} x(n) e^{-i \frac{2\pi}{N} kn} & \text{Nếu } 0 \leq k \leq N-1 \\ 0 & \text{Nếu } k \text{ thuộc phần còn lại} \end{cases}$$

trong đó: i là đơn vị phức.

Tính toán biến đổi DFT có độ phức tạp là $O(N^2)$.

FFT đã được ra đời để khắc phục tốc độ xử lý của DFT. Giả thiết dãy $x(n)$ có độ dài $N = 2^i$, nếu không có dạng lũy thừa 2 thì thêm vài mẫu 0 vào sau dãy $x(n)$. Nguyên tắc cơ bản mà các thuật toán FFT đều dựa vào là phân chia DFT N mẫu thành các DFT nhỏ hơn một cách liên tục.

Với $N = 2^i$, đầu tiên ta sẽ phân chia DFT N mẫu thành các DFT $N/2$ mẫu, sau đó phân chia DFT $N/2$ mẫu thành $N/4$ mẫu và cứ thế cho đến khi được DFT dài $N = 2$. Về bản chất nó là đệ quy. Do đó độ phức tạp của FFT được xác định là $O(N \log_2(N))$.

Thuật toán FFT:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-i \frac{2\pi}{N} (2n)k} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-i \frac{2\pi}{N} (2n+1)k}$$

G. TBATS

TBATS là một mô hình dự báo chuỗi thời gian nâng cao, cải tiến từ mô hình BATS, và được phát triển bởi Rob J. Hyndman và Nikolaos Kourentzes. Mô hình TBATS được tạo ra vào năm 2011 với mục đích để dự báo chuỗi thời gian với nhiều chu kỳ mùa vụ (chu kỳ theo giờ, ngày, quý, năm,...). TBATS có thể xử lý các dạng mùa vụ phức tạp như tính mùa vụ không phải số nguyên, không lồng nhau và mùa vụ có chu kỳ dài. TBATS thì ít phổ biến và ít được dùng hơn so với các mô hình khác trong phả hệ mô hình ARIMA. TBATS gồm 5 thành phần:

- **Trigonometric seasonality** (tính mùa vụ theo lượng giác): dùng phép biến đổi lượng giác (sin-cos) để mô hình các yếu tố mùa vụ có tần suất cố định.

- **Box-Cox transformation** (hàm biến đổi Box-Cox): xử lý tính không đồng nhất của dữ liệu, giúp cải thiện phân phối dữ liệu và làm cho phương sai của chúng đồng đều hơn.

- **ARMA errors** (sai số tự hồi quy kết hợp trung bình trượt): giúp dự đoán các sai số ngẫu nhiên không có tính mùa vụ.

- **Trend components** (xu hướng): thể hiện mô hình dự đoán xu hướng tăng, giảm.

- **Seasonal components** (các yếu tố mùa vụ): loại bỏ các biến động chu kỳ khỏi dữ liệu.

Công thức:

• Hàm biến đổi Box-Cox:

– Trường hợp tuyến tính:

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^{(\omega)} - 1}{\omega}, & \omega \neq 0 \\ \log y_t, & \omega = 0 \end{cases}$$

$$y_t^{(\omega)} = l_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t$$

– Trường hợp phi tuyến tính:

$$y_t^{(\omega)} = w' x_{t-1} + \varepsilon_t$$

$$x_t = F x_{t-1} + g \varepsilon_t$$

Trong đó, m_1, \dots, m_T nói về chu kỳ mùa, l_t là mức riêng của chu kỳ t , $s_t^{(i)}$ biểu diễn cho thành phần mùa vụ tại thời điểm t , w' là một dòng véc-tơ, g là một cột véc-tơ, F là một ma trận và x_t là véc-tơ trạng thái không quan sát được tại thời điểm t (tương lai).

• Hàm xử lý với phương pháp làm mịn – giảm dần xu hướng (trend damping), tìm các hệ số ARMA, mức riêng.

$$l_t = l_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$d_t = \sum_{i=1}^p \varphi_i d_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t$$

$$b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t$$

Trong đó, b_t là xu hướng ngắn hạn trong chu kỳ t , d_t nói về hệ số ARMA(p, q) và sai số ε_t lấy từ quá trình nhiễu trắng với trung bình bằng 0 và phương sai cố định σ^2 , α và β và γ_i ($i = 1, \dots, T$) là các hệ số làm mịn, θ_i là hệ số thành phần i của trung bình trượt (MA). Xu hướng làm mịn Gardener McKenzie với tham số ϕ được sử dụng theo hướng Snyder để tạo nên b_t .

