

# Bitcoin Price Prediction Using Machine Learning

Lê Châu Khánh Vy, Nguyễn Hạ Phương, Phạm Thị Anh Đào, Lạc Ngọc Như và Bùi Minh Tuấn

Lớp IS403.P21, Trường Đại học Công nghệ Thông tin, ĐHQG-HCM  
MSSV: 22521704, 22521166, 22520204, 22521048, 22521596  
{22521704, 22521166, 22520204, 22521048, 22521596}@gm.uit.edu.vn

**Tóm tắt** Bitcoin là một loại tiền điện tử phi tập trung hoạt động trên nền tảng công nghệ blockchain. Với đặc tính phi tập trung và khối lượng giao dịch lớn toàn cầu, giá Bitcoin biến động mạnh và chịu ảnh hưởng từ nhiều yếu tố như cung cầu thị trường, tin tức kinh tế, chính sách tiền tệ và cảm xúc nhà đầu tư. Việc dự đoán giá Bitcoin là một bài toán quan trọng trong lĩnh vực tài chính nhằm hỗ trợ các quyết định đầu tư. Trong nghiên cứu này, chúng tôi thu thập dữ liệu giá Bitcoin từ năm 4/2014 đến 12/2022 và áp dụng nhiều mô hình dự đoán khác nhau như ARIMA, SARIMAX, LSTM, BiLSTM, GRU, RNN, GPR, SVR, Prophet, Transformer, Diffusion-VAE, SARIMAX-LSTM, SARIMAX-Transformer và mô hình kết hợp LightGBM-LSTM. Kết quả cho thấy ba mô hình có hiệu suất dự đoán tốt nhất là SARIMAX-LSTM với MAPE là 0.0028%, SARIMAX-Transformer với MAPE là 0.3746% và GPR với MAPE là 0.70%.

**Từ khóa:** Dự đoán giá Bitcoin, ARIMA, LSTM, GPR, SARIMAX, Transformer, học máy, chuỗi thời gian

## 1 Giới thiệu đề tài

Trong kỷ nguyên số hóa, Bitcoin đã nổi lên như một loại tài sản kỹ thuật số có giá trị, thu hút sự quan tâm của các nhà đầu tư, tổ chức tài chính, và nhà nghiên cứu trên toàn thế giới. Với tính chất phi tập trung, ẩn danh và giới hạn nguồn cung, Bitcoin thường được ví như “vàng kỹ thuật số” và trở thành đối tượng đầu cơ cũng như đầu tư dài hạn.

Giá Bitcoin thường xuyên biến động với biên độ lớn, gây khó khăn trong việc đưa ra quyết định đầu tư hiệu quả. Do đó, việc phát triển các mô hình dự đoán giá Bitcoin mang lại nhiều lợi ích to lớn. Đầu tiên, giúp nhận diện xu hướng thị trường và thời điểm đầu tư hợp lý. Thứ hai, hỗ trợ nhà đầu tư xây dựng chiến lược mua – bán tối ưu. Thứ ba, quản lý rủi ro tài chính cá nhân và doanh nghiệp hiệu quả. Cuối cùng là tăng cường hiểu biết về hành vi thị trường tiền điện tử.

Mặc dù các tài liệu hiện có đã có những đóng góp đáng kể vào dự báo giá Bitcoin, nhưng vẫn còn một số khoảng cách. Hầu hết các nghiên cứu tập trung vào các phương pháp tiếp cận mô hình đơn lẻ mà không có sự so sánh toàn diện giữa các phương pháp khác nhau. Ngoài ra, việc tích hợp các kỹ thuật học sâu tiên tiến với các phương pháp kinh tế lượng truyền thống vẫn chưa được khám phá đầy đủ. Bản chất biến động và không ổn định của giá Bitcoin đặt ra những thách thức liên tục đòi hỏi các phương pháp tiếp cận mô hình sáng tạo.

Tần suất cao và tính phức tạp của dữ liệu Bitcoin đòi hỏi các mô hình có thể nắm bắt hiệu quả cả các mô hình tuyến tính và phi tuyến tính, xử lý nhiều thang thời gian và kết hợp nhiều yếu tố bên ngoài. Điều này thúc đẩy nhu cầu về các nghiên cứu toàn diện so sánh và kết hợp một cách có hệ thống các phương pháp dự báo khác nhau để đạt được độ chính xác dự đoán vượt trội. Trong nghiên cứu này, chúng tôi áp dụng và so sánh hiệu quả của nhiều mô hình học máy, bao gồm các mô hình thống kê (ARIMA, SARIMAX), mô hình học máy truyền thống (SVR, GPR, Prophet, LightGBM-LSTM), mô hình học sâu (LSTM, BiLSTM, GRU, RNN, Transformer), mô hình lai

(SARIMAX-LSTM, SARIMAX-Transformer, LightGBM-LSTM) và mô hình sinh hiện đại (Diffusion-VAE).

Chúng tôi huấn luyện các mô hình trên dữ liệu giá Bitcoin từ năm 2014 đến năm 2022, sau đó đánh giá độ chính xác dự báo bằng các chỉ số MAE, RMSE và MAPE. Qua đó, nhóm nghiên cứu hướng tới việc tìm ra mô hình tối ưu nhất về độ chính xác để đề xuất phương pháp dự báo hiệu quả trong thực tiễn. Đây là bước khởi đầu quan trọng trong việc khai thác tiềm năng của công nghệ học sâu trong bối cảnh thị trường tài sản số đầy biến động.

## 2 Nghiên cứu liên quan

“Linkage Analysis Between Bitcoin and Nasdaq Index Based on ARIMAX Model” (2022) [1], các tác giả R. Hu và X. Wang sử dụng cả ARIMA và ARIMAX để mô hình hóa giá đóng cửa của Bitcoin từ tháng 1 năm 2020 đến tháng 5 năm 2022, sử dụng chỉ số Nasdaq làm biến ngoại sinh trong mô hình ARIMAX. Kết quả cho thấy ARIMAX đạt hiệu suất dự đoán tốt hơn so với ARIMA đơn lẻ, làm nổi bật lợi ích của việc đưa các chỉ số thị trường bên ngoài vào.

“Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach” (2023) [2] của tác giả Seabe và cộng sự đã nghiên cứu so sánh hiệu suất của LSTM, GRU và LSTM song hướng (Bi-LSTM) trong việc dự báo giá tiền điện tử (BTC, ETH, LTC). Kết quả cho thấy Bi-LSTM đạt độ chính xác dự đoán cao nhất, tiếp theo là GRU, trong khi LSTM hoạt động kém hiệu quả hơn.

Các tác giả Hakan Pabuccu, Serdar Ongan và Ayse Ongan đã cung cấp các phương pháp dự đoán giá bitcoin trong bài báo “Forecasting the Movements of Bitcoin Prices: An Application of Machine Learning Algorithms” (2023) [3]. Nghiên cứu này đánh giá hiệu suất của một số thuật toán học máy, bao gồm SVM, ANN, Naive Bayes, Random Forest và Logistic Regression, trong việc dự đoán biến động giá Bitcoin. SVM vượt trội hơn các phương pháp khác, đặc biệt là về độ chính xác của phân tích tính.

“Cryptocurrency Price Movement Prediction Using the Hybrid SARIMAX-LSTM Method” (2023) [4], của các tác giả G. R. Achmadi, A. Saikhu, B. Amaliah, nghiên cứu này tích hợp SARIMAX, kết hợp khối lượng giao dịch như một biến ngoại sinh, với LSTM để dự đoán biến động giá tiền điện tử. Mô hình lai chứng minh độ chính xác dự đoán vượt trội, được chứng minh bằng các số liệu lỗi thấp hơn (MSE, RMSE, MAPE, MAE) so với các mô hình SARIMAX hoặc LSTM riêng lẻ.

“Performance Analysis of Bitcoin Forecasting Using Deep Learning Techniques” (2023) [5], của tác giả Nrusingha Tripathy và các cộng sự đã thực hiện so sánh hiệu năng giữa ARIMA, LSTM và FB-Prophet trên dữ liệu BTC từ Jan 2020 đến Sep 2020. Kết quả cho thấy FB-Prophet đạt RMSE thấp nhất (322.6), tốt hơn so với LSTM (421.3) và ARIMA (2462.5).

Bài báo “Machine Learning-Based Forecasting of Bitcoin Price Movements” (2024) [6], của D. Angelovski và cộng sự đánh giá hiệu suất của nhiều mô hình học máy khác nhau, bao gồm LSTM, Gradient Boosting Regression (GBR), LightGBM Regressor và Random Forest, để dự đoán biến động giá Bitcoin. Nghiên cứu này đánh giá chuẩn các phương pháp này về độ chính xác dự đoán và hiệu quả tính toán.

“Diffusion Variational Autoencoder for Tackling Stochasticity in Multi-Step Regression Stock Price Prediction” (2023) [7], của các tác giả Kelvin J.L. Koa, Yunshan Ma, Ritchie Ng, Tat-Seng Chua giới thiệu một phương pháp mới tích hợp các Mô hình khuếch tán với các Bộ mã hóa tự động biến thiên (VAE) để nâng cao khả năng dự báo giá cổ phiếu. Mô hình này giải quyết hiệu quả các thách thức như tính ngẫu nhiên và dự đoán nhiều bước trong chuỗi thời gian tài chính. Bằng cách tận dụng sức mạnh biểu diễn của VAE và mô hình ngẫu nhiên của các quá trình khuếch tán, phương pháp này nắm bắt được cả nhiễu ngắn hạn và xu hướng dài hạn.

### 3 Thuật toán sử dụng

#### 3.1 Tổng quan dữ liệu

Bitcoin, loại tiền điện tử đầu tiên và nổi tiếng nhất, được giới thiệu dưới dạng phần mềm mã nguồn mở vào năm 2009 bởi Satoshi Nakamoto. Nó hoạt động như một phương tiện trao đổi kỹ thuật số phi tập trung mà không cần đến các trung tâm thanh toán bù trừ hoặc các cơ quan lưu trữ hồ sơ đáng tin cậy. Các giao dịch được xác thực và ghi lại trong một sổ cái công khai phân tán, blockchain, mà không cần trung gian. Việc áp dụng nhanh chóng Bitcoin đã dẫn đến sự phát triển của các công cụ tài chính và hệ thống giao dịch xung quanh nó, khiến việc phân tích dữ liệu lịch sử của nó để tìm ra các mô hình và dự đoán trở nên cần thiết.

Bộ dữ liệu được sử dụng chứa các giá trị trung bình hàng ngày cho giá và các thuộc tính liên quan đến Bitcoin: priceUSD (Giá Bitcoin trung bình hàng ngày tính bằng USD), size (Tổng kích thước blockchain tính bằng gigabyte), sentbyaddress (Số lượng địa chỉ gửi duy nhất), transactions (Tổng số giao dịch Bitcoin), mining\_profitability (Chỉ số lợi nhuận khai

thác Bitcoin), sentinusd (Tổng giá trị Bitcoin được gửi bằng USD), transactionfees (Tổng phí giao dịch), average\_transaction\_fee (Phí giao dịch trung bình cho mỗi giao dịch), confirmationtime (Thời gian xác nhận giao dịch trung bình), marketcap (Vốn hóa thị trường của Bitcoin), transactionvalue (Tổng giá trị giao dịch), averagetransactionvalue (Giá trị trung bình cho mỗi giao dịch), tweets (Số lượng tweet liên quan đến Bitcoin), google\_trends (Xu hướng tìm kiếm Bitcoin trên Google), fee\_to\_reward (Tỷ lệ phí/phần thưởng), activeaddresses (Số lượng địa chỉ Bitcoin đang hoạt động), top100cap (Vốn hóa thị trường của 100 người nắm giữ Bitcoin hàng đầu).

Bộ dữ liệu được trích xuất từ bitinfocharts.com bằng công cụ thu thập dữ liệu web trong khoảng thời gian từ ngày 9 tháng 4 năm 2014 đến ngày 30 tháng 12 năm 2022.

Trong đề tài này nhóm sử dụng giá các thuộc tính chỉ số giá bitcoin để làm đầu vào dự đoán giá bitcoin trong khoảng thời gian tương lai. Bằng cách nắm bắt, thu thập các khía cạnh kỹ thuật, tài chính và hành vi của thị trường Bitcoin, các thuộc tính này cùng nhau cho phép dự đoán giá chính xác và đáng tin cậy hơn, điều này rất cần thiết cho các nhà giao dịch, nhà đầu tư và nhà nghiên cứu. Nhóm chia bộ dữ liệu thành 3 tập train, validation, test với tỉ lệ 80/10/10.

	count	mean	std	min	25%	50%	75%	max	range	median
priceUSD	3180.0	1.221066e+04	1.600727e+04	1.975770e+02	6.039980e+02	6.571500e+03	1.523850e+04	6.754700e+04	6.734942e+04	6.571500e+03
size	3180.0	7.021254e+05	2.083265e+05	1.622600e+05	5.838952e+05	7.405370e+05	8.792220e+05	9.980920e+05	8.358320e+05	7.405370e+05
sentbyaddress	3180.0	3.709351e+05	1.329934e+05	7.090200e+04	2.880686e+05	3.902455e+05	4.760812e+05	7.301990e+05	6.592970e+05	3.902455e+05
transactions	3180.0	2.368830e+05	8.428374e+04	4.757900e+04	1.922695e+05	2.520945e+05	2.965680e+05	4.904590e+05	4.428800e+05	2.520945e+05
mining_profitability	3180.0	1.957467e+00	5.041255e+00	5.250000e-02	1.740000e-01	3.775000e-01	1.467500e+00	4.070600e+01	4.065350e+01	3.775000e-01
sentinusd	3180.0	1.192082e+10	2.344175e+10	9.471432e+07	9.354786e+08	4.374355e+09	1.152109e+10	8.832295e+11	3.832348e+11	4.374355e+09
transactionfees	3180.0	2.704139e+00	5.803913e+00	2.410000e-02	1.547500e-01	8.875000e-01	2.380500e+00	6.277900e+01	6.275490e+01	8.875000e-01
median_transaction_fee	3180.0	1.309616e+00	3.093767e+00	9.970000e-03	6.217500e-02	3.045000e-01	9.380000e+00	3.409500e+01	3.408030e+01	3.045000e-01
confirmationtime	3180.0	9.787805e+00	1.183435e+00	0.000000e+00	9.057000e+00	9.664000e+00	1.036000e+01	2.482300e+01	2.482300e+01	9.664000e+00
marketcap	3180.0	2.257468e+11	3.026721e+11	2.658832e+09	9.040045e+09	1.135541e+11	2.580889e+11	1.274513e+12	1.271857e+12	1.135541e+11
transactionvalue	3180.0	1.246725e+05	2.371334e+05	1.166000e+03	5.850000e+03	2.560200e+04	1.000435e+05	2.450460e+05	2.449294e+05	2.560200e+04
mediantransactionvalue	3180.0	4.066500e+02	4.522582e+02	9.170000e-02	1.292378e+02	3.055255e+02	5.423808e+02	5.417700e+03	5.416908e+03	3.055255e+02
tweets	3180.0	5.272518e+04	4.720937e+04	0.000000e+00	2.191756e+04	3.019150e+04	7.505750e+04	3.635660e+05	3.635660e+05	3.019150e+04
google_trends	3180.0	5.119651e+01	5.764727e+01	5.268000e+00	9.880500e+00	3.571800e+01	6.369425e+01	6.168670e+02	6.115990e+02	3.571800e+01
fee_to_reward	3180.0	4.113215e+00	5.351474e+00	1.890000e-01	9.310000e-01	1.608000e+00	5.380500e+00	4.358000e+01	4.339100e+01	1.608000e+00
activeaddresses	3180.0	5.769545e+05	2.405334e+05	0.000000e+00	4.152070e+05	5.958555e+05	7.662425e+05	1.196069e+06	1.196069e+06	5.958555e+05
top100cap	3180.0	1.717996e+01	2.401756e+00	0.000000e+00	1.515350e+01	1.722350e+01	1.920750e+01	2.180900e+01	2.180900e+01	1.722350e+01

Hình 1: Thống kê mô tả các đặc trưng trong tập dữ liệu Bitcoin từ 2014 đến 2022

#### 3.2 Thuật toán

**LSTM** là một biến thể của mạng RNN, được thiết kế để giải quyết vấn đề mất thông tin khi xử lý chuỗi dài (long-term dependencies). Nhờ vào cấu trúc đặc biệt gồm ba cổng (input, forget, output), LSTM có khả năng chọn lọc thông tin cần nhớ hoặc quên, phù hợp cho các bài toán chuỗi thời gian như dự báo giá Bitcoin.

*Đặc điểm chính:*

- LSTM giải quyết hiện tượng mất dần gradient (vanishing gradient) trong RNN truyền thống.
- Gồm ba cổng điều khiển: cổng vào (input gate), cổng quