**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****🙜🙢🙠🙞

**ĐỒ ÁN CHUYÊN NGÀNH**

**ĐỀ TÀI: DỰ ĐOÁN GIÁ VÀNG THEO THUẬT TOÁN RFNN**

Giảng viên hướng dẫn: Đào Minh Châu

Sinh viên thực hiện:

1. MSSV: 2001216021 - Nguyễn Hoàng Phát

2. MSSV: 2001216097 - Nguyễn Như Quỳnh

3. MSSV: 2001210578 - Nguyễn Ngọc Nghi

TP. HỒ CHÍ MINH, NGÀY 15 THÁNG 09 NĂM 2024

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN CHUYÊN NGÀNH**

**ĐỀ TÀI: GIẢI THUẬT RFNN ỨNG DUNG TRONG DỰ BÁO GIÁ VÀNG**

**Giảng viên hướng dẫn: Đào Minh Châu**

**Sinh viên thực hiện:**

1. MSSV: 2001216021 - Nguyễn Hoàng Phát

2. MSSV: 2001216097 - Nguyễn Như Quỳnh

3. MSSV: 2001210578 - Nguyễn Ngọc Nghi

TP. HỒ CHÍ MINH, NGÀY 15 THÁNG 09 NĂM 2024

**LỜI CẢM ƠN**

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép em được bày tỏ lòng biết ơn đến quý thầy cô ở Khoa Công nghệ Thông tin đã tạo môi trường học tập và truyền đạt vốn kiến thức quý báu cho chúng em trong suốt gần 4 năm học tại trường Đại học Công Thương TP.HCM. Nhờ có những lời hướng dẫn, dạy bảo và sự giúp đỡ của các thấy cô nên nhóm em đã có thể hoàn thành đề tài một cách tốt đẹp.

Em xin chân thành cảm ơn đến cô **Đào Minh Châu** người đã trực tiếp giúp đỡ, góp ý và hướng dẫn nhóm em hoàn thành đề tài này trong suốt khoảng thời gian qua. Với khoảng thời gian ngắn ngủi, nhóm em có thể sẽ không tránh khỏi những thiếu sót, nhóm em rất mong sẽ nhận được những ý kiến đóng góp quý báu của quý thầy cô để nhóm em có thể nắm vững được những kiến thức đồng thời có điều kiện bổ sung nâng cao ý thức của mình.

Nhóm em xin chân thành cảm ơn quý thầy cô!

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**Tên đề tài:** Giải Thuật RFNN Ứng Dụng Trong Dự Báo Giá Vàng

**Sinh viên thực hiện:**

1. Họ và tên: Nguyễn Hoàng Phát MSSV: 2001216021 **Lớp:** 12DHTH08
2. Họ và tên: Nguyễn Như Quỳnh MSSV: 2001216097 **Lớp:** 12DHTH11
3. Họ và tên: Nguyễn Ngọc Nghi MSSV: 2001210578 **Lớp:** 12DHTH14

**Niên Khoá:** 2021-2025

**Nhận xét của giảng viên hướng dẫn:**

**............................................................................................................................ ............................................................................................................................ ............................................................................................................................ ............................................................................................................................ ............................................................................................................................ ............................................................................................................................ ........................................................................................................................... ............................................................................................................................ ............................................................................................................................ ............................................................................................................................ ............................................................................................................................ ............................................................................................................................**

TP. Hồ Chí Minh, Ngày 15 Tháng 9 Năm 2024

**Giảng viên hướng dẫn**

(Ký và ghi rõ họ tên)

**MỤC LỤC**

[MỤC LỤC HÌNH ẢNH 7](#_Toc184733471)

[LỜI MỞ ĐẦU 10](#_Toc184733472)

[CHƯƠNG 1. TỔNG QUAN 11](#_Toc184733473)

[**1.1.** **Mục tiêu lý do chọn đề tài:** 11](#_Toc184733474)

[**1.1.1.** **Lý do chọn đề tài:** 11](#_Toc184733475)

[**1.1.2.** **Mục tiêu:** 11](#_Toc184733476)

[**1.2.** **Ý tưởng xây dựng phần mềm:** 11](#_Toc184733477)

[**1.3.** **Phương pháp nghiên cứu:** 12](#_Toc184733478)

[**1.3.1.** **Nghiên cứu lý thuyết:** 12](#_Toc184733479)

[**1.3.2.** **Nghiên cứu thực nghiệm:** 12](#_Toc184733480)

[**1.4.** **Đối tượng nghiên cứu:** 12](#_Toc184733481)

[**1.5. Phạm vi nghiên cứu:** 12](#_Toc184733482)

[CHƯƠNG 2. CẤU TRÚC VÀ HOẠT ĐỘNG CỦA MÔ HÌNH RFNN 13](#_Toc184733483)

[**2.1.** **Giới thiệu về Recurrent Fuzzy Neural Network (RFNN):** 13](#_Toc184733484)

[**2.2.** **Cấu trúc của mô hình RFNN:** 13](#_Toc184733485)

[**2.3.** **Cách thức hoạt động của mô hình RFNN** 18](#_Toc184733486)

[**2.4.** **Các yếu tố ảnh hưởng đến hiệu quả của RFNN** 19](#_Toc184733487)

[**2.4.1.** **Số lượng nút ẩn:** 19](#_Toc184733488)

[**2.4.2.** **Chọn lựa các quy tắc mờ:** 19](#_Toc184733489)

[**2.4.3.** **Hệ số học và tham số:** 20](#_Toc184733490)

[**2.5.** **Ưu điểm và nhược điểm của RFNN** 20](#_Toc184733491)

[**2.6.** **Các ứng dụng của RFNN:** 20](#_Toc184733492)

[CHƯƠNG 3. CÀI ĐẶT RFNN TRONG DỰ ĐOÁN GIÁ VÀNG 21](#_Toc184733493)

[**3.1.** **Cài đặt mô hình RFNN:** 21](#_Toc184733494)

[**3.2.** **Áp dụng RFNN trong dự đoán giá vàng:** 21](#_Toc184733495)

[**3.2.1.** **Tạo Mô Hình Dự Báo RFNN:** 21](#_Toc184733496)

[**3.2.2.** **Hiển Thị Biểu Đồ và Cập Nhật Tooltip:** 24](#_Toc184733497)

[CHƯƠNG 4. KẾT LUẬN 43](#_Toc184733498)

[**4.1.** **Kết quả đạt được:** 43](#_Toc184733499)

[**4.2.** **Đề xuất và nghiên cứu:** 43](#_Toc184733500)

[**4.3.** **Kết luận** 43](#_Toc184733501)

[TÀI LIỆU THAM KHẢO 45](#_Toc184733502)

# MỤC LỤC HÌNH ẢNH

[Hình 1: Các lớp của mô hình 14](#_Toc184229380)

[Hình 2: Phần hồi tiếp (recurrent loop) 15](#_Toc184229381)

[Hình 3: Biểu Đồ Giá Vàng và Khối Lượng Giao Dịch 25](#_Toc184229382)

[Hình 4: Tooltip và Dự Báo Giá Vàng Tương Lai 30](#_Toc184229383)

[Hình 5: Chức năng các nút 34](#_Toc184229384)

[Hình 6: Mô hình dự đoán 35](#_Toc184229385)

[Hình 7: Mô hình dự đoán 1 tuần 36](#_Toc184229386)

[Hình 8: Mô hình dự đoán 1 tháng 37](#_Toc184229387)

[Hình 9: Mô hình dự đoán 1 năm 38](#_Toc184229388)

[Hình 10: Mô hình dự đoán RMSE 39](#_Toc184229389)

**DANH MỤC CHỮ, KÍ HIỆU VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **KÝ HIỆU  CHỮ VIẾT TẮT** | **CHỨ VIẾT ĐẦY ĐỦ** |
| 1 | API | Application Programming Interface |
| 2 | IT | information technology |
| 3 | BI | Business Intelligence |
| 4 | GUI | Graphical User Interface |
| 5 | HTML | HyperText Markup Language |
| 6 | CSS | Cascading Style Sheets |
| 7 | JSON | JavaScript Object Notation |
| 8 | AI | Artificial Intelligence |
| 9 | ML | Machine Learning |
| 10 | Machine Learning | Liquid Crystal Display |
| 11 | DL | Deep Learning |
| 12 | RFNN | Radial Basis Function Neural Network |
| 13 | MSE | Mean Squared Error |
| 14 | RMSE | Root Mean Squared Error |
| 15 | GA | Genetic Algorithm |
| 16 | SGD | Stochastic Gradient Descent |
| 17 | **RNN** | Fuzzy Neural Network |
| 18 | FM | Fuzzy Modeling |
| 19 | FIS | Fuzzy Inference System |
| 20 | CNN | Convolutional Neural Network |

# LỜI MỞ ĐẦU

Hiện nay, trong thời đại công nghệ phát triển vượt bậc, lĩnh vực tài chính và đầu tư đang trở thành một trong những ngành hot, đặc biệt là việc dự đoán giá vàng. Trước kia, việc dự đoán và phân tích giá vàng chủ yếu dựa vào kinh nghiệm và các phương pháp thống kê truyền thống. Tuy nhiên, với sự xuất hiện của các thuật toán học máy và trí tuệ nhân tạo (AI), việc dự đoán giá vàng đã trở nên chính xác và hiệu quả hơn bao giờ hết, đặc biệt khi áp dụng thuật toán RFNN.

Thuật toán RFNN, một loại mạng nơ-ron với hàm cơ sở đối xứng (RBF), có khả năng học hỏi và nhận diện các mẫu dữ liệu phức tạp từ nhiều nguồn thông tin. Điều này rất hữu ích trong việc dự đoán giá vàng, vì thị trường vàng thường bị ảnh hưởng bởi rất nhiều yếu tố khó lường như tình hình kinh tế toàn cầu, sự biến động của các loại tài sản khác, chính sách của các ngân hàng trung ương, và các yếu tố chính trị.

Trước đây, việc dự đoán giá vàng là một công việc rất khó khăn và tốn thời gian. Các phương pháp thủ công như phân tích biểu đồ, tính toán từ dữ liệu quá khứ thường không mang lại kết quả chính xác. Hơn nữa, những phương pháp này thường không thể xử lý được các mối quan hệ phức tạp giữa các yếu tố tác động đến giá vàng. Việc áp dụng thuật toán RFNN đã giải quyết được rất nhiều khó khăn trong việc dự đoán giá vàng. Cụ thể:

* **Dự đoán chính xác hơn**: RFNN có khả năng học và điều chỉnh theo các mẫu dữ liệu lịch sử, giúp đưa ra dự đoán chính xác hơn.
* **Tiết kiệm thời gian**: Quy trình dự đoán được tự động hóa, giúp giảm thiểu thời gian tính toán và phân tích, từ đó nhanh chóng đưa ra quyết định đầu tư.