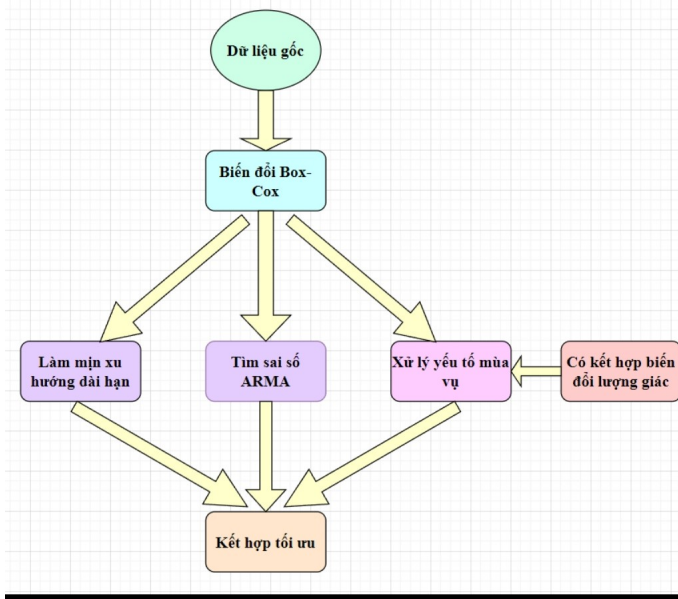
• Hàm biến đổi lượng giác:

$$s_t^{(i)} = \sum_{j=i}^{k_i} s_{j,t}^{(i)}$$

$$k_i = \begin{cases} \frac{m_i}{2} & \text{nếu giá trị của mùa } i \text{ là chẵn,} \\ \frac{(m_i-1)}{2} & \text{nếu giá trị của mùa } i \text{ là lẻ} \end{cases}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{*(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \\ \lambda_j^{(i)} &= 2\pi j / m_i \end{aligned}$$

Trong đó, γ_1 và γ_2 là tham số làm mịn, $s_{j,t}^{(i)}$ là mức độ tăng trưởng ngẫu nhiên trong khoảng thời gian $s_j^{*(i)}$.



Hình 8. Hình luồng hoạt động của mô hình TBATS

Lưu ý: để có được mô hình TBATS cuối cùng thì có thể thêm hoặc bỏ các phần xử lý: biến đổi Box-Cox, làm mịn xu hướng (Trend Damping), tính sai số ARMA nhằm tạo được mô hình tốt nhất.

H. LSTM

LSTM (Bộ nhớ dài hạn) là một loại mạng nơ-ron hồi quy (RNN) được thiết kế để giải quyết vấn đề gradient biến mất trong các RNN truyền thống. Lợi thế của LSTM là khả năng xử lý khoảng cách xa giữa các thông tin, làm cho nó trở nên hữu ích trong các ứng dụng như nhận dạng giọng nói, dịch máy và phân tích chuỗi thời gian.

Một đơn vị LSTM bao gồm một ô nhớ, một cổng đầu vào, một cổng đầu ra và một cổng quên. Ô nhớ lưu giữ thông tin trong thời gian dài, trong khi các cổng kiểm soát dòng chảy của thông tin vào và ra khỏi ô. Cổng quên quyết định thông tin nào cần bỏ qua, cổng đầu vào quyết định thông tin mới nào cần lưu trữ, và cổng đầu ra quyết định thông tin nào cần xuất ra, giúp mạng LSTM duy trì các phụ thuộc dài hạn hữu ích để đưa ra dự đoán. Cổng quên xóa thông tin không còn hữu ích khỏi ô nhớ. Phương trình của cổng quên là:

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

Trong đó:

- σ : Hàm kích hoạt Sigmoid
- W_f và b_f : Trọng số và độ lệch của cổng quên

Cổng đầu vào sẽ thêm thông tin hữu ích vào ô nhớ. Phương trình cho cổng đầu vào là:

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \end{aligned}$$

Trong đó:

- i_t : Đầu vào tại thời điểm t
- C_{t-1} : Ô nhớ tại thời điểm $t-1$
- W_c, b_c : Trọng số và độ lệch của ô nhớ
- \tanh : Hàm kích hoạt Tanh

Cổng đầu ra sẽ trích xuất thông tin hữu ích từ ô nhớ hiện tại để trình bày như đầu ra. Phương trình cho cổng đầu ra là:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

I. N-HiTS

Thuật toán N-HiTS (Neural Hierarchical Interpolation for Time Series) là một phương pháp tiên tiến cho dự báo chuỗi thời gian, được xây dựng trên nền tảng Phân Tích Mở Rộng Cơ Sở Thần Kinh (N-BEATS). N-HiTS nhằm cải thiện độ chính xác và hiệu quả tính toán, đặc biệt cho các dự báo dài hạn. Nó giới thiệu các cải tiến chính như lấy mẫu tín hiệu đa tỷ lệ và tổng hợp đa tỷ lệ theo thứ bậc để xây dựng các dự báo theo cách thứ bậc, giảm yêu cầu tính toán trong khi tăng độ chính xác.

1) **Lấy Mẫu Tín Hiệu Đa Tỷ Lệ:** Mỗi khối trong kiến trúc N-HiTS sử dụng một lớp MaxPool với kích thước nhân k_ℓ để tập trung vào các quy mô khác nhau của tín hiệu đầu vào. Lấy mẫu đa tỷ lệ này đảm bảo rằng mỗi khối phân tích các thành phần của đầu vào ở các tỷ lệ mẫu khác nhau, cho phép các khối với nhân lớn hơn tập trung vào các thành phần quy mô lớn hơn và các khối với nhân nhỏ hơn tập trung vào các chi tiết tinh tế hơn. Về mặt chính thức, với đầu vào $y_{t-L:t,\ell}$ cho khối ℓ , đầu ra sau khi pooling là:

$$y_{t-L:t,\ell}^{(p)} = \text{MaxPool}(y_{t-L:t,\ell}, k_\ell)$$

2) **Hồi Quy Phi Tuyến:** Sau khi lấy mẫu đa tỷ lệ, mỗi khối thực hiện hồi quy phi tuyến để tính toán các hệ số tiến θ_ℓ^f và lùi θ_ℓ^b cho nội suy. Các hệ số này được tạo ra bằng cách sử dụng một mạng perceptron nhiều lớp (MLP) xử lý đầu vào đã pooling:

$$h_\ell = \text{MLP}_\ell(y_{t-L:t,\ell}^{(p)})$$

$$\theta_\ell^f = \text{Linear}^f(h_\ell)$$

$$\theta_\ell^b = \text{Linear}^b(h_\ell)$$

3) **Nội Suy Theo Thứ Bậc:** Trong các mô hình dự báo đa chân trời điển hình, kích thước đầu ra trực tiếp tương ứng với chân trời dự báo H . Điều này thường dẫn đến chi phí tính toán cao và tăng độ phức tạp của mô hình. N-HiTS giảm thiểu điều này bằng cách sử dụng nội suy theo thứ bậc, nơi mà độ lớn của các hệ số dự báo θ_ℓ^f được kiểm soát bởi tỷ lệ biểu đạt r_ℓ như sau:

$$|\theta_\ell^f| = \lceil r_\ell H \rceil$$

Để khôi phục đầy đủ chân trời dự báo H , nội suy được áp dụng:

$$\hat{y}_{\tau,\ell} = g(\tau, \theta_\ell^f), \forall \tau \in \{t+1, \dots, t+H\}$$

$$\tilde{y}_{\tau,\ell} = g(\tau, \theta_\ell^b), \forall \tau \in \{t-L, \dots, t\}$$

Hàm nội suy g có thể thay đổi về độ mượt mà, với các lựa chọn như nội suy lân cận gần nhất, tuyến tính từng đoạn, và nội suy cubic.

4) **Cấu Trúc Khối và Chồng:** N-HiTS được cấu tạo từ nhiều khối được tổ chức thành các chồng, với mỗi chồng được thiết kế để chuyên biệt trong việc nắm bắt các đặc điểm khác nhau của dữ liệu chuỗi thời gian. Mỗi khối chứa một MLP tạo ra đầu ra dự báo $\hat{y}_{t+1:t+H,\ell}$ và đầu ra backcast $\tilde{y}_{t-L:t,\ell}$. Các khối được kết nối theo cách xếp chồng đôi dư, cho phép mỗi khối tinh chỉnh tín hiệu đầu vào cho các khối tiếp theo và dần dần cải thiện các dự báo.

Về mặt chính thức, đầu vào cho khối ℓ trong chồng s có thể được biểu diễn như sau:

$$y_{t-L:t,\ell} = y_{t-L:t} - \sum_{i=1}^{\ell-1} \tilde{y}_{t-L:t,i}$$

Đầu ra của mỗi khối sau đó được tổng hợp để tạo thành dự báo cuối cùng:

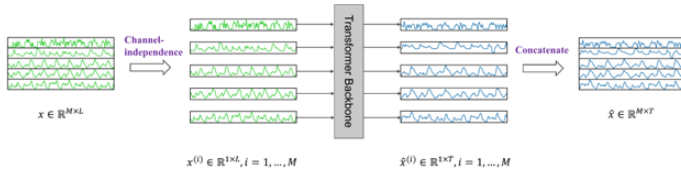
$$\hat{y}_{t+1:t+H} = \sum_{\ell=1}^B \hat{y}_{t+1:t+H,\ell}$$

J. PatchTST

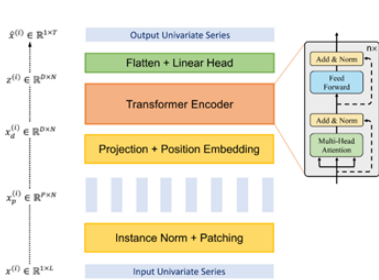
PatchTST là mô hình sử dụng Transformer để dự đoán đa biến và học biểu diễn tự giám sát. Dựa trên hai phần chính:

- 1) **Vá dữ liệu (Patching):** phân đoạn chuỗi thời gian thành các đoạn nhỏ cấp độ chuỗi con, được sử dụng làm các token đầu vào cho Transformer.
- 2) **Kênh truyền độc lập:** mỗi kênh chứa một chuỗi thời gian đơn biến duy nhất, sử dụng cùng trọng số Transformer và trọng số nhưng trên tất cả các chuỗi.

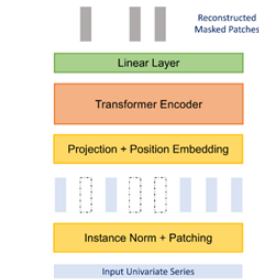
Kiến trúc mô hình PatchTST sử dụng bộ mã hóa Transformer gốc, mỗi chuỗi đơn biến có độ dài L được chia thành các mảnh, đưa vào Transformer một cách độc lập, và tạo ra T giá trị tương lai (x_{L+1}, \dots, x_{L+T}).



(a) PatchTST Model Overview



(b) Transformer Backbone (Supervised)



(c) Transformer Backbone (Self-supervised)

Hình 9. Mô hình PatchTST

Dữ liệu chuỗi thời gian đa biến $x \in \mathbb{R}^{M \times L}$ được chia thành các chuỗi đơn biến $x^{(i)} \in \mathbb{R}^{1 \times L}$ với $i = 1, \dots, M$, sau đó

được đưa vào Transformer Backbone để xử lý. Các quy trình tiến tới (forward process) là độc lập, và các dự đoán đầu ra $\hat{x}^{(i)} \in \mathbb{R}^{1 \times T}$, $i = 1, \dots, M$ được gộp lại để tạo ra đầu ra cuối cùng $\hat{x} \in \mathbb{R}^{M \times T}$.

PatchTST có hai biến thể:

- **PatchTST có giám sát:** Sử dụng dữ liệu có nhãn.
- **PatchTST tự giám sát:** Không yêu cầu dữ liệu có nhãn.

V. KẾT QUẢ THỰC NGHIỆM

A. Các phương pháp đánh giá

- 1) **Mean Absolute Percentage Error (MAPE)** là trung bình cộng phần trăm độ lệch tuyệt đối giữa giá trị dự đoán và giá trị thực.

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

- 2) **Root Mean Square Error (RMSE)** là căn bậc hai của trung bình bình phương độ lệch giữa giá trị dự đoán và giá trị thực tế.

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

- 3) **Mean Average Error (MAE)** là trung bình độ lệch tuyệt đối giữa giá trị dự đoán và giá trị thực tế.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Trong đó:

- n là tổng số quan sát
- y_i là giá trị thực tế của quan sát thứ i .
- \hat{y}_i là giá trị dự đoán của quan sát thứ i .

B. Kết quả thực nghiệm và đánh giá

1) Dữ liệu giá Vàng:



Hình 10. Linear Regression tỷ lệ 8:2



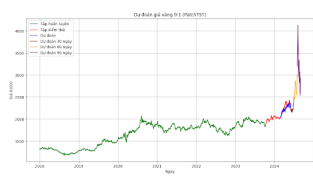
Hình 12. Mô hình RNN tỷ lệ 7:3



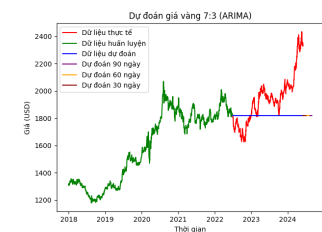
Hình 14. Mô hình LSTM tỷ lệ 7:3



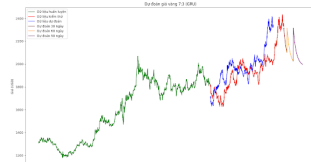
Hình 16. Mô hình TBATS tỷ lệ 8:2



Hình 18. Mô hình PatchTST tỷ lệ 9:1



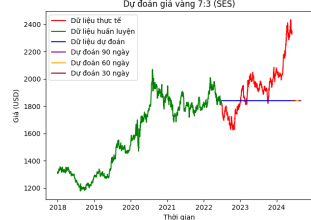
Hình 11. Mô hình ARIMA tỷ lệ 7:3



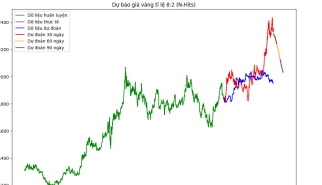
Hình 13. Mô hình GRU tỷ lệ 7:3



Hình 15. Mô hình FFT tỷ lệ 8:2



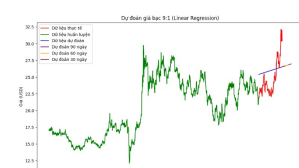
Hình 17. Mô hình SES tỷ lệ 7:3



Hình 19. Mô hình N-Hits tỷ lệ 8:2

Hình 20. Các mô hình dự đoán giá vàng tốt nhất

2) Dữ liệu giá Bạc:



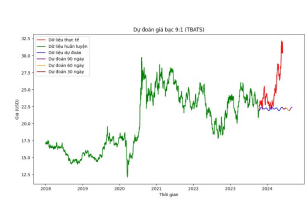
Hình 21. Linear Regression tỷ lệ 9:1



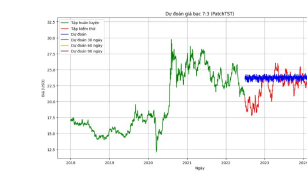
Hình 23. Mô hình RNN tỷ lệ 8:2



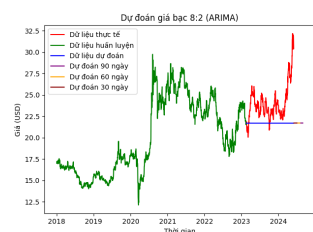
Hình 25. Mô hình LSTM tỷ lệ 8:2



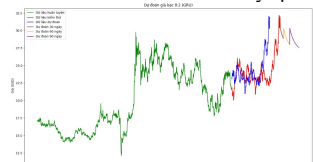
Hình 27. Mô hình TBATS tỷ lệ 9:1



Hình 29. Mô hình PatchTST tỷ lệ 7:3



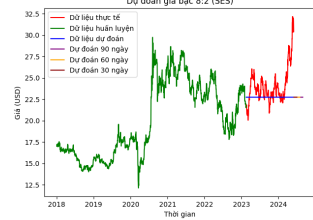
Hình 22. Mô hình ARIMA tỷ lệ 8:2



Hình 24. Mô hình GRU tỷ lệ 8:2



Hình 26. Mô hình FFT tỷ lệ 8:2



Hình 28. Mô hình SES tỷ lệ 8:2



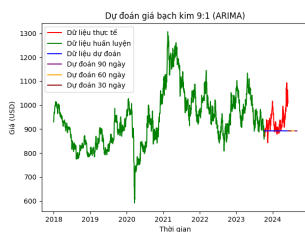
Hình 30. Mô hình N-Hits tỷ lệ 9:1

Hình 31. Các mô hình dự đoán giá bạc tốt nhất

3) Dữ liệu giá Bạch Kim:



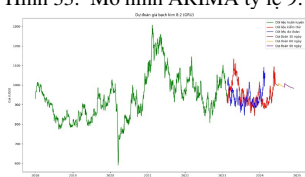
Hình 32. Linear Regression tỷ lệ 9:1



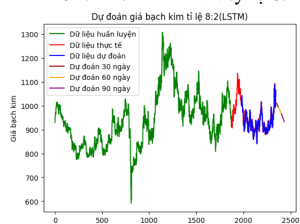
Hình 33. Mô hình ARIMA tỷ lệ 9:1



Hình 34. Mô hình RNN tỷ lệ 8:2



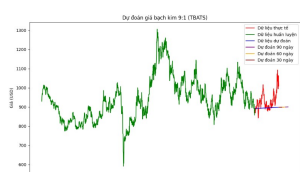
Hình 35. Mô hình GRU tỷ lệ 8:2



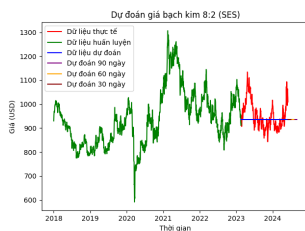
Hình 36. Mô hình LSTM tỷ lệ 8:2



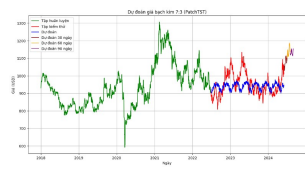
Hình 37. Mô hình FFT tỷ lệ 8:2



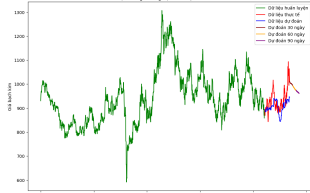
Hình 38. Mô hình TBATS tỷ lệ 9:1



Hình 39. Mô hình SES tỷ lệ 8:2



Hình 40. Mô hình PatchTST tỷ lệ 7:3



Hình 41. Mô hình N-Hits tỷ lệ 9:1

Hình 42. Các mô hình dự đoán giá vàng tốt nhất

VI. KẾT LUẬN

Xét theo chiều ngang, với dữ liệu giá vàng thì tỉ lệ tốt nhất cho các mô hình là 7:3 với số lần có kết quả tốt nhất là 5, tỉ lệ tốt thứ hai là 8:2 (4 lần), kém nhất là tỉ lệ 9:1 (1 lần). Tương tự với dữ liệu giá bạc kim thì tốt nhất là tỉ lệ 8:2 (5 lần), tiếp đó là tỉ lệ 9:1 (4 lần) và 7:3 (1 lần). Với dữ liệu giá bạc thì kém nhất là tỉ lệ 7:3 (2 lần) và 9:1 (3 lần) và tốt nhất là tỉ lệ 8:2 (5 lần). Như vậy tỉ lệ tốt nhất cho toàn bộ dữ liệu là 8:2.

Xét theo chiều dọc, trong với mỗi tỉ lệ, ứng với mỗi phương pháp kiểm định, nhóm nghiên cứu sẽ chọn ra 3 mô hình dự báo tốt nhất; top 1, 2, 3, sẽ được gán hệ số 3, 2, 1 và mô hình nào có tổng hệ số cao nhất thì chọn. Từ đó, chọn ra 3 mô hình tốt nhất của mỗi bộ dữ liệu.

Bộ dữ liệu	Xếp hạng	Tỉ lệ 7:3	Tỉ lệ 8:2	Tỉ lệ 9:1	3 mô hình tốt nhất
Giá vàng	Top 1	GRU (9)	GRU (9)	LSTM (9)	GRU
	Top 2	RNN (6)	RNN (6)	GRU (6)	RNN
	Top 3	LSTM (3)	LSTM (3)	RNN (3)	LSTM (top 2)
Giá bạc kim	Top 1	LSTM (8)	GRU (9)	GRU (8)	GRU
	Top 2	GRU (7)	LSTM (6)	LSTM (7)	LSTM
	Top 3	RNN (3)	RNN (3)	RNN (3)	RNN
Giá bạc	Top 1	GRU (9)	GRU (9)	GRU (9)	GRU
	Top 2	LSTM (6)	RNN (6)	LSTM (6)	LSTM
	Top 3	RNN (3)	LSTM (3)	RNN (3)	RNN

Bảng II

BẢNG XẾP HẠNG CÁC MÔ HÌNH ĐƯỢC ĐÁNH GIÁ THEO CHIỀU DỌC

Cuối cùng sau khi xét theo chiều dọc, từ những mô hình được chọn là tốt nhất trong cả ba bộ dữ liệu thì nhóm nhận thấy mô hình tốt nhất là GRU. Trong số những mô hình được kỳ vọng sẽ cho ra kết quả tốt thì chỉ có mô hình LSTM được chọn là tốt, còn mô hình ARIMA và TBATS thì có kết quả dự đoán không thực sự tốt. Trong cả 3 mô hình tốt nhất thì luôn là GRU, LSTM, RNN và đều là thuật toán học sâu và kết quả chênh lệch nhau không quá lớn, điều này cho thấy các thuật toán dạng này cho ra kết quả khả quan hơn cho các bộ dữ liệu giá khoáng sản này, PatchTST và N-HiTS cũng cho ra kết quả khá tốt khi kết quả luôn không quá lệch so với top 3.

Như vậy, mô hình GRU theo kết quả thì cho thấy khả năng học tuyệt vời với tiềm năng dự đoán giá ba loại kim loại quý trong tương lai. Từ kết quả này, các nhóm nghiên cứu và nhà phân tích dữ liệu trong tương lai có thể cân nhắc về những mô hình có thể dự đoán tốt và mở rộng nghiên cứu dự đoán cho cả các loại kim loại khác hoặc lĩnh vực khác có liên quan cũng như cải tiến mô hình nhằm phục vụ tốt hơn cho các nhu cầu học thuật và kinh doanh.

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến Phó Giáo sư Tiến sĩ Nguyễn Đình Thuận và Trợ giảng Nguyễn Minh Nhựt vì sự tận tâm và hướng dẫn nhiệt tình của thầy và anh. Những bài học quý báu từ thầy đã giúp chúng em hoàn thành dự án này và sẽ là nền tảng vững chắc cho những nỗ lực tương lai của chúng em. Dự án này sẽ không thể hoàn thành nếu thiếu sự giám sát và chỉ đạo của thầy.

Trong suốt quá trình thực hiện dự án, chúng em đã gặp không ít khó khăn và thử thách. Nhưng nhờ có sự đoàn kết và hợp tác,

chúng em đã vượt qua tất cả và hoàn thành công việc. Dự án này cũng đã mang lại cho chúng em cơ hội thực hành thực tế, giúp chúng em tích lũy được nhiều kinh nghiệm quý báu.

Chúng em cũng xin gửi lời cảm ơn đến các thành viên trong nhóm vì những đóng góp và sự hỗ trợ nhiệt tình trong suốt quá trình thực hiện dự án. Nhờ có sự hỗ trợ và cộng tác của các bạn, chúng em đã học hỏi được nhiều điều và hoàn thành công việc một cách tốt nhất.

TÀI LIỆU

- [1] Tawum Juvert Mbah, Haiwang Ye, Jianhua Zhang Mei Long, Using LSTM and ARIMA to Simulate and Predict Limestone Price Variations (06/01/2021).
- [2] Bojun Yin, Renguang Zuo, Yihui Xiong, Mineral Prospectivity Mapping via Gated Recurrent Unit Model (25/11/2021).
- [3] F.Javier Galán-Sales, Pablo Reina-Jiménez, Manuel Carranza-García, José María Luna-Romera, An Approach to Enhance Time Series Forecasting by Fast Fourier Transform (31/08/2023).
- [4] Improve, A. 27 F. (2018, October 3). Introduction to recurrent neural network. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>
- [5] Gated recurrent unit networks. (2019, July 9). GeeksforGeeks. <https://www.geeksforgeeks.org/gated-recurrent-unit-networks/>
- [6] De Livera, A. M., Hyndman, R. J., Snyder, R. D. (n.d.). Forecasting time series with complex seasonal patterns using exponential smoothing. Robjhyndman.com. Retrieved June 20, 2024, from <https://robjhyndman.com/papers/ComplexSeasonality.pdf> (trang 9, 10, 11, 12)
- [7] VanderPlas, J. (n.d.). Understanding the FFT algorithm. Github.Io. Retrieved June 20, 2024, from <https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/>
- [8] Yuqi Nie, Nam h. Nguyen, Phanwadee Sinthong, Jayant Kalagnanam, A Time Series is Worth 64 Words: Long-term Forecasting with Transformers (published 5 Mar 2023).
- [9] Gough, B. (n.d.). FFT Algorithms. Cinvestav.Mx. Retrieved June 20, 2024, from https://www.tamps.cinvestav.mx/wgomez/material/AID/fft_algorithms.pdf