

# Dự báo giá của tiền điện tử bằng phương pháp chuỗi thời gian

**Tóm tắt nội dung**—Nghiên cứu này sử dụng một loạt các mô hình dự báo để phân tích và dự đoán tỷ giá trao đổi của Binance Coin (BNB), Bitcoin (BTC), và Ethereum (ETH) so với USD từ năm 2019 đến 2024. Bằng cách sử dụng các mô hình như TimesNet, Random Forest, CNN-LSTM, Bagging Model, và VAR để dự báo các biến động, cùng với hồi quy tuyến tính, nghiên cứu của chúng tôi nhằm cung cấp những thông tin quý báu cho các nhà đầu tư, các nhà phân tích tài chính, và những nhà quyết định chính sách trong việc điều hướng thị trường tiền điện tử dao động mạnh mẽ.

**Index Terms**—Chuỗi thời gian, phương pháp thống kê, tỷ giá trao đổi tiền điện tử, học máy, học sâu

## I. GIỚI THIỆU

Trong thời đại hiện nay, thị trường tiền điện tử đã trở thành một phần không thể thiếu của hệ thống tài chính toàn cầu, và việc theo dõi và dự báo tỷ giá trao đổi của tiền điện tử so với USD là vô cùng quan trọng. Trong phạm vi này, nhóm tập trung vào ba loại tiền điện tử phổ biến nhất: Binance Coin (BNB), Bitcoin (BTC), và Ethereum (ETH).

Nghiên cứu này sẽ sử dụng một loạt các mô hình dự báo để hiểu và dự đoán các biến động trong tỷ giá trao đổi của các loại tiền điện tử này so với USD. Cụ thể, nhóm sẽ sử dụng các mô hình hồi quy tuyến tính để phân tích các xu hướng dài hạn, và sau đó sử dụng các mạng nơ-ron như TimesNet và CNN-LSTM để khám phá mối quan hệ phức tạp giữa các yếu tố và dự báo các biến động ngắn hạn.

Ngoài ra, nhóm cũng sẽ áp dụng các mô hình học máy như Random Forest, XGBoost, và VAR (Vector Autoregression) để đảm bảo tính linh hoạt và độ chính xác trong việc dự báo. Sự kết hợp của những mô hình này sẽ cung cấp một cái nhìn toàn diện về các xu hướng và biến động trong tỷ giá trao đổi của tiền điện tử so với USD, giúp các nhà đầu tư và chuyên gia tài chính đưa ra quyết định đầu tư thông minh và hiệu quả.

## II. NGHIÊN CỨU LIÊN QUAN

Có một số nghiên cứu đã được thực hiện về chủ đề này:

Yecheng Yao và cộng sự [1] đã đề xuất một cách tiếp cận học sâu để dự đoán giá của tiền điện tử. Bài báo này tập trung vào việc sử dụng Long Short-Term Memory (LSTM) và Recurrent Neural Network (RNN) để phân tích dự báo giá của tiền điện tử. Các yếu tố được xem xét bao gồm vốn hóa thị trường, khối lượng giao dịch, nguồn cung lưu thông và nguồn cung tối đa. Tác giả cũng đề cập đến một số yếu tố như môi trường chính trị và quy định của con người.

Muhammad Ali Nasir và cộng sự [2] đã đề xuất một mô hình trong đó lợi suất và khối lượng của tiền điện tử được dự đoán bằng cách sử dụng các công cụ tìm kiếm. Một tập dữ

liệu hàng tuần từ năm 2013 đến 2017 được sử dụng và thu thập cấu trúc phụ thuộc bằng cách sử dụng các phương pháp kinh nghiệm như VAR framework, một phương pháp copulas vv.

Aggarwal A. và cộng sự [3] đã thảo luận về các thông số khác nhau ảnh hưởng đến dự đoán giá Bitcoin dựa trên Root Mean Square Error (RMSE) sử dụng các mô hình học sâu như Convolutional Neural Network (CNN), Long ShortTerm Memory (LSTM) và Gated Recurrent Unit (GRU).

Alessandretti, L., ElBahrawy, A., Aiello, L.M. và Baronchelli, A. [4] đã kiểm tra hiệu suất của ba mô hình trong việc dự đoán giá tiền điện tử hàng ngày cho 1681 loại tiền điện tử. Hai trong số các mô hình dựa trên cây quyết định tăng cường độ dốc và một dựa trên Long ShortTerm memory. Trong tất cả các trường hợp, các danh mục đầu tư được xây dựng dựa trên các dự đoán và hiệu suất được so sánh dựa trên lợi nhuận từ đầu tư.

Mittal R. và cộng sự [5] đã dự đoán giá của tiền điện tử dựa trên giá mở, giá thấp và giá cao của chúng. Một số thuật toán Machine Learning được sử dụng để dự đoán các thay đổi giá hàng ngày của tiền điện tử. Bộ dữ liệu được sử dụng trong bài báo này là tương đối nhỏ. Hồi quy tuyến tính đa biến đã được sử dụng để dự đoán giá cao nhất và thấp nhất của tiền điện tử.

Suhwanji và cộng sự [6] đã phát triển và so sánh các mô hình dự đoán giá Bitcoin dựa trên thông tin blockchain Bitcoin. Cụ thể, họ đã kiểm tra các mô hình học sâu tiên tiến như mạng nơ-ron sâu (DNN), mô hình long short-term memory (LSTM), mạng nơ-ron tích chập (CNN), mạng nơ-ron còn sót (ResNet) và các sự kết hợp của chúng. Đối với các vấn đề hồi quy, LSTM đạt hiệu suất tốt hơn so với các mô hình khác, trong khi đối với các vấn đề phân loại, DNN đạt hiệu suất tốt hơn một chút.

Phumudzo Lloyd Seabe và cộng sự [7] kết luận rằng ba loại kỹ thuật học sâu - LSTM, GRU và Bi-LSTM - đã được sử dụng để dự đoán giá của ba loại tiền điện tử lớn nhất, được đo bằng vốn hóa thị trường của chúng: Bitcoin, Ethereum và Litecoin. Kết quả của nghiên cứu cho thấy rằng mô hình Bi-LSTM cung cấp các dự đoán chính xác nhất cho cả ba loại tiền tệ, tiếp theo là mô hình GRU.

Nguyen Dinh Thuan và cộng sự [8] kết luận rằng việc dự đoán giá của tiền điện tử là một công việc khó khăn, đòi hỏi nghiên cứu sâu sắc và toàn diện về thị trường tiền điện tử. Ngoài ra, nó cũng cần sự hỗ trợ của máy học và mô hình thống kê. Trong việc dự đoán giá của loại tiền này vào ngày mai. Sự kết hợp của mô hình hoặc mô hình hỗn hợp dẫn đến mô hình dự đoán với chất lượng hiệu quả hơn, như là kết quả

cao hơn, tỷ lệ lỗi thấp trong chi tiết.

### III. TÀI NGUYÊN

#### A. Bộ dữ liệu

Tập dữ liệu được sử dụng trong nghiên cứu này được thu thập từ <https://www.investing.com/>, một nền tảng tài chính uy tín nổi tiếng với thông tin thị trường toàn diện và cập nhật. Tập dữ liệu này bao gồm các giá trị của ba loại tiền điện tử phổ biến nhất trên thế giới: Binance Coin, Ethereum và Bitcoin so với đồng đô la Mỹ từ ngày 01/03/2019 đến ngày 01/06/2024, cung cấp một phạm vi thời gian mạnh mẽ cho phân tích của nhóm.

Mỗi mục nhập trong tập dữ liệu bao gồm các chỉ số tài chính quan trọng:

**Date:** Thuộc tính thể hiện ngày cập nhật dữ liệu.

**Open:** Thuộc tính đại diện cho giá mở cửa của tiền điện tử tương ứng cho ngày đó, tức là giá của tiền điện tử khi thị trường mở cửa để giao dịch.

**High:** Thuộc tính biểu thị giá cao nhất mà loại tiền điện tử tương ứng đạt được trong ngày đó.

**Low:** Thuộc tính biểu thị giá thấp nhất mà loại tiền điện tử tương ứng đạt được trong ngày đó.

**Close:** Thuộc tính biểu thị giá đóng cửa của loại tiền điện tử tương ứng cho ngày đó, tức là giá của tiền điện tử khi thị trường đóng cửa giao dịch.

**Adj Close:** Thuộc tính giá đóng cửa đã điều chỉnh được sử dụng để tính toán các điều chỉnh như chia cổ tức, phát hành cổ phiếu mới, cổ tức cổ phiếu, v.v. Đối với một số loại tiền điện tử, giá đóng cửa đã điều chỉnh và giá đóng cửa có thể rất tương tự hoặc giống nhau.

**Volumne:** Thuộc tính biểu thị tổng khối lượng giao dịch của loại tiền điện tử tương ứng trong ngày đó.

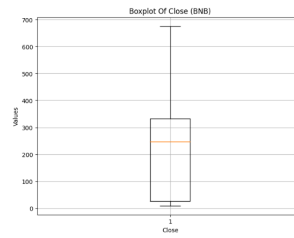
Vì mục tiêu là dự báo giá đóng cửa, nhóm chỉ sử dụng dữ liệu liên quan đến cột "Close" (USD) sẽ được xử lý.

Bằng cách sử dụng dữ liệu có sẵn trên nền tảng này, nhóm đảm bảo một nền tảng đáng tin cậy cho phân tích.

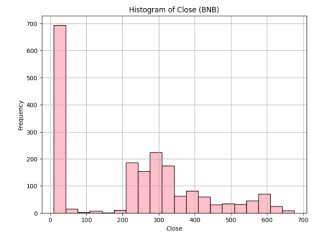
#### B. Mô tả thống kê

Bảng I  
MÔ TẢ THỐNG KÊ DỮ LIỆU BNB, BTC, ETH

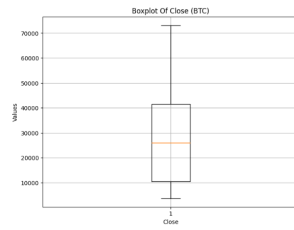
	BNB	BTC	ETH
Count	1920	1920	1920
Mean	229.615657	27933.663833	1581.028056
Std	184.419311	17923.062895	1205.741249
Min	9.386050	3761.557129	110.605873
25%	27.154293	10457.237549	269.458023
50%	247.572518	25935.820312	1622.698242
75%	332.063485	41462.885742	2341.759216
Max	675.684082	73083.5	4812.087402



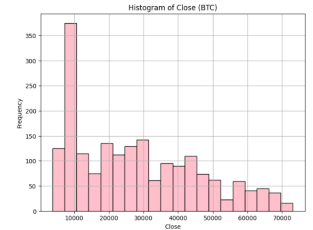
Hình 1. Biểu đồ Boxplot giá trị Close của Binance Coin(BNB)



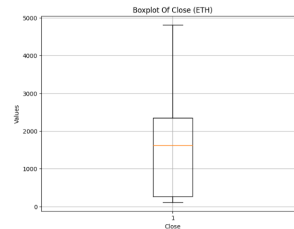
Hình 2. Biểu đồ Histogram giá trị Close của Binance Coin(BNB)



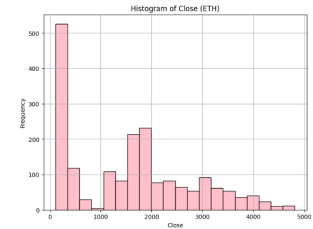
Hình 3. Biểu đồ Boxplot giá trị Close của Bitcoin (BTC)



Hình 4. Biểu đồ Histogram giá trị Close của Bitcoin (BTC)



Hình 5. Biểu đồ Boxplot giá trị Close của Ethereum (ETH)



Hình 6. Biểu đồ Histogram giá trị Close của Ethereum (ETH)

#### C. Công cụ

Trong quá trình nghiên cứu, nhóm nhận thấy rằng có nhiều công cụ và thư viện Python phổ biến được sử dụng cho phân tích dữ liệu, hỗ trợ học sâu và trực quan hóa dữ liệu. Sau khi nghiên cứu và lựa chọn, nhóm quyết định sử dụng một số công cụ chính: numpy, pandas, sklearn, matplotlib.pyplot. Việc sử dụng những công cụ này đã giúp nhóm xác định dữ liệu, hiểu sâu hơn về ý nghĩa của dữ liệu. Đồng thời, khả năng trực quan hóa dữ liệu đã hỗ trợ đáng kể quá trình hiểu rõ chi tiết cũng như mô tả các bộ dữ liệu một cách rõ ràng, mang lại cái nhìn tổng quan rộng hơn trong việc khám phá dữ liệu.

#### D. Tỷ lệ dữ liệu

Trong quá trình phân tích dữ liệu chuỗi thời gian, nhóm chia tập dữ liệu thành các tập huấn luyện và kiểm tra bằng các tỷ lệ khác nhau: 70% cho huấn luyện và 30% cho kiểm tra, 80% cho huấn luyện và 20% cho kiểm tra, và 90% cho huấn luyện và 10% cho kiểm tra. Các tỷ lệ này giúp nhóm xem xét cách mà hiệu suất của mô hình bị ảnh hưởng bởi phân phối dữ liệu trong mỗi tập.

Tỷ lệ thông thường sử dụng là 7:3, dành 70% cho huấn luyện và 30% cho kiểm tra, tạo ra một sự cân bằng giữa việc cung cấp đủ dữ liệu huấn luyện và đảm bảo các tập dữ liệu khác biệt để điều chỉnh tinh chỉnh và đánh giá. Một lựa chọn khác là tỷ lệ 8:2, ưa chuộng tập huấn luyện 80%, có lợi cho các mô hình phức tạp đòi hỏi một tập dữ liệu huấn luyện lớn hơn. Trong một số trường hợp, một cách tiếp cận cẩn thận như tỷ lệ 9:1 có thể được ưu tiên, đặc biệt là với một tập dữ liệu lớn và một mô hình đơn giản. Tỷ lệ này đảm bảo đủ dữ liệu huấn luyện trong khi cung cấp một tập kiểm tra đáng kể để đánh giá hiệu suất.