Khi áp dụng thuật toán RFNN vào việc dự đoán giá vàng, các nhà đầu tư có thể dự đoán được xu hướng giá vàng trong tương lai một cách chính xác và kịp thời, từ đó đưa ra quyết định đầu tư hiệu quả hơn. Điều này không chỉ giúp giảm thiểu rủi ro mà còn tối ưu hóa lợi nhuận cho nhà đầu tư.

Nhờ vào sự hỗ trợ mạnh mẽ từ các công nghệ tiên tiến và các thuật toán học máy, việc dự đoán giá vàng đã trở thành một công cụ hữu ích trong việc ra quyết định đầu tư. Dự án sử dụng thuật toán RFNN để dự đoán giá vàng đã thành công trong việc cung cấp những dự đoán chính xác, từ đó tạo ra giá trị đáng kể cho các nhà đầu tư. Tuy nhiên, như với bất kỳ công nghệ nào, luôn cần cải thiện và tối ưu hóa thuật toán để đáp ứng tốt hơn yêu cầu và thách thức của thị trường tài chính đầy biến động

# CHƯƠNG 1. TỔNG QUAN

1. **Mục tiêu lý do chọn đề tài:**
2. **Lý do chọn đề tài:**

**Tầm quan trọng của dự báo giá vàng**: Giá vàng có ảnh hưởng sâu rộng đến nền kinh tế toàn cầu và các quyết định đầu tư của cá nhân, tổ chức. Do tính chất biến động mạnh mẽ của giá vàng, việc dự báo chính xác giá vàng giúp các nhà đầu tư và chuyên gia tài chính đưa ra các quyết định đúng đắn, giảm thiểu rủi ro.

**Máy học trong dự báo**: Các phương pháp truyền thống như ARIMA hay hồi quy không đủ linh hoạt để dự báo các chuỗi thời gian phức tạp như giá vàng. Các mô hình máy học, đặc biệt là **Recurrent Fuzzy Neural Network (RFNN)**, có khả năng học và mô phỏng các mối quan hệ phi tuyến tính trong dữ liệu. RFNN kết hợp các đặc điểm của mạng nơ-ron hồi quy (Recurrent Neural Networks - RNN) và hệ mờ (Fuzzy Systems), giúp nó xử lý các yếu tố không chắc chắn và dự báo chuỗi thời gian hiệu quả hơn.

**Tính ứng dụng của RFNN**: RFNN là một mô hình mạnh mẽ trong máy học, đặc biệt là trong các bài toán dự báo chuỗi thời gian và phân loại. Việc áp dụng RFNN vào dự báo giá vàng sẽ giúp cải thiện độ chính xác so với các mô hình truyền thống nhờ khả năng xử lý các yếu tố mờ và phi tuyến trong dữ liệu.

1. **Mục tiêu:**

**Mục tiêu tổng quát**: Tìm hiểu và áp dụng mô hình **RFNN** vào dự báo giá vàng, nhằm giúp các nhà đầu tư có công cụ dự báo chính xác hơn.

**Mục tiêu cụ thể**:

* Tìm hiểu và phân tích các phương pháp máy học trong dự báo chuỗi thời gian, đặc biệt là RFNN.
* Nghiên cứu cấu trúc và hoạt động của mô hình RFNN.
* Cài đặt mô hình RFNN bằng Python để dự báo giá vàng.
* Đánh giá hiệu quả của mô hình qua chỉ số như **RMSE** (Root Mean Squared Error).

1. **Ý tưởng xây dựng phần mềm:**

Ứng dụng RFNN trong phần mềm: Phát triển một phần mềm dự báo giá vàng dựa trên mô hình RFNN, giúp người dùng nhập dữ liệu và nhận kết quả dự báo.

**Chức năng phần mềm**:

* Nhập dữ liệu giá vàng lịch sử từ các nguồn uy tín.
* Áp dụng mô hình RFNN để dự báo giá vàng trong tương lai.
* Hiển thị kết quả dự báo và các chỉ số đánh giá (RMSE) để người dùng có thể đánh giá độ chính xác của dự báo.

1. **Phương pháp nghiên cứu:**
2. **Nghiên cứu lý thuyết:**

Tìm hiểu về các phương pháp dự báo truyền thống và phương pháp máy học, đặc biệt là RFNN. Nghiên cứu các mô hình mạng nơ-ron hồi quy (RNN), mạng nơ-ron mờ (Fuzzy Systems) và cách kết hợp chúng trong RFNN.

Đọc và phân tích các nghiên cứu trước về việc áp dụng RFNN trong dự báo chuỗi thời gian và tài chính.

1. **Nghiên cứu thực nghiệm:**

* Cài đặt mô hình RFNN bằng Python, sử dụng các thư viện như scikit-learn, pandas, numpy, và matplotlib
* Tiến hành tiền xử lý dữ liệu (chuẩn hóa, làm sạch dữ liệu) và huấn luyện mô hình RFNN.
* Đánh giá mô hình bằng các chỉ số hiệu suất như RMSE trên dữ liệu thực tế về giá vàng.

1. **Đối tượng nghiên cứu:**

**Dữ liệu**: Dữ liệu lịch sử về giá vàng từ các nguồn uy tín như [Gold Futures Price Today - Investing.com](https://www.investing.com/commodities/gold), https://xauusd.vn/ (bao gồm giá mở cửa, đóng cửa, cao nhất, thấp nhất và khối lượng giao dịch).

**Mô hình RFNN**: Nghiên cứu cấu trúc của RFNN, từ lớp đầu vào, lớp ẩn mờ (fuzzy inference layer), lớp hồi quy (recurrent layer), và lớp đầu ra.

# Phạm vi nghiên cứu:

Nội dung nghiên cứu của đề tài này tập trung vào việc tìm hiểu và ứng dụng mô hình Recurrent Fuzzy Neural Network (RFNN) trong dự báo giá vàng, một chủ đề có tính ứng dụng cao trong lĩnh vực tài chính và đầu tư.

# CHƯƠNG 2. CẤU TRÚC VÀ HOẠT ĐỘNG CỦA MÔ HÌNH RFNN

1. **Giới thiệu về Recurrent Fuzzy Neural Network (RFNN):**

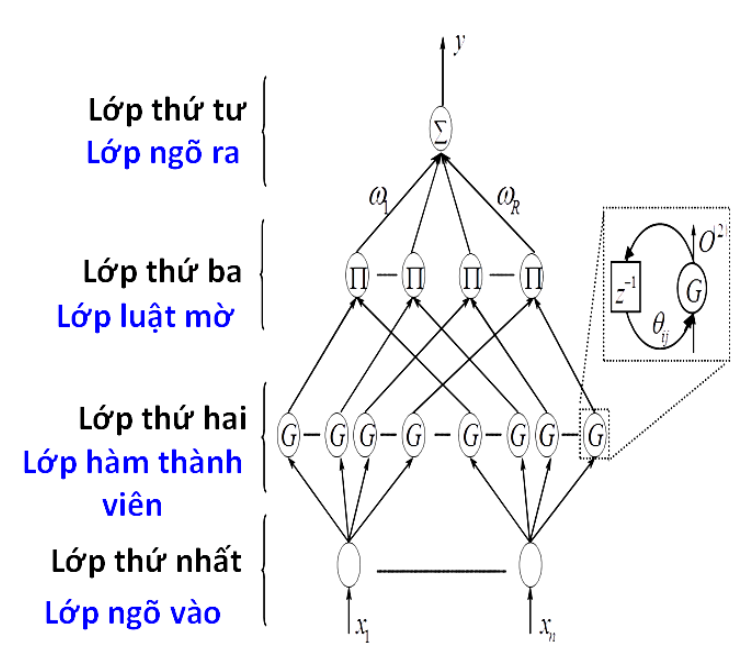
RFNN là một mạng nơ-ron mờ hồi quy, kết hợp giữa các nguyên lý của mạng nơ-ron hồi quy (Recurrent Neural Networks - RNN) và hệ thống mờ (Fuzzy Systems). Mô hình này được thiết kế để xử lý các bài toán phức tạp, có sự biến đổi không xác định hoặc không chắc chắn, thường gặp trong các dự báo giá trị theo thời gian, như giá vàng, giá cổ phiếu, hoặc các chỉ số tài chính khác.

RFNN có khả năng học và mô phỏng các mối quan hệ phi tuyến trong dữ liệu, đồng thời xử lý sự không chắc chắn và biến động trong dữ liệu, đặc biệt là dữ liệu **chuỗi thời gian**. Chuỗi thời gian, vốn có tính chất tuần tự và phụ thuộc vào các giá trị trước đó, là yếu tố quan trọng giúp mô hình RFNN hiểu và dự báo các giá trị tương lai.

Mô hình này sử dụng các mạng nơ-ron hồi quy (RNN) để lưu trữ thông tin từ quá khứ và lý thuyết mờ (Fuzzy Logic) để xử lý sự không chắc chắn, giúp nâng cao khả năng dự báo trong các bài toán liên quan đến chuỗi thời gian.

1. **Cấu trúc của mô hình RFNN:**

Mô hình RFNN kết hợp các yếu tố của mạng nơ-ron hồi quy và hệ mờ, gồm các thành phần chính sau:



Hình 1: Các lớp của mô hình

**Lớp ngõ vào:**

Đây là lớp **đầu vào** của RFNN, nơi mà các đầu vào của hệ thống được cung cấp. Các biến đầu vào thường được biểu diễn bởi x1,x2,…,xn​. Mỗi giá trị đầu vào này sẽ được gửi đến các nơ-ron ở lớp tiếp theo (lớp hàm thành viên) để tiếp tục xử lý.

Công thức: Oi(1)​(k) = xi​(k) (1)

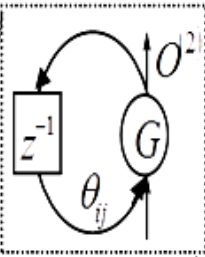
Trong đó:   
Oi(1)​(k) là đầu ra của ngõ vào thứ i tại thời điểm k  
xi(k) là ngõ vào thứ icủa lớp thứ nhất.  
**Ý nghĩa**: Đây là lớp **đầu vào**, nơi nhận các giá trị xi(k) tại thời điểm k. Mỗi xi(k) là một thành phần của dữ liệu đầu vào.

Lớp này đưa dữ liệu thô vào hệ thống để xử lý. Ở mỗi thời điểm k, các giá trị đầu vào có thể thay đổi và được chuyển qua các lớp kế để xử lý kỹ hơn

**Lớp hàm thành viên:**

Lớp này áp dụng các **hàm thành viên mờ** để biến đổi các giá trị đầu vào thành các giá trị mờ. Mỗi hàm thành viên (thường được ký hiệu là G) đại diện cho các mức độ thành viên của các giá trị đầu vào đối với một tập mờ cụ thể. Ví dụ: Nếu một đầu vào x1​ đại diện cho nhiệt độ, lớp hàm thành viên này có thể xác định mức độ mà x1​ thuộc vào các tập mờ.

Đây là bước đầu tiên của việc làm mờ đầu vào trong hệ thống logic mờ.

Phần hồi tiếp (recurrent loop):

Hình 2: Phần hồi tiếp (recurrent loop)

* Ở góc phải của hình, bạn có thể thấy một phần mô tả mạch hồi tiếp. Đây là một thành phần quan trọng trong RFNN.
* Ký hiệu z-1 đại diện cho trạng thái trước đó của hệ thống, tức là giá trị đầu ra của RFNN tại một thời điểm trước đó sẽ được đưa trở lại làm đầu vào cho các lần tính toán tiếp theo.
* Mạch hồi tiếp này cho phép RFNN có khả năng ghi nhớ các trạng thái quá khứ và sử dụng chúng để đưa ra quyết định dựa trên thông tin liên tục trong thời gian.

Công thức: Oij(2)​(k) = exp( ) (2)

Trong đó:

Oij(2)​(k): Đầu ra của nút thứ j trên lớp thứ 2 tại thời điểm k. Đây là kết quả của hàm thành viên Gaussian mô tả mức độ thuộc về một tập mờ nhất định.

zj(k) = Oij(2)(k−1)θij + Oi(1)(k) : Đầu vào của hàm thành viên tại thời điểm k. Đây là tổ hợp tuyến tính giữa đầu ra của lớp trước (lớp 1) và trọng số θij

mij : Trung bình của hàm Gaussian, thể hiện điểm giữa của tập mờ.​

σij​: Độ lệch chuẩn của hàm Gaussian, thể hiện độ rộng của tập mờ.

θij ​: trọng số liên kết giữa các nơ-ron.

Ý nghĩa: Lớp này áp dụng các hàm thành viên Gaussian để biến đổi đầu vào thành các giá trị mờ. Công thức này cho phép hệ thống tính toán mức độ thành viên của mỗi đầu vào đối với các tập mờ khác nhau.

Lớp này đại diện cho lớp hàm thành viên (membership functions), với mỗi nút tính toán mức độ mà đầu vào của hệ thống thuộc về một tập mờ nào đó. Hàm Gaussian được dùng phổ biến để biểu diễn tập mờ này.

mij và σij được cập nhật trong quá trình học để hàm Gaussian có thể điều chỉnh tốt hơn theo tập dữ liệu huấn luyện.

Công thức hàm Gaussian:

Mỗi neuron trong lớp ẩn sẽ tính toán giá trị của hàm cơ sở xuyên tâm dựa trên khoảng cách giữa điểm đầu vào và "trung tâm" của neuron. Một hàm phổ biến là **hàm Gaussian**:

Trong đó:

* x là vector đầu vào.
* ci ​là vector trung tâm của neuron thứ iii.
* σi ​là tham số điều chỉnh (độ rộng của hàm Gaussian, còn gọi là "spread").

Khoảng cách thể hiện sự tương đồng giữa đầu vào và neuron. Nếu giá trị đầu vào gần với trung tâm neuron, đầu ra của neuron sẽ cao; nếu xa, đầu ra sẽ thấp.

**Lớp luật mờ:**

Đây là lớp quan trọng nhất của RFNN, nơi mà các luật mờ được áp dụng. Mỗi nơ-ron trong lớp này đại diện cho một luật mờ (rule) dựa trên các đầu vào từ lớp hàm thành viên. Lớp này thực hiện quá trình suy luận để quyết định đầu ra dựa trên các quy tắc mờ được xác định trước. Ký hiệu Π trong hình là quá trình nhân (product) của các đầu vào mờ từ lớp trước. Quá trình này tính toán mức độ thỏa mãn của một luật mờ dựa trên các giá trị đầu vào mờ tương ứng.

Công thức Oj(3)(k) = (3)

Trong đó:

Oj(3)​(k): Đầu ra của nút thứ j trên lớp thứ 3 tại thời điểm k, là giá trị của một **luật mờ**.

Oij(2)​(k): Đầu ra của lớp hàm thành viên thứ j ở lớp thứ 2.

**∏**: Ký hiệu tích, nghĩa là đầu ra Oj(3)​(k) được tính bằng cách nhân tất cả các đầu ra của các hàm thành viên từ lớp thứ 2.

Lớp này là lớp luật mờ (fuzzy rules layer), nơi các luật mờ được hình thành bằng cách kết hợp các kết quả từ lớp hàm thành viên.

Mỗi nút ở lớp này tính toán mức độ "kích hoạt" của một luật mờ bằng cách nhân các đầu ra từ lớp hàm thành viên tương ứng với các điều kiện đầu vào.

Quy tắc mờ (rule) sẽ xác định mức độ mà một tập hợp đầu vào có thể dẫn đến một đầu ra cụ thể.

Ý nghĩa: Lớp này sử dụng phép nhân giữa các kết quả từ lớp thứ hai để thực hiện quá trình suy luận dựa trên các luật mờ. Kết quả này được tính toán để xác định mức độ thỏa mãn của một luật mờ dựa trên các giá trị đầu vào mờ tương ứng. Đây là nơi mà các luật mờ được áp dụng. Bằng cách nhân các giá trị từ lớp trước, nó xác định các điều kiện của mỗi luật mờ và tính toán mức độ thỏa mãn của luật đó.

**Lớp ngõ ra:**

Sau khi lớp luật mờ hoàn thành, kết quả từ mỗi luật sẽ được tổng hợp lại (ký hiệu ∑) tại lớp ngõ ra để đưa ra kết quả cuối cùng y. Đây là lớp đầu ra của RFNN, nơi mà quyết định hoặc giá trị cuối cùng của hệ thống được xác định.

Công thức: O(4)(k) = (4)

O(4)(k): Đầu ra cuối cùng của mạng RFNN tại thời điểm k.

ωj: Trọng số kết nối từ nút thứ j của lớp thứ 3 tới đầu ra cuối cùng.

Oj(3)(k): Đầu ra của nút thứ j ở lớp thứ 3.

Ý nghĩa: Đây là lớp ngõ ra của RFNN, nơi mà tất cả các kết quả từ lớp luật mờ được tổng hợp lại để đưa ra kết quả cuối cùng. ωj là trọng số liên kết giữa lớp luật mờ và lớp ngõ ra.

Sau khi suy luận, lớp ngõ ra thực hiện quá trình tổng hợp tất cả các kết quả lại và đưa ra quyết định hoặc giá trị dự đoán cuối cùng.

Lớp này là lớp đầu ra (output layer), nơi các tín hiệu từ lớp luật mờ được tổng hợp lại để tạo ra đầu ra cuối cùng của mạng RFNN.

Đầu ra của mỗi luật mờ ở lớp thứ 3 được nhân với một trọng số ωj , và sau đó được tổng hợp để tạo ra dự đoán cuối cùng O(4)(k).

Trọng số ωj ​đóng vai trò điều chỉnh tầm quan trọng của mỗi luật mờ trong quá trình tính toán đầu ra.

Trong các bài toán dự báo, việc đánh giá độ chính xác của mô hình là một yếu tố quan trọng. Một trong những chỉ số phổ biến dùng để đánh giá độ chính xác của mô hình dự báo chuỗi thời gian là RMSE (Root Mean Square Error).

* **Định nghĩa RMSE**: RMSE là một chỉ số đo lường độ lệch chuẩn của các sai số giữa giá trị thực tế và giá trị dự báo của mô hình. Nó phản ánh mức độ chệch lệch trung bình giữa giá trị dự báo và giá trị thực tế, với sai số càng nhỏ thì RMSE càng thấp, biểu thị mô hình càng chính xác.
* **Công thức tính RMSE**: RMSE được tính theo công thức sau:

Trong đó:

* yi ​là giá trị thực tế tại thời điểm thứ i,
* là giá trị dự báo của mô hình tại thời điểm thứ i,
* n là số lượng mẫu dữ liệu.

RMSE giúp đo lường sự chênh lệch giữa giá trị thực tế và giá trị dự báo của mô hình. Giá trị RMSE càng nhỏ, mô hình càng dự báo chính xác.

## **Cách thức hoạt động của mô hình RFNN**

* **Quá trình huấn luyện:**

**Khởi tạo trọng số và tham số:** Các trọng số trong mạng nơ-ron và các tham số trong các quy tắc mờ được khởi tạo ngẫu nhiên.

**Tính toán giá trị mờ:** Mỗi đầu vào từ chuỗi thời gian được tính toán qua các hàm mờ trong lớp ẩn mờ, tạo ra các kết quả mờ (fuzzy outputs). Điều này giúp mô hình xử lý các yếu tố không chắc chắn trong dữ liệu chuỗi thời gian.

**Cập nhật trạng thái hồi quy:** Các trạng thái ẩn trong lớp hồi quy được cập nhật theo thời gian, giữ lại thông tin từ các bước trước, giúp mô hình nhận diện các xu hướng và thay đổi trong chuỗi thời gian. Đây là phần quan trọng giúp RFNN xử lý dữ liệu có tính tuần tự.

**Tính toán đầu ra:** Sau khi dữ liệu đi qua các lớp ẩn và hồi quy, lớp đầu ra sẽ đưa ra giá trị dự báo cuối cùng.

**Cập nhật trọng số:** Sử dụng thuật toán học (như Gradient Descent hoặc Backpropagation) để tối ưu hóa các trọng số và tham số mờ, sao cho sai số giữa dự đoán và giá trị thực tế được giảm thiểu.

* **Chuỗi thời gian và tính tuần tự trong RFNN:**

**Chuỗi thời gian:** RFNN xử lý dữ liệu chuỗi thời gian bằng cách sử dụng lớp hồi quy, cho phép mô hình học từ các bước trước trong chuỗi dữ liệu. Điều này giúp mô hình nhận diện các xu hướng, chu kỳ, và các yếu tố phụ thuộc theo thời gian.

**Tính tuần tự:** Mô hình RFNN có khả năng "ghi nhớ" các giá trị quá khứ (thông qua các trạng thái hồi quy) và sử dụng thông tin này để đưa ra dự báo cho các bước tiếp theo. Việc sử dụng các tham số mờ giúp mô hình hoạt động linh hoạt hơn trong việc xử lý các yếu tố không chắc chắn và biến động trong chuỗi thời gian.

**Dự báo chuỗi thời gian:** RFNN có thể dự báo giá trị trong tương lai dựa trên các dữ liệu trong quá khứ. Việc kết hợp giữa lý thuyết mờ và mạng nơ-ron giúp cải thiện độ chính xác của dự báo, đặc biệt trong các bài toán tài chính và các dữ liệu có tính biến động cao.

* **Dự báo giá vàng:**

Với mô hình RFNN, sau khi huấn luyện xong, bạn có thể sử dụng mô hình để dự báo giá vàng trong tương lai dựa trên các yếu tố đầu vào từ chuỗi thời gian như giá vàng trong quá khứ, khối lượng giao dịch, và các chỉ số kinh tế vĩ mô khác. RFNN có thể nhận diện các mối quan hệ giữa các yếu tố này và đưa ra dự báo giá trị vàng trong tương lai một cách chính xác hơn nhờ khả năng kết hợp giữa mạng nơ-ron hồi quy và lý thuyết mờ.

## **Các yếu tố ảnh hưởng đến hiệu quả của RFNN**

1. **Số lượng nút ẩn:**

Số lượng nút ẩn trong lớp ẩn mờ và hồi quy ảnh hưởng lớn đến khả năng học của mô hình. Cần phải tối ưu số lượng nút sao cho mô hình vừa đủ khả năng học nhưng không bị overfitting.

1. **Chọn lựa các quy tắc mờ:**

Các quy tắc mờ "If-Then" đóng vai trò quan trọng trong việc chuyển đổi đầu vào thành đầu ra. Việc chọn các quy tắc mờ hợp lý có thể cải thiện độ chính xác của mô hình.

1. **Hệ số học và tham số:**

Các tham số như tốc độ học (learning rate), độ rộng của các hàm mờ, và các tham số trong lớp hồi quy phải được điều chỉnh để tối ưu hóa hiệu suất mô hình.

1. **Ưu điểm và nhược điểm của RFNN**

**Ưu điểm**:

* **Khả năng học chuỗi thời gian**: Với cơ chế hồi quy, RFNN có thể "ghi nhớ" các giá trị quá khứ và sử dụng thông tin đó để cải thiện dự báo trong tương lai.
* **Khả năng mô hình hóa phi tuyến tính**: RFNN có thể học được các mối quan hệ phi tuyến trong dữ liệu, giúp dự báo chính xác hơn so với các phương pháp tuyến tính.

**Nhược điểm**:

* **Cần nhiều dữ liệu để huấn luyện**: RFNN yêu cầu một lượng dữ liệu lớn để huấn luyện, đặc biệt khi số lượng nút ẩn lớn.
* **Khó khăn trong việc tối ưu hóa tham số**: Việc tìm kiếm các tham số tối ưu cho mô hình có thể khó khăn và tốn thời gian.

## **Các ứng dụng của RFNN:**

RFNN được sử dụng rộng rãi trong các bài toán dự báo chuỗi thời gian, nhận dạng mẫu, và các bài toán liên quan đến dự báo tài chính, bao gồm:

**Dự báo giá vàng**:

RFNN có thể sử dụng dữ liệu lịch sử về giá vàng, khối lượng giao dịch và các yếu tố kinh tế vĩ mô để dự đoán xu hướng giá vàng trong tương lai.

**Các ứng dụng tài chính khác**: RFNN cũng có thể được áp dụng trong các bài toán dự báo giá cổ phiếu, tỷ giá ngoại tệ, và các thị trường tài chính khác.

Dự báo chuỗi thời gian không chắc chắn, nơi có sự biến động mạnh và các yếu tố không xác định (ví dụ: sự tác động của các yếu tố kinh tế vĩ mô).

# CHƯƠNG 3. CÀI ĐẶT RFNN TRONG DỰ ĐOÁN GIÁ VÀNG

## **Cài đặt mô hình RFNN:**

**Cài đặt mô hình bằng Python:**

**Thư viện sử dụng**: Sử dụng các thư viện như **scikit-learn**, **numpy**, **pandas**, **matplotlib** để cài đặt mô hình RFNN.

**Tiền xử lý dữ liệu**: Các bước tiền xử lý bao gồm việc chuẩn hóa dữ liệu và chuyển đổi dữ liệu chuỗi thời gian thành dạng phù hợp với mô hình RFNN.

## **Áp dụng RFNN trong dự đoán giá vàng:**

### **Tạo Mô Hình Dự Báo RFNN:**

Sau khi dữ liệu được chuẩn hóa, mô hình RFNN được áp dụng. Các bước cơ bản trong RFNN là:

**Lớp Đầu Vào (Input Layer)**: Dữ liệu đầu vào là các giá trị đã chuẩn hóa của Open, High, Low, Close, và Volume.

# Chuẩn hóa dữ liệu

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(data[['Open', 'High', 'Low', 'Close', 'Volume']])

# Lớp Đầu Vào

X = scaled\_data

**StandardScaler:** Sử dụng StandardScaler từ sklearn.preprocessing để chuẩn hóa dữ liệu, đảm bảo rằng các đặc trưng có trung bình bằng 0 và độ lệch chuẩn bằng 1.

**X:** Biến X chứa dữ liệu đã chuẩn hóa, được sử dụng làm đầu vào cho mô hình RFNN.

**Lớp Hàm Thành Viên:** Xử lý các hàm thành viên, thường là các hàm kích hoạt hoặc các chức năng xử lý đặc trưng trong mạng nơ-ron.

def gaussian\_rbf(x, c, s):

return np.exp(-np.linalg.norm(x - c) \*\* 2 / (2 \* s \*\* 2))

**Lớp Cơ Sở Radial Basis Function (RBF Layer):** Sử dụng hàm RBF để chuyển đổi dữ liệu đầu vào thành không gian đặc trưng mới dựa trên các tâm (centroids) và độ rộng (spread).

**def predict\_rfnn(days\_to\_predict):**

**global data, scaled\_data**

**X = scaled\_data**

**y\_close = data['Close'].values**

**y\_volume = data['Volume'].values**

**# Chọn 5 tâm từ dữ liệu**

**centroids = X[:5]**

**spreads = np.std(X, axis=0)**

**# Tính giá trị RBF cho từng điểm dữ liệu**

**rbf\_outputs = np.array([[gaussian\_rbf(x, c, s) for c, s in zip(centroids, spreads)] for x in X])**

**Lớp Luật Mờ (Fuzzy Rule Layer)**: Áp dụng các luật mờ bằng cách nhân các giá trị RBF với các trọng số ngẫu nhiên để tạo ra giá trị dự báo cho giá Close và Volume.

# Tạo trọng số ngẫu nhiên

weights\_close = np.random.uniform(size=rbf\_outputs.shape[1])

weights\_volume = np.random.uniform(size=rbf\_outputs.shape[1])

# Tính giá trị dự đoán mờ

fuzzy\_close = rbf\_outputs @ weights\_close

fuzzy\_volume = rbf\_outputs @ weights\_volume

**Lớp Đầu Ra (Output Layer)**: Cuối cùng, mô hình tính toán giá vàng dự báo và khối lượng dự báo cho các ngày tiếp theo. Mỗi giá trị dự đoán sẽ bị ảnh hưởng bởi trọng số ngẫu nhiên và nhiễu (random noise) để làm mô hình trở nên thực tế hơn.

predictions\_close = []

predictions\_volume = []

# Giá trị gốc để bắt đầu dự báo

base\_close = y\_close[-1]

base\_volume = y\_volume[-1]

for day in range(days\_to\_predict):

# Thêm nhiễu vào dự báo

close\_variation = fuzzy\_close[-1] + np.random.normal(0, base\_close \* 0.01)

volume\_variation = fuzzy\_volume[-1] + np.random.normal(0, base\_volume \* 0.05)

# Tính giá trị dự báo

predicted\_close = base\_close + close\_variation

predicted\_volume = base\_volume + volume\_variation

predictions\_close.append(predicted\_close)

predictions\_volume.append(predicted\_volume)

# Cập nhật giá trị gốc cho ngày tiếp theo

base\_close = predicted\_close

base\_volume = predicted\_volume

return predictions\_close, predictions\_volume

**Tính RMSE**: Để đánh giá chất lượng của mô hình, **RMSE (Root Mean Squared Error)** được tính toán giữa giá trị dự đoán và giá trị thực tế. Đây là một chỉ số quan trọng để đánh giá độ chính xác của mô hình.

from sklearn.metrics import mean\_squared\_error

import numpy as np

def calculate\_rmse(actual, predicted):

actual = np.array(actual)

predicted = np.array(predicted)

mse = np.mean((actual - predicted) \*\* 2)

return np.sqrt(mse)

Mô hình dự báo giá vàng sẽ tính toán các giá trị Close dự báo cho những ngày tới, cùng với khối lượng giao dịch (Volume). Dự báo này được hiển thị trên biểu đồ dưới dạng đường màu vàng (Predict). Kết quả dự báo cũng sẽ được so sánh với dữ liệu thực tế (dòng màu xanh).

### **Hiển Thị Biểu Đồ và Cập Nhật Tooltip:**

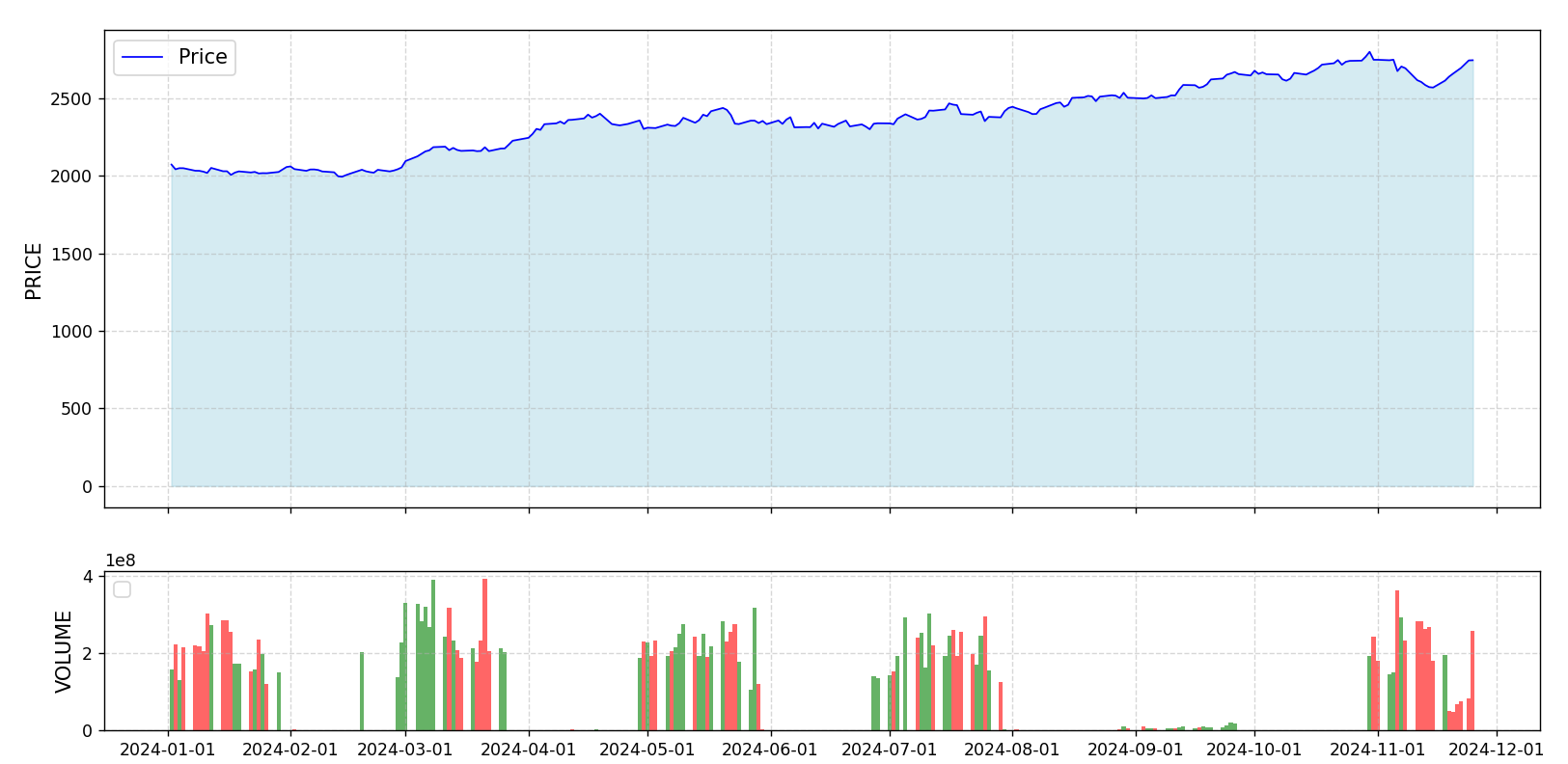
Phần tiếp theo là **hiển thị biểu đồ** cho phép người dùng quan sát trực quan các dữ liệu lịch sử (giá vàng và khối lượng) cũng như các dự báo trong tương lai. **Tooltip** được sử dụng để hiển thị chi tiết của giá vàng và khối lượng tại điểm người dùng rê chuột qua trên biểu đồ.

Biểu đồ sẽ gồm:

* **Biểu đồ Giá Vàng**: Hiển thị giá vàng trong quá khứ và các dự báo.
* **Biểu đồ Khối Lượng Giao Dịch**: Hiển thị khối lượng giao dịch của vàng, màu sắc của cột thể hiện sự thay đổi giá của vàng (màu xanh nếu giá vàng tăng, màu đỏ nếu giảm).

Phần cập nhật biểu đồ được thực hiện khi người dùng chọn khoảng thời gian dự báo (1 ngày, 1 tuần, 1 tháng, 1 năm). Mô hình sẽ tính toán các dự báo cho khoảng thời gian đó và hiển thị chúng trên biểu đồ.

**Biểu Đồ Giá Vàng và Khối Lượng Giao Dịch**



Hình 3: Biểu Đồ Giá Vàng và Khối Lượng Giao Dịch

Tạo một figure chứa hai subplots (ax1 và ax2) xếp theo cột, ax1 chiếm tỷ lệ cao hơn để hiển thị giá vàng, trong khi ax2 hiển thị khối lượng giao dịch. Tiêu đề tổng thể của biểu đồ được đặt là "GOLD PRICE PREDICTION CHART".

fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(16, 10), sharex=True, gridspec\_kw={'height\_ratios': [3, 1]})

fig.suptitle('GOLD PRICE PREDICTION CHART', fontsize=30, fontweight='bold', x=0.5)

* + Hàm show\_current\_data() vẽ biểu đồ dữ liệu hiện tại:

def show\_current\_data():

global future\_dates, future\_close, future\_volume

future\_dates = []

future\_close = []

future\_volume = []

ax1.clear()

ax2.clear()

ax1.plot(data.index, data['Close'], label='Price', color='blue', linewidth=1)

ax1.fill\_between(data.index, data['Close'], color='lightblue', alpha=0.5)

ax1.set\_ylabel('PRICE', fontsize=12)

ax1.grid(True, linestyle='--', alpha=0.5)

ax1.legend(loc='upper left', fontsize=12)

colors = ['green' if data['Close'].iloc[i] >= data['Open'].iloc[i] else 'red' for i in range(len(data))]

ax2.bar(data.index, data['Volume'], color=colors, alpha=0.6, width=1.0)

ax2.set\_ylabel('VOLUME', fontsize=12)

ax2.set\_xlabel('TIME', fontsize=13)

ax2.grid(True, linestyle='--', alpha=0.5)

ax2.legend(loc='upper left', fontsize=12)

ax2.xaxis.set\_major\_formatter(mdates.DateFormatter('%Y-%m-%d'))

plt.xticks(rotation=45)

plt.draw()

Các dòng mã quan trọng:

* ax1.plot(data.index, data['Close'], ...): Vẽ đường biểu diễn giá đóng cửa của vàng trên trục ax1.
* ax1.fill\_between(data.index, data['Close'], ...): Tô màu dưới đường giá để làm nổi bật vùng giá.
* ax2.bar(data.index, data['Volume'], ...): Vẽ biểu đồ thanh khối lượng giao dịch trên trục ax2, với màu xanh lá cây nếu giá tăng và đỏ nếu giá giảm.
* Các dòng thiết lập nhãn trục, lưới, và định dạng ngày tháng.

Cách thực thi:

Hàm này được gọi sau khi dữ liệu được xử lý thành công trong hàm select\_file(). Nó xóa các trục hiện tại (ax1.clear(), ax2.clear()) và vẽ lại biểu đồ với dữ liệu mới.

* Hàm update\_plot() Cập nhật biểu đồ với dự đoán tương lai:

def update\_plot(interval):

global future\_dates, future\_close, future\_volume

if data is None:

print("Chưa có dữ liệu. Vui lòng chọn file trước!")

return

days\_map = {'day': 1, 'week': 7, 'month': 30, 'year': 365}

days\_to\_predict = days\_map[interval]

future\_close, future\_volume = predict\_rfnn(days\_to\_predict)

base\_date = data.index[-1]

future\_dates = [base\_date + pd.Timedelta(days=i) for i in range(1, days\_to\_predict + 1)]

ax1.clear()

ax2.clear()

if interval == 'day':

ax1.scatter(future\_dates[0], future\_close[0], color='orange', s=5)

ax1.plot(data.index, data['Close'], label='Price', color='blue', linewidth=1)

ax1.fill\_between(data.index, data['Close'], color='lightblue', alpha=0.5)

ax1.plot(future\_dates, future\_close, color='#FFC107', linewidth=2, label='Predict')

ax1.fill\_between(future\_dates, future\_close, color='#FFC107', alpha=0.5)

ax1.set\_ylabel('PRICE', fontsize=12)

ax1.grid(True, linestyle='--', alpha=0.5)

ax1.legend(loc='upper left', fontsize=12)

colors = ['green' if data['Close'].iloc[i] >= data['Open'].iloc[i] else 'red' for i in range(len(data))]

ax2.bar(data.index, data['Volume'], color=colors, alpha=0.6, width=1.0)

predicted\_colors = ['purple' if i > 0 and future\_volume[i] < future\_volume[i - 1] else 'orange' for i in

range(len(future\_volume))]

ax2.bar(

future\_dates,

future\_volume,

color=predicted\_colors,

alpha=0.6,

width=1.0,

label='Volume Prediction'

)

ax2.set\_ylabel('VOLUME', fontsize=12)

ax2.set\_xlabel('TIME', fontsize=13)

ax2.grid(True, linestyle='--', alpha=0.5)

ax2.legend(loc='upper left', fontsize=12)

ax2.xaxis.set\_major\_formatter(mdates.DateFormatter('%Y-%m-%d'))

plt.xticks(rotation=45)

plt.draw()

Các dòng mã quan trọng:

* future\_close, future\_volume = predict\_rfnn(days\_to\_predict): Dự đoán giá và khối lượng tương lai.
* ax1.plot(...) và ax1.fill\_between(...): Vẽ đường giá hiện tại và dự đoán tương lai trên ax1.
* ax2.bar(...): Vẽ khối lượng giao dịch hiện tại và dự đoán tương lai trên ax2 với màu sắc khác nhau để phân biệt.

Cách thực thi: Hàm này được gọi khi người dùng nhấn các nút tương ứng (1 DAY, 1 WEEK, 1 MONTH, 1 YEAR). Nó cập nhật biểu đồ bằng cách vẽ lại cả hai trục với dữ liệu hiện tại và dự đoán.

* Hàm select\_file(): Chọn và xử lý file dữ liệu

def select\_file(event=None):

global current\_rmse\_annotation

Tk().withdraw()

file\_path = filedialog.askopenfilename(filetypes=[("Excel files", "\*.xlsx")])

if file\_path:

try:

process\_data(file\_path)

show\_current\_data()

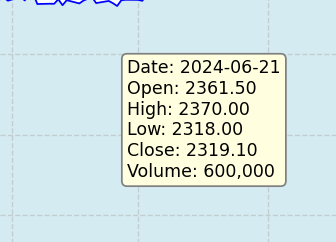
except Exception as e:

print(f"Lỗi khi xử lý file: {e}")

Khi người dùng nhấn nút "BROWSE", hàm này sẽ mở hộp thoại để chọn file Excel. Sau khi chọn file, dữ liệu được xử lý bằng process\_data(file\_path) và biểu đồ được vẽ lại bằng show\_current\_data().

ax\_button\_browse = plt.axes([0.05, 0.01, 0.1, 0.04])

button\_browse = Button(ax\_button\_browse, 'BROWSE', color='lightyellow', hovercolor='skyblue') button\_browse.on\_clicked(select\_file)

**Tooltip và Dự Báo Giá Vàng Tương Lai**

Hình 4: Tooltip và Dự Báo Giá Vàng Tương Lai

Khi người dùng di chuyển chuột qua các điểm dữ liệu trong biểu đồ (dữ liệu lịch sử hoặc dự báo tương lai), **tooltip** sẽ hiển thị thông tin chi tiết về giá và khối lượng giao dịch của ngày đó, giúp người dùng dễ dàng theo dõi các thay đổi.

* Định nghĩa biến toàn cục tooltip

tooltip = None

**Mô tả:** Biến tooltip được khai báo toàn cục để lưu trữ đối tượng tooltip hiện tại. Điều này cho phép xóa tooltip cũ khi di chuột di chuyển đến vị trí mới.

* Hàm update\_tooltip(event): Tạo và cập nhật Tooltip

def update\_tooltip(event):

global tooltip

if tooltip:

tooltip.remove()

tooltip = None

if event.inaxes == ax1 and event.xdata and event.ydata:

if event.xdata >= mdates.date2num(data.index[-1]):

closest\_index = min(range(len(future\_dates)),

key=lambda i: abs(mdates.date2num(future\_dates[i]) - event.xdata))

future\_date = future\_dates[closest\_index]

tooltip\_text = (

f"Date: {future\_date.strftime('%Y-%m-%d')}\n"

f"Open: {future\_close[closest\_index] \* 0.99:.2f}\n"

f"High: {future\_close[closest\_index]:.2f}\n"

f"Low: {future\_close[closest\_index] \* 0.98:.2f}\n"

f"Close: {future\_close[closest\_index]:.2f}\n"

f"Volume: {int(future\_volume[closest\_index]):,}"

)

else:

closest\_index = min(range(len(data.index)), key=lambda i: abs(mdates.date2num(data.index[i]) - event.xdata))

row = data.iloc[closest\_index]

tooltip\_text = (

f"Date: {row.name.strftime('%Y-%m-%d')}\n"

f"Open: {row['Open']:.2f}\n"

f"High: {row['High']:.2f}\n"

f"Low: {row['Low']:.2f}\n"

f"Close: {row['Close']:.2f}\n"

f"Volume: {int(row['Volume']):,}"

)

tooltip = ax1.annotate(

tooltip\_text,

(event.xdata, event.ydata),

xytext=(15, 15), textcoords='offset points',

bbox=dict(boxstyle="round,pad=0.3", edgecolor="gray", facecolor="lightyellow"),

fontsize=10, color="black"

)

plt.draw()

if event.inaxes == ax2:

if event.xdata and event.ydata:

closest\_index = min(range(len(data.index)), key=lambda i: abs(mdates.date2num(data.index[i]) - event.xdata))

row = data.iloc[closest\_index]

actual\_volume = row['Volume']

if event.xdata >= mdates.date2num(data.index[-1]):

future\_index = min(range(len(future\_dates)),

key=lambda i: abs(mdates.date2num(future\_dates[i]) - event.xdata))

predicted\_volume = future\_volume[future\_index]

rmse = calculate\_rmse([actual\_volume], [predicted\_volume])

tooltip\_text = f"Date: {future\_dates[future\_index].strftime('%Y-%m-%d')}\n" \

f"Volume: {predicted\_volume:.0f}\n" \

f"RMSE : {rmse:.2f}"

else:

tooltip\_text = f"Date: {row.name.strftime('%Y-%m-%d')}\n" \

f"Actual Volume: {int(actual\_volume):,}"

tooltip = ax2.annotate(

tooltip\_text,

(event.xdata, event.ydata),

xytext=(15, 15), textcoords='offset points',

bbox=dict(boxstyle="round,pad=0.3", edgecolor="gray", facecolor="lightyellow"),

fontsize=10, color="black"

)

plt.draw()

Xóa Tooltip Cũ: Nếu một tooltip hiện tại đang hiển thị (if tooltip:), nó sẽ được xóa trước khi tạo tooltip mới để tránh chồng lấp.

Kiểm Tra Vùng Axes: Hàm kiểm tra xem sự kiện di chuột (event) có xảy ra trong ax1 hay ax2.

* **Trên ax1:** Nếu di chuột trên trục giá (ax1), hàm sẽ xác định xem điểm di chuột nằm trong dữ liệu hiện tại hay trong dữ liệu dự đoán tương lai (future\_dates). Sau đó, nó sẽ tạo nội dung tooltip tương ứng và sử dụng ax1.annotate để hiển thị tooltip.
* **Trên ax2:** Nếu di chuột trên trục khối lượng (ax2), hàm sẽ xác định điểm gần nhất và hiển thị thông tin khối lượng thực tế hoặc dự đoán, cùng với RMSE nếu là dữ liệu dự đoán.

Tạo Tooltip: Sử dụng annotate để tạo hộp tooltip với nội dung được xác định và vị trí dựa trên sự kiện di chuột. Sau đó gọi plt.draw() để cập nhật biểu đồ.

* Kết nối sự kiện với hàm update\_tooltip

fig.canvas.mpl\_connect('motion\_notify\_event', update\_tooltip)

Dòng mã này kết nối sự kiện di chuyển chuột (motion\_notify\_event) trên figure với hàm update\_tooltip. Mỗi khi người dùng di chuyển chuột trên biểu đồ, hàm update\_tooltip sẽ được gọi để kiểm tra vị trí của chuột và hiển thị tooltip tương ứng.

**Nút Browse:**

ax\_button\_browse = plt.axes([0.05, 0.01, 0.1, 0.04])

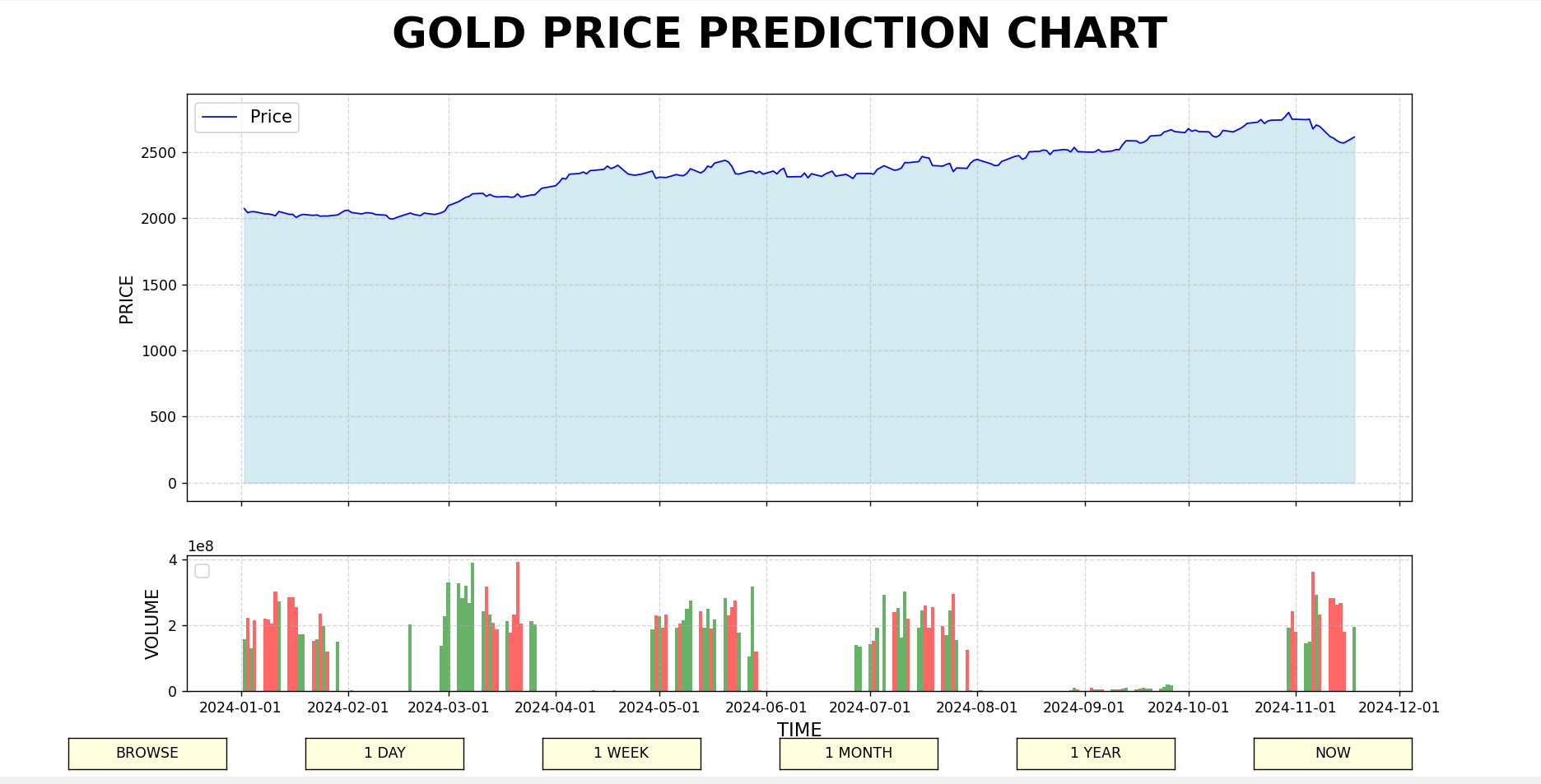
button\_browse = Button(ax\_button\_browse, 'BROWSE', color='lightyellow', hovercolor='skyblue') button\_browse.on\_clicked(select\_file)

* **Chức năng**: Nút này cho phép người dùng chọn file Excel chứa dữ liệu giá vàng và khối lượng giao dịch. Khi người dùng nhấn nút này, cửa sổ chọn file sẽ xuất hiện. Sau khi chọn file, dữ liệu sẽ được xử lý, và các biểu đồ sẽ được cập nhật để hiển thị dữ liệu đó.
* **Chi tiết**: Khi người dùng chọn một file hợp lệ (định dạng .xlsx), hàm process\_data(file\_path) sẽ được gọi để xử lý dữ liệu trong file Excel. Nếu file không hợp lệ hoặc có lỗi trong quá trình xử lý, chương trình sẽ thông báo lỗi.

**Nút Now:**

ax\_button\_now = plt.axes([0.8, 0.01, 0.1, 0.04])

button\_now = Button(ax\_button\_now, 'NOW', color='lightyellow', hovercolor='skyblue') button\_now.on\_clicked(lambda x: show\_current\_data())

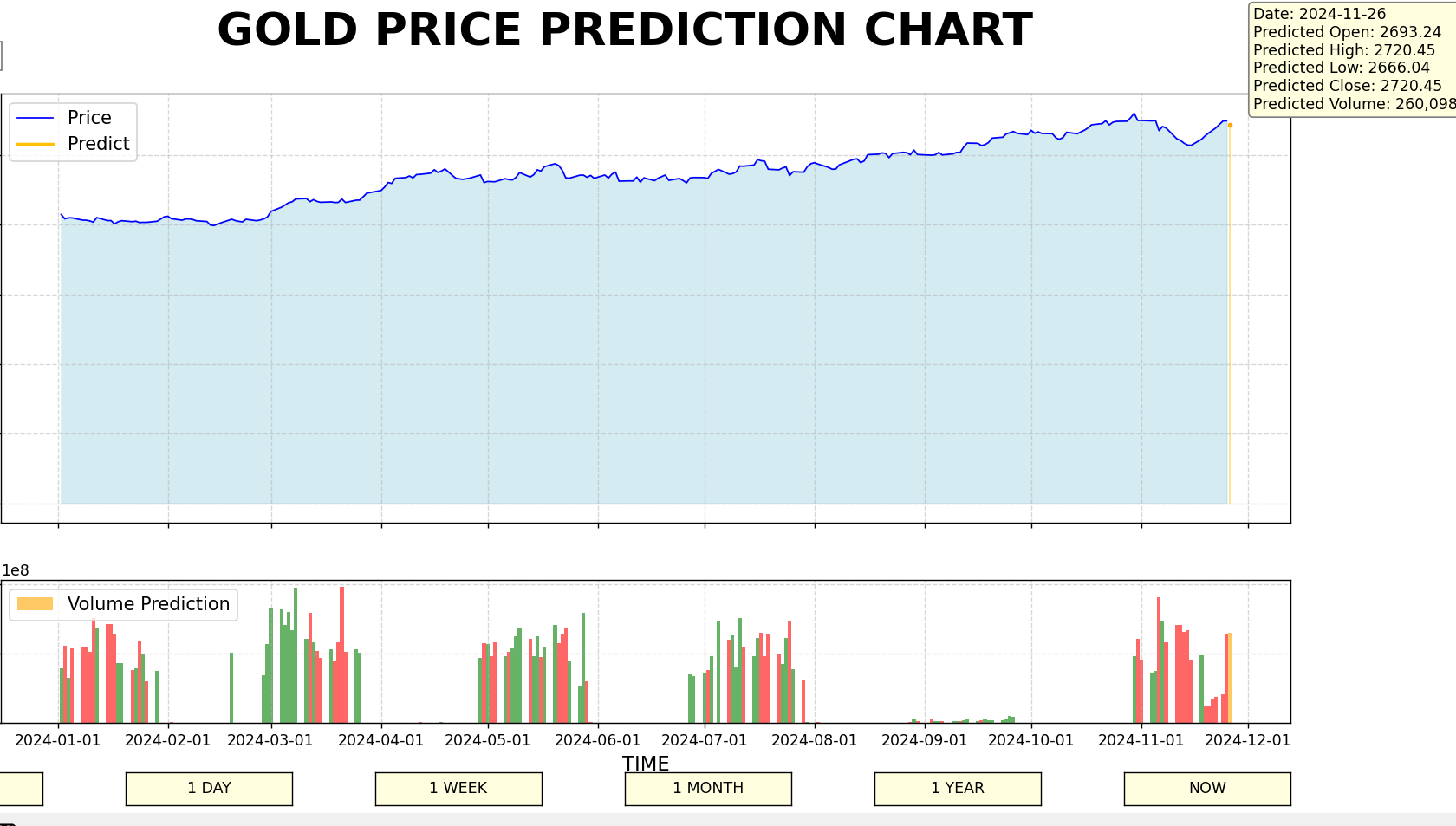
* **Chức năng**: Khi người dùng chọn **Now**, biểu đồ sẽ được cập nhật và hiển thị dữ liệu hiện tại, tức là dữ liệu giá vàng và khối lượng giao dịch của ngày hôm nay. Điều này sẽ không có dự đoán mà chỉ hiển thị dữ liệu thực tế đã thu thập được.
* **Chi tiết**: Khi nhấn nút **Now**, hàm show\_current\_data() sẽ được gọi để vẽ lại biểu đồ với dữ liệu thực tế của thị trường vàng trong quá khứ (bắt đầu từ năm 2024).

Hình 5: Chức năng các nút

**Nút 1 DAY:**

ax\_button\_day = plt.axes([0.2, 0.01, 0.1, 0.04])

button\_day = Button(ax\_button\_day, '1 DAY', color='lightyellow', hovercolor='skyblue') button\_day.on\_clicked(lambda x: update\_plot('day'))



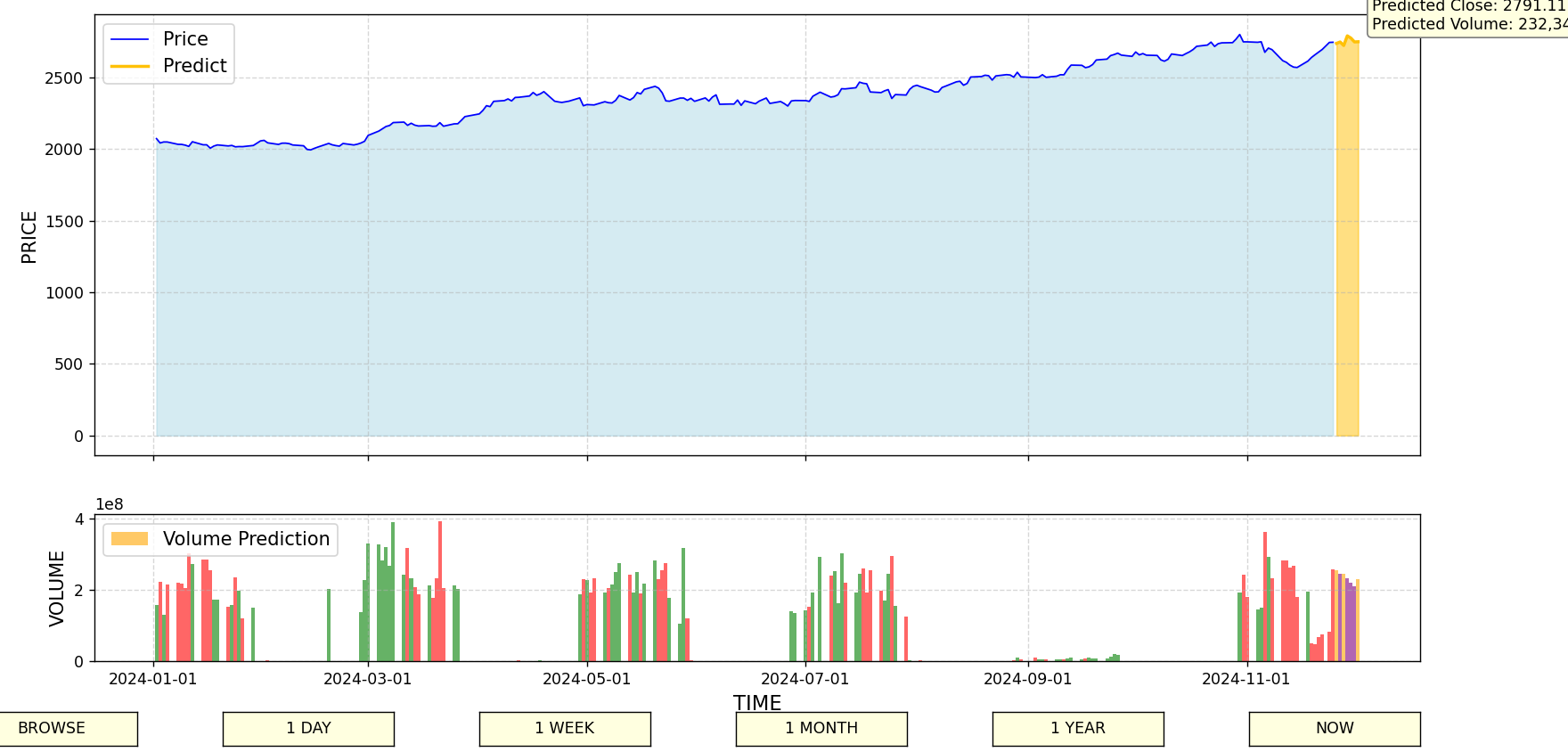
Hình 6: Mô hình dự đoán 1 ngày

* **Chức năng**: Khi người dùng chọn **1 DAY**, biểu đồ sẽ được cập nhật để hiển thị dự đoán giá vàng và khối lượng giao dịch trong **1 ngày** tới. Hàm update\_plot(interval='day') sẽ được gọi với tham số 'day', điều này sẽ khiến chương trình tính toán và hiển thị dữ liệu dự báo cho một ngày sắp tới.
* **Chi tiết**: Dữ liệu dự báo được tạo ra thông qua mô hình **RFNN** (Recurrent Fuzzy Neural Network) với mục đích dự đoán giá trị giá vàng và khối lượng giao dịch cho 1 ngày sau.

**Nút 1 WEEK:**

ax\_button\_week = plt.axes([0.35, 0.01, 0.1, 0.04])

button\_week = Button(ax\_button\_week, '1 WEEK', color='lightyellow', hovercolor='skyblue') button\_week.on\_clicked(lambda x: update\_plot('week'))



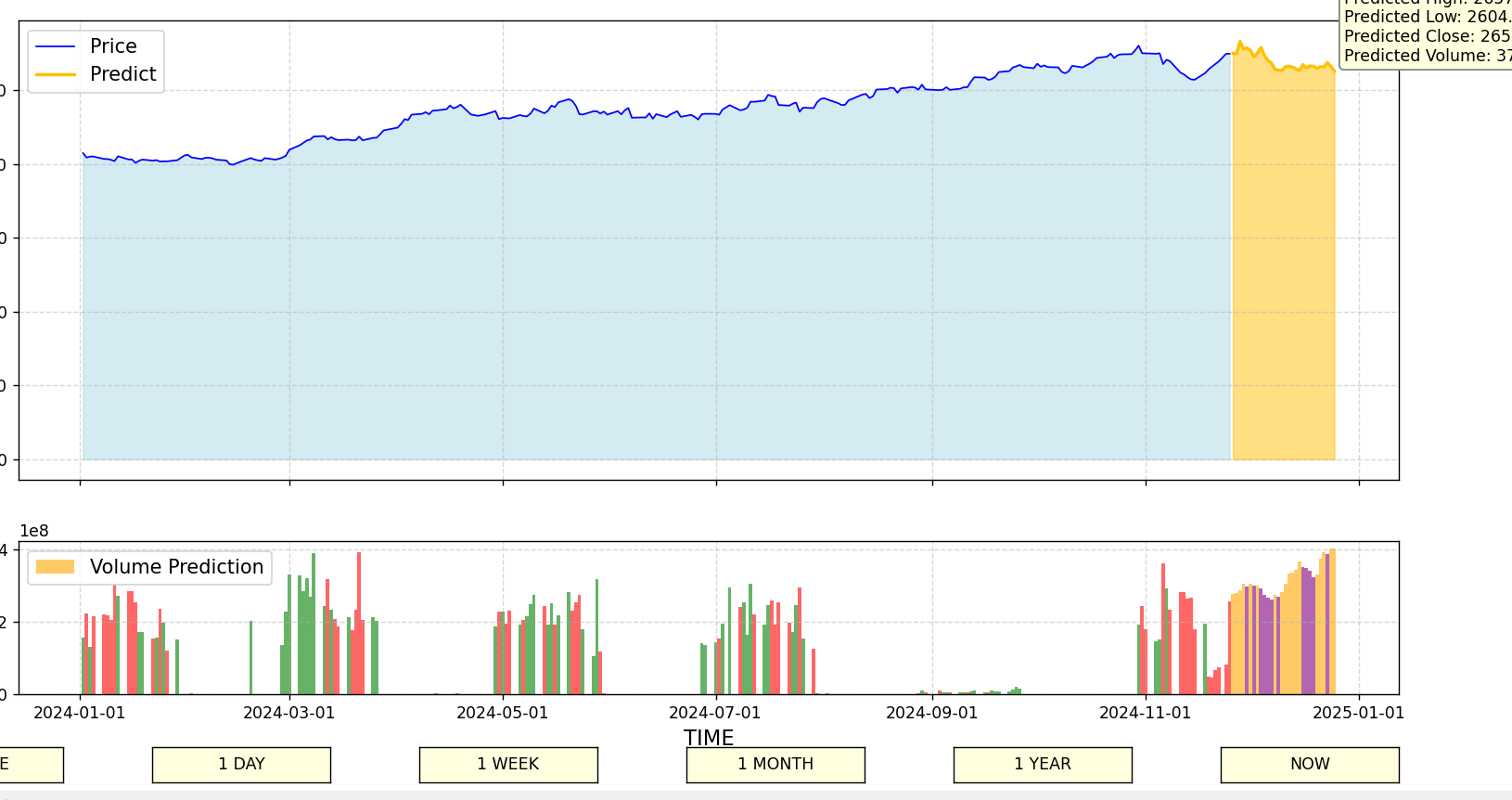
Hình 7: Mô hình dự đoán 1 tuần

* **Chức năng**: Khi chọn **1 WEEK**, chương trình sẽ dự đoán giá vàng và khối lượng giao dịch trong **1 tuần** tới.
* **Chi tiết**: Tương tự như nút **1 DAY**, hàm update\_plot(interval='week') sẽ được gọi, và tham số 'week' sẽ giúp chương trình tính toán các dự đoán trong vòng 7 ngày tới.

**Nút 1 MONTH:**

ax\_button\_month = plt.axes([0.5, 0.01, 0.1, 0.04])

button\_month = Button(ax\_button\_month, '1 MONTH', color='lightyellow', hovercolor='skyblue') button\_month.on\_clicked(lambda x: update\_plot('month'))



Hình 8: Mô hình dự đoán 1 tháng

* **Chức năng**: Khi người dùng chọn **1 MONTH**, biểu đồ sẽ hiển thị dự đoán giá vàng và khối lượng giao dịch trong **1 tháng** tới.
* **Chi tiết**: Chức năng này sẽ gọi hàm update\_plot(interval='month'), với tham số 'month' để tính toán dự đoán cho 30 ngày tiếp theo. Đây là khoảng thời gian dài hơn, và dự đoán có thể thay đổi theo độ chính xác của mô hình.

**Nút 1 YEAR:**

ax\_button\_year = plt.axes([0.65, 0.01, 0.1, 0.04])

button\_year = Button(ax\_button\_year, '1 YEAR', color='lightyellow', hovercolor='skyblue') button\_year.on\_clicked(lambda x: update\_plot('year'))

**A chart showing different colored lines

Description automatically generated**

Hình 9: Mô hình dự đoán 1 năm

* **Chức năng**: Khi chọn **1 YEAR**, biểu đồ sẽ hiển thị dự đoán giá vàng và khối lượng giao dịch trong **1 năm** tới.
* **Chi tiết**: Nút này sẽ gọi hàm update\_plot(interval='year'), và tham số 'year' sẽ khiến chương trình tính toán dự đoán cho 365 ngày tiếp theo. Đây là dự đoán dài hạn, có thể ít chính xác hơn do ảnh hưởng của nhiều yếu tố không thể dự báo trước.

**Ứng dụng RMSE trong phần mềm dự đoán giá vàng**

**A graph of a graph

Description automatically generated with medium confidence**

Hình 10: Mô hình dự đoán RMSE

Trong phần mềm dự đoán giá vàng, RMSE sẽ được sử dụng để đo lường sự chính xác của các dự báo giá vàng và khối lượng giao dịch. Cụ thể, sau khi mô hình học (ở đây sử dụng **RandomForestRegressor**) dự báo giá vàng và khối lượng giao dịch trong tương lai, chúng ta sẽ tính toán sai số giữa giá trị thực tế và giá trị dự báo từ mô hình. Mức độ chính xác của dự báo sẽ được thể hiện thông qua giá trị **RMSE**.

**Cách RMSE hoạt động trong phần mềm:**

def calculate\_rmse(actual, predicted):

actual = np.array(actual)

predicted = np.array(predicted)

mse = np.mean((actual - predicted) \*\* 2)

return np.sqrt(mse)

Hàm này nhận vào hai danh sách actual (giá trị thực tế) và predicted (giá trị dự đoán).

Nó tính toán **Mean Squared Error (MSE)** bằng cách lấy trung bình của bình phương sự khác biệt giữa các giá trị thực tế và dự đoán.

Cuối cùng, hàm trả về căn bậc hai của MSE, tức là **RMSE**.

* Ứng Dụng RMSE Trong Hàm update\_tooltip:

Phần sử dụng RMSE nằm trong hàm update\_tooltip(event), nơi mà tooltip được hiển thị khi người dùng di chuyển chuột trên biểu đồ. Cụ thể, RMSE được tính toán và hiển thị khi người dùng di chuột trên trục ax2 (trục khối lượng giao dịch) và vị trí chuột nằm trong phần dự đoán tương lai.

a. Vị Trí Tính RMSE:

if event.inaxes == ax2:

if event.xdata and event.ydata:

closest\_index = min(range(len(data.index)), key=lambda i: abs(mdates.date2num(data.index[i]) - event.xdata))

row = data.iloc[closest\_index]

actual\_volume = row['Volume']

if event.xdata >= mdates.date2num(data.index[-1]):

future\_index = min(range(len(future\_dates)),

key=lambda i: abs(mdates.date2num(future\_dates[i]) - event.xdata))

predicted\_volume = future\_volume[future\_index]

rmse = calculate\_rmse([actual\_volume], [predicted\_volume])

tooltip\_text = f"Date: {future\_dates[future\_index].strftime('%Y-%m-%d')}\n" \

f"Volume: {predicted\_volume:.0f}\n" \

f"RMSE : {rmse:.2f}"

else:

tooltip\_text = f"Date: {row.name.strftime('%Y-%m-%d')}\n" \

f"Actual Volume: {int(actual\_volume):,}"

Kiểm Tra Vị Trí Chuột: Đầu tiên, hàm kiểm tra xem sự kiện chuột (event) có nằm trong trục ax2 (trục khối lượng) hay không.

* Xác Định Chỉ Số Gần Nhất:

Sử dụng min và hàm lambda để tìm chỉ số (closest\_index) của điểm dữ liệu gần nhất với vị trí chuột trên trục ax2.

Lấy actual\_volume từ dữ liệu thực tế tại chỉ số này.

* Kiểm Tra Vị Trí Dự Báo Tương Lai:

Nếu vị trí chuột (event.xdata) nằm trong phần dự báo tương lai (event.xdata >= mdates.date2num(data.index[-1])), hàm sẽ:

Tìm future\_index gần nhất với vị trí chuột.

Lấy predicted\_volume từ danh sách future\_volume tại future\_index.

Tính RMSE: Gọi calculate\_rmse với actual\_volume và predicted\_volume

rmse = calculate\_rmse([actual\_volume], [predicted\_volume])

Tạo Nội Dung Tooltip: Hiển thị ngày dự báo, khối lượng dự báo và giá trị RMSE.

Nếu vị trí chuột không nằm trong phần dự báo tương lai, chỉ hiển thị thông tin về khối lượng thực tế.

b. Hiển Thị Tooltip Với RMSE:

tooltip = ax2.annotate(

tooltip\_text,

(event.xdata, event.ydata),

xytext=(15, 15), textcoords='offset points',

bbox=dict(boxstyle="round,pad=0.3", edgecolor="gray", facecolor="lightyellow"),

fontsize=10, color="black"

)

plt.draw()

annotate: Tạo một hộp văn bản (tooltip) tại vị trí chuột với nội dung được định nghĩa trong tooltip\_text.

Nội Dung Tooltip: Khi chuột nằm trên phần dự báo tương lai, tooltip sẽ hiển thị:

Ngày dự báo.

Khối lượng dự báo.

RMSE: Đánh giá mức độ sai số giữa giá trị thực tế và dự báo.

plt.draw(): Cập nhật biểu đồ để hiển thị tooltip mới.

# CHƯƠNG 4. KẾT LUẬN

1. **Kết quả đạt được:**

Nhóm chúng em đã nghiên cứu và hiểu rõ thuật toán **RFNN (Recurrent Fuzzy Neural Network)**, sau đó áp dụng thành công vào việc dự đoán giá vàng. Bằng cách thu thập và xử lý dữ liệu lịch sử một cách hiệu quả, chúng em đã phát triển mô hình RFNN có khả năng dự báo chính xác giá vàng với chỉ số **RMSE** thấp, chứng minh hiệu quả của thuật toán trong việc phân tích và dự đoán biến động thị trường tài chính. Ngoài ra, nhóm còn học hỏi được cách tối ưu hóa các tham số mô hình, xử lý dữ liệu phức tạp, và phát triển giao diện trực quan bằng matplotlib, cho phép người dùng tương tác và xem kết quả dự báo một cách dễ dàng. Những kết quả này không chỉ giúp hoàn thành dự án mà còn nâng cao kỹ năng về học máy, phân tích dữ liệu và phát triển ứng dụng trực quan cho các dự án tương lai.

1. **Đề xuất và nghiên cứu:**

**Cải tiến mô hình RFNN**:

* Nghiên cứu thêm về các hàm mờ khác để cải thiện độ chính xác của mô hình.
* Thử nghiệm các phương pháp khác nhau để lựa chọn các tham số mờ và số lượng nút ẩn.

**Ứng dụng thực tiễn**:

* Phát triển phần mềm hoặc hệ thống dự báo giá vàng dựa trên mô hình RFNN cho các nhà đầu tư và chuyên gia tài chính.
* Áp dụng mô hình RFNN vào các thị trường tài chính khác để dự báo các biến động trong thị trường chứng khoán, tỷ giá ngoại tệ.

1. **Kết luận**

Qua quá trình nghiên cứu và áp dụng thuật toán RFNN (Recurrent Fuzzy Neural Network), nhóm chúng em đã nắm vững các kiến thức cơ bản về cấu trúc và cách thức hoạt động của mô hình này. Việc tích hợp RFNN vào hệ thống dự báo giá vàng thông qua công thức 4 lớp trong mã nguồn đã giúp chúng em hiểu sâu hơn về quy trình triển khai thực tiễn của mô hình, từ tiền xử lý dữ liệu đến tối ưu hóa các siêu tham số. Kết quả đạt được chứng minh rằng RFNN không chỉ cải thiện đáng kể độ chính xác của dự báo mà còn tăng tính linh hoạt khi ứng dụng vào các tình huống thị trường khác nhau. Bên cạnh đó, nhóm chúng em còn học hỏi được nhiều kỹ năng quan trọng về xử lý dữ liệu, phát triển mô hình học máy và tạo giao diện người dùng trực quan, góp phần nâng cao hiệu quả và trải nghiệm người dùng của phần mềm dự báo. Những thành tựu này không chỉ giúp hoàn thành dự án mà còn cung cấp nền tảng vững chắc cho các nghiên cứu và ứng dụng tiếp theo trong lĩnh vực học máy và phân tích dữ liệu tài chính.

# TÀI LIỆU THAM KHẢO

1. **SÁCH :**

**[1]** **Dự báo giá vàng nhìn từ mô hình hồi quy tuyến tính và mô hình học máy.** https://nsti.vista.gov.vn/publication/download/hE/qFIDhEZhnsXZh.html

**[2]** **Sử dụng RFNN để dự báo giá vàng.** http://www.nsl.hcmus.edu.vn/greenstone/collect/hnkhbk/index/assoc/HASH0162.dir/doc.pdf

**[3]** **Liu, Y., & Li, Y.** (2015). "Forecasting Financial Time Series Using Fuzzy Neural Networks," International Journal of Computational Intelligence Systems, 8(1), 1–9.

**[4]** **Jang, J.-S. R., Sun, C.-T., & Mizutani, E.** (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall.

1. **NGUỒN TỪ INTERNET VÀ WEBSITE :**

**[1]** [**https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-recurrent-neural-networks**](https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-recurrent-neural-networks)

**[2]** [**https://www.sciencedirect.com/science/article/abs/pii/S0925231222010074**](https://www.sciencedirect.com/science/article/abs/pii/S0925231222010074)

**[3]** [**Investing.com - Stock Market Quotes & Financial News**](https://www.investing.com/)