#### E. Độ đo đánh giá mô hình

**RMSE:** Sai số bình phương trung bình căn, đại diện cho căn bậc hai của sai số trung bình bình phương trong các giá trị dự đoán. Cơ bản, nó đo lường sự chênh lệch giữa các giá trị dự đoán và thực tế. Các giá trị RMSE thấp hơn cho thấy các mô hình dự đoán xuất sắc hơn.

**MAPE:** Là một phương pháp được sử dụng để đánh giá độ chính xác của một mô hình dự báo hoặc dự đoán. Nó tính toán sự lệch phần trăm trung bình giữa các giá trị dự đoán và thực tế, cung cấp một phép đo về hiệu suất của mô hình dựa trên sai số phần trăm.

**MSE:** Sai số bình phương trung bình, tính toán giá trị trung bình của sự khác biệt bình phương giữa các giá trị dự đoán và thực tế trong một mô hình. Nó cung cấp một phép đo về độ chính xác tổng thể bằng cách đo lường sự lớn mạnh của các sai số.

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

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

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

Trong đó:

$y_i$  is the observer value,

$\hat{y}_i$  is the predicted value,

$n$  is the number of observers

## IV. PHƯƠNG PHÁP LUẬN

### A. RNN

Mạng RNN (Recurrent Neural Network) là một dạng của mạng nơ-ron nhân tạo được thiết kế đặc biệt để xử lý dữ liệu chuỗi, trong đó mỗi dữ liệu đầu vào không chỉ phụ thuộc vào dữ liệu hiện tại mà còn phụ thuộc vào dữ liệu đã được xử lý trước đó trong chuỗi. Điều này giúp RNN hiểu và ánh xạ các mối quan hệ phức tạp giữa các thành phần trong chuỗi, như dữ liệu thời gian, ngôn ngữ tự nhiên, v.v.

### B. CCN-LSTM

Mạng CNN-LSTM là một kiến trúc mạng nơ-ron sử dụng cả Convolutional Neural Network (CNN) và Long Short-Term Memory (LSTM). Mạng này kết hợp các lợi ích của cả hai loại mạng để xử lý dữ liệu chuỗi 2D hoặc 3D như hình ảnh hoặc video.

#### Convolutional Neural Network (CNN):

CNN thường được sử dụng để xử lý dữ liệu không gian như hình ảnh. Chúng sử dụng các lớp tích chập để trích xuất đặc trưng từ dữ liệu không gian bằng cách di chuyển các bộ lọc qua toàn bộ hình ảnh. Các lớp gộp được sử dụng để giảm kích thước của đặc trưng và tăng tính tổng quát của mô hình.

#### Long Short-Term Memory (LSTM):

LSTM thường được sử dụng để xử lý dữ liệu chuỗi thời gian như văn bản hoặc âm nhạc. Chúng có khả năng ghi nhớ thông tin từ quá khứ và lưu giữ nó trong trạng thái ẩn, cho phép mô hình hiểu các mối quan hệ phức tạp trong dữ liệu chuỗi.

#### Kết hợp CNN và LSTM:

Mạng CNN-LSTM kết hợp CNN để trích xuất đặc trưng không gian từ dữ liệu (ví dụ: các khối hình ảnh) và LSTM để hiểu mối quan hệ thời gian trong dữ liệu (ví dụ: chuỗi hình ảnh). Quá trình này cho phép mô hình hiểu được cả cấu trúc không gian và thời gian của dữ liệu, phù hợp cho nhiều ứng dụng như nhận dạng hình ảnh/video và dự báo chuỗi thời gian. Mạng CNN-LSTM thường được sử dụng trong các lĩnh vực như xử lý hình ảnh/video, nhận dạng cử chỉ, dự báo thời tiết, dự báo chuỗi thời gian trong tài chính, và nhiều ứng dụng khác liên quan đến dữ liệu có cấu trúc không gian và thời gian.

### C. ARIMA - autoregressive integrated moving average (THUẬT TOÁN TRUNG BÌNH TRƯỢT KẾT HỢP TỰ HỒI QUY)

ARIMA, thường được gọi là phương pháp luận Box-Jenkins, là một mô hình kết hợp tự hồi quy (AR) và trung bình trượt (MA) với quá trình tích hợp (I) để dự báo dữ liệu chuỗi thời gian.

**1. Auto regression (AR):** AR là thành phần tự hồi quy sử dụng các giá trị quá khứ của chính chuỗi thời gian để dự báo giá trị hiện tại. Mô hình AR(p) được biểu diễn như sau:

$$AR(p) = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} \quad (1)$$

**2. Moving Average (MA):** MA là thành phần trung bình trượt, sử dụng các giá trị nhiễu trắng (white noise) quá khứ để dự báo giá trị hiện tại. Mô hình MA(q) được biểu diễn như sau:

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

**3. Integrated (I):** Thành phần tích hợp (I) giúp biến đổi chuỗi thời gian thành chuỗi dừng bằng cách lấy sai phân. Quá trình sai phân bậc  $d$  được thực hiện như sau:

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

Phương trình hồi quy ARIMA( $p, d, q$ ) có thể được biểu diễn dưới dạng:

$$\Delta^d(x_t) = \phi_1 \Delta^d(x_{t-1}) + \phi_2 \Delta^d(x_{t-2}) + \dots + \phi_p \Delta^d(x_{t-p}) + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} \quad (3)$$

Tóm lại, ARIMA là mô hình kết hợp của quá trình tự hồi quy và trung bình trượt, sử dụng dữ liệu quá khứ để dự báo tương lai. Trước khi huấn luyện mô hình, chuỗi thời gian cần được chuyển hóa thành chuỗi dừng bằng cách lấy sai phân hoặc biến đổi logarit.

#### D. Random Forest

1) *Lý thuyết:* Random Forest hoạt động như sau:

- Bootstrap Sampling:** Mỗi cây quyết định được huấn luyện trên một tập dữ liệu con được lấy mẫu từ tập dữ liệu huấn luyện ban đầu thông qua phương pháp tái chọn mẫu Bootstrap.
- Random Feature Selection:** Trong quá trình xây dựng mỗi cây quyết định, chỉ một số lượng ngẫu nhiên các đặc trưng được chọn để tạo ra một cây quyết định.
- Voting hoặc Average:** Kết quả dự đoán từ tất cả các cây được kết hợp lại để đưa ra dự đoán cuối cùng. Trong phân loại, phương pháp voting được sử dụng; trong dự đoán, thường là trung bình hoặc trọng số của các dự đoán từ các cây.

Random Forest là một phương pháp mạnh mẽ và linh hoạt, thường được sử dụng cho các bài toán phân loại và dự đoán với dữ liệu lớn và đa dạng.

2) *Huấn luyện từng cây quyết định:* Quá trình huấn luyện từng cây quyết định có thể mô tả như sau:

- Lấy mẫu dữ liệu:** Đặc trưng đầu vào  $X$  và nhãn tương ứng  $Y$ . Lấy mẫu dữ liệu từ  $X$  và  $Y$  bằng phương pháp tái chọn mẫu Bootstrap.
- Xây dựng cây quyết định:** Xây dựng một cây quyết định trên tập dữ liệu con này.

#### E. LSTM

LSTMs (Long Short-Term Memory) là một loại kiến trúc đặc biệt của mạng nơ-ron hồi quy (RNN) được thiết kế để giải quyết một thách thức cụ thể—nhớ thông tin trong thời gian dài. Các mô hình này tăng cường khả năng ghi nhớ của các mạng nơ-ron hồi quy. Trong khi các mạng RNN thường chỉ giữ được bộ nhớ ngắn hạn, sử dụng thông tin trước đó cho các nhiệm vụ ngay lập tức trong mạng nơ-ron hiện tại, LSTMs được sử dụng rộng rãi trong các mạng nơ-ron dựa trên RNNs. Hiệu quả của LSTMs trải rộng qua nhiều vấn đề mô hình hóa chuỗi trong nhiều lĩnh vực ứng dụng khác nhau, bao gồm video, xử lý ngôn ngữ tự nhiên (NLP), dữ liệu địa không gian, và phân tích chuỗi thời gian. **Vấn Đề Tiêu Biểu của**

#### RNNs và Giải Pháp:

Một vấn đề đáng kể với RNNs là hiện tượng gradient biến mất. Vấn đề này phát sinh do việc sử dụng lại các tham số giống nhau trong các khối RNN ở mỗi bước. Để giải quyết vấn đề này, chúng ta cần cố gắng giới thiệu các tham số thay đổi ở mỗi bước thời gian. **Tìm Kiếm Sự Cân Bằng:**

Tìm kiếm sự cân bằng trong các tình huống như vậy là rất quan trọng. Chúng ta cần kết hợp các tham số mới ở mỗi bước đồng thời tổng quát hóa các chuỗi có độ dài biến đổi và duy trì số lượng tham số có thể học được không đổi. Điều này dẫn đến việc giới thiệu các tế bào RNN có cổng, chẳng hạn như LSTM và GRU (Gated Recurrent Unit).

LSTMs và GRUs là những tế bào có cổng trong mạng RNN, giúp giải quyết vấn đề gradient biến mất và cải thiện khả năng ghi nhớ trong thời gian dài, qua đó nâng cao hiệu quả của các ứng dụng như xử lý ngôn ngữ tự nhiên, phân tích video, và dữ liệu địa không gian.

#### F. VAR

Vector autoregression (VAR) là một mô hình thống kê được sử dụng để nắm bắt mối quan hệ giữa nhiều đại lượng khi chúng thay đổi theo thời gian. VAR là một loại mô hình quá trình ngẫu nhiên. Các mô hình VAR mở rộng mô hình tự hồi quy đơn biến (univariate autoregressive model) bằng cách cho phép phân tích các chuỗi thời gian đa biến. VAR thường được sử dụng trong kinh tế học và các khoa học tự nhiên. **Cấu Trúc và Hoạt Động của Mô Hình VAR:**

Giống như mô hình tự hồi quy, mỗi biến trong mô hình VAR có một phương trình mô tả sự tiến triển của nó theo thời gian. Phương trình này bao gồm các giá trị trễ (giá trị trong quá khứ) của biến đó, các giá trị trễ của các biến khác trong mô hình và một hạng tử sai số (error term).

Cụ thể: -Giá trị trễ của biến: Những giá trị trong quá khứ của biến đó. -Giá trị trễ của các biến khác: Những giá trị trong quá khứ của các biến khác trong mô hình. -Hạng tử sai số: Biểu diễn những yếu tố ngẫu nhiên hoặc không thể đoán trước được ảnh hưởng đến biến đó.

#### G. GRU

Gated Recurrent Unit (GRU) là một biến thể của mạng neural hồi quy (RNN), được thiết kế để cải thiện khả năng lưu giữ và xử lý thông tin trong các chuỗi dữ liệu dài hạn mà không bị mất mát thông tin. GRU sử dụng các cơ chế cổng để điều chỉnh luồng thông tin vào và ra khỏi các trạng thái ẩn của mạng, giúp cập nhật trạng thái một cách chọn lọc và hiệu quả. Hai cổng chính của GRU là cổng đặt lại (reset gate) và cổng cập nhật (update gate).

- Cổng đặt lại (reset gate):** Điều chỉnh mức độ thông tin từ trạng thái ẩn trước đó cần bị lãng quên.
- Cổng cập nhật (update gate):** Quyết định mức độ thông tin mới từ đầu vào sẽ được sử dụng để cập nhật trạng thái ẩn.

GRU tính toán đầu ra dựa trên trạng thái ẩn được cập nhật qua các cổng này, giúp duy trì thông tin một cách hiệu quả

trong suốt quá trình xử lý chuỗi.

Dưới đây là các phương trình mô tả cách tính toán của GRU:

- Cổng đặt lại:

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (4)$$

- Cổng cập nhật:

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (5)$$

- Trạng thái dự kiến:

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \quad (6)$$

- Trạng thái ẩn cuối cùng:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (7)$$

Trong các phương trình trên:

- $W_{xr}, W_{xz}, W_{xh}$  là các ma trận trọng số có thể được học.
- $x_t$  là đầu vào tại thời điểm  $t$ .
- $h_{t-1}$  là trạng thái ẩn từ thời điểm trước.
- $\sigma$  là hàm kích hoạt sigmoid và  $\tanh$  là hàm kích hoạt hyperbolic tangent.
- $b_r, b_z, b_h$  là các giá trị bias.

Bằng cách điều chỉnh thông tin qua các cổng, GRU có thể xử lý các chuỗi dữ liệu dài hiệu quả hơn, đặc biệt hữu ích trong các ứng dụng như dự báo giá tiền điện tử. GRU đơn giản hơn so với LSTM nhưng vẫn duy trì được khả năng mạnh mẽ trong việc xử lý các chuỗi dữ liệu phức tạp.

#### H. TimesNet

TimesNet là một mô hình hiện đại cho dự báo chuỗi thời gian, tận dụng kiến trúc đặc biệt để học và nắm bắt các mô hình thời gian phức tạp và đa dạng. Nó được thiết kế để cải thiện hiệu quả so với các phương pháp truyền thống bằng cách sử dụng cơ chế mạng lưới đặc trưng thời gian (Time-Series Network) và đặc biệt nhắm đến các đặc điểm độc đáo của dữ liệu chuỗi thời gian. TimesNet xây dựng các mạng con đặc trưng thời gian bằng cách:

- Bộ Biểu Diễn Thời Gian: Sử dụng các kiến trúc convolution và attention đặc trưng thời gian để học và tách các mẫu từ dữ liệu chuỗi thời gian. Bộ biểu diễn này được thiết kế để phát hiện và tổng hợp các đặc trưng từ cả thành phần tuần hoàn (như mùa) và phi tuần hoàn (như xu hướng ngắn hạn).
- Biểu Diễn Khối và Nối Kết: Các biểu diễn thời gian được tổ chức thành các khối và được nối kết để duy trì thông tin thời gian trong quá trình huấn luyện và dự báo.

TimeBlock là một thành phần quan trọng trong kiến trúc của TimesNet, được sử dụng để xử lý và trích xuất các đặc trưng từ dữ liệu chuỗi thời gian. TimeBlock kết hợp các phương pháp convolutional và attention để phân tích dữ liệu thời gian theo các khía cạnh khác nhau và tạo ra các biểu diễn đặc trưng cho mô hình dự báo. Mỗi TimeBlock thường bao gồm một chuỗi các lớp convolutional và attention, cùng với các kết nối và kết hợp đặc trưng để duy trì thông tin thời gian trong quá trình huấn luyện và dự báo.

- Convolution và Attention Thời Gian: Sử dụng convolution để khai thác các đặc trưng ngắn hạn và attention để phát hiện các mẫu quan trọng trong chuỗi dữ liệu dài hạn.
- Cơ Chế Tập Trung (Attention): Tăng cường khả năng của mô hình trong việc tập trung vào các thành phần quan trọng của dữ liệu thời gian, điều chỉnh sự chú ý đến các điểm dữ liệu có ảnh hưởng lớn đến dự báo.
- Tích Hợp Đa Chiều: Tích hợp thông tin từ nhiều chiều dữ liệu, đảm bảo rằng tất cả các khía cạnh của dữ liệu chuỗi thời gian được xem xét trong dự báo.

TimesNet có thể được mô tả bằng một tập hợp các công thức toán học, bao gồm:

- Khối Convolution:

$$\mathbf{F}^{(l)} = \text{Conv1D}(\mathbf{F}^{(l-1)}, \mathbf{W}^{(l)}) + \mathbf{b}^{(l)} \quad (8)$$

Trong đó,  $\mathbf{F}^{(l)}$  là đặc trưng tại lớp  $l$ , Conv1D là phép tích chập 1 chiều,  $\mathbf{W}^{(l)}$  là trọng số của lớp  $l$ , và  $\mathbf{b}^{(l)}$  là vector bù.

- Khối Attention:

$$\mathbf{Z}^{(l)} = \text{Attention}(\mathbf{Q}^{(l)}, \mathbf{K}^{(l)}, \mathbf{V}^{(l)}) \quad (9)$$

Trong đó,  $\mathbf{Q}^{(l)}, \mathbf{K}^{(l)}, \mathbf{V}^{(l)}$  là các ma trận query, key, và value tại lớp  $l$ .

- Kết Hợp Đặc Trưng:

$$\mathbf{F}_{\text{out}} = \text{Concat}(\mathbf{F}_{\text{conv}}, \mathbf{Z}_{\text{attn}}) \quad (10)$$

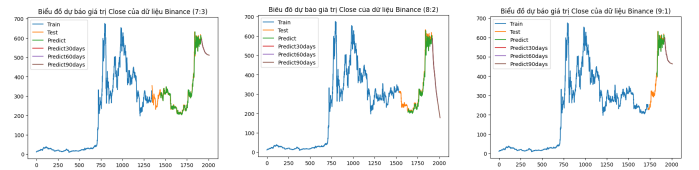
Trong đó,  $\mathbf{F}_{\text{conv}}$  và  $\mathbf{Z}_{\text{attn}}$  là các đặc trưng từ các khối convolution và attention.

TimesNet có khả năng được tùy chỉnh cho nhiều loại dữ liệu chuỗi thời gian khác nhau. Thông qua việc sử dụng TimeBlock, TimesNet có khả năng học và nắm bắt các mô hình thời gian phức tạp và đa dạng, thích nghi tốt với các đặc điểm cụ thể của từng tập dữ liệu, từ các mẫu ngắn hạn đến các xu hướng dài hạn trong dữ liệu chuỗi thời gian.

## V. THỰC NGHIỆM

### A. Xây dựng mô hình

#### 1) RNN:



Hình 7. Dữ liệu BNB-USD



Hình 8. Dữ liệu BTC-USD

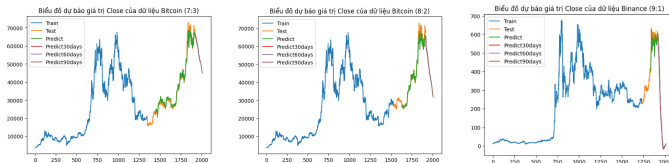


Hình 9. Dữ liệu ETH-USD

## 2) CNN-LSTM:



Hình 10. Dữ liệu BNB-USD



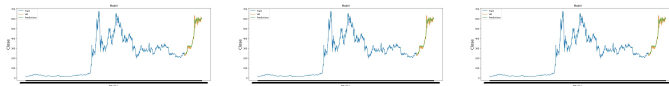
Hình 11. Dữ liệu BTC-USD



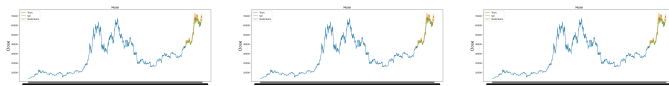
Hình 12. Dữ liệu ETH-USD

## 3) CNN-LSTM:

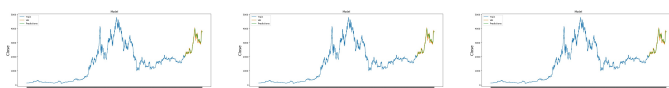
## 4) LSTM:



Hình 13. Dữ liệu BNB-USD

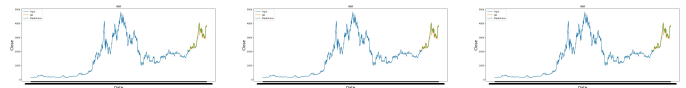


Hình 14. Dữ liệu BTC-USD

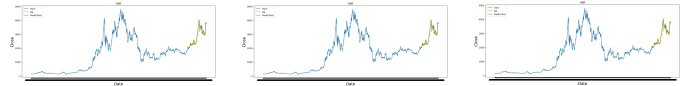


Hình 15. Dữ liệu ETH-USD

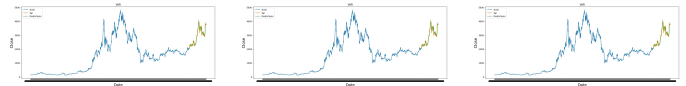
## 5) VAR:



Hình 16. Dữ liệu BNB-USD



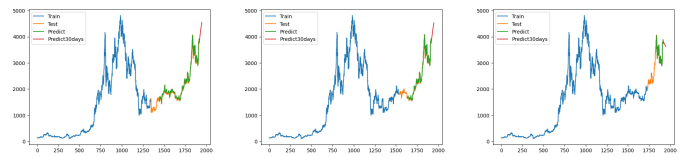
Hình 17. Dữ liệu BTC-USD



Hình 18. Dữ liệu ETH-USD

## 6) VAR:

## 7) GRU:



Hình 19. Dữ liệu ETH-USD

## 8) GRU:

## B. Độ đo đánh giá

## TÀI LIỆU

- [1] Yecheng Yao, Jungho Yi, Shengjun Zhai, Yuwen Lin, Taekseung Kim Guihongxuan Zhang, Leonard Yoonjae Lee, "Predictive Analysis of Cryptocurrency Price Using Deep Learning", International Journal of Engineering and Technology, Volume 7, Issue 3.27, pp. 258-264, 2018 [https://www.sciencepubco.com/index.php/ijet/article/view/17889]
- [2] Muhammad Ali Nasir, Toan Luu Duc Huynh, Sang Phu Nguyen and Duy Duong, "Forecasting cryptocurrency returns and volume using search engines", Financial Innovation 5, Article number 2, 2019 [https://jfin-swufe.springeropen.com/articles/10.1186/s40854-018-0119-8]
- [3] Aggarwal A., Gupta I., Garg N., and Goel A., "Deep Learning Approach to Determine the Impact of Socio-Economic Factors on Bitcoin Price Prediction", Twelfth International Conference on Contemporary Computing (IC3), 2019 [https://ieeexplore.ieee.org/document/8844928]
- [4] Alessandretti, L., ElBahrawy, A., Aiello, L.M. and Baronchelli, A., 2018. Anticipating cryptocurrency prices using machine learning. Complexity, 2018. [https://www.hindawi.com/journals/complexity/2018/8983590/]
- [5] Mittal R., Arora S. and Bhatia M. P. S., "Automated cryptocurrencies prices prediction using machine learning", International Journal on Soft Computing, Volume 8, Issue 4, pp. 1758-1761, 2018
- [6] Suhwan Ji, Jongmin Kim ORCID and Hyeonseung Im, "A Comparative Study of Bitcoin Price Prediction Using Deep Learning", Department of Computer Science, Kangwon National University, Chuncheon-si, Gangwon-do 24341, Korea, 2019 [https://www.mdpi.com/2227-7390/7/10/898]
- [7] Phumudzo Lloyd Seabe, Claude Rodrigue Bambe Moutsinga, and Edson Pindza, "Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach", 2023 [https://www.mdpi.com/2504-3110/7/2/203]
- [8] Nguyen Dinh Thuan, Nguyen Minh Nhut, Hoang Tung, Vu Minh Sang, "PREDICTING THE CLOSING PRICE OF CRYPTOCURRENCY USING HYBRID ARIMA, REGRESSION AND MACHINE LEARNING", Ky yeu Hoi nghi KHCN Quoc gia lan thu XIV ve Nghien cuu co ban va ung dung Cong nghe thong tin (FAIR), TP. HCM, ngày 23-24/12/2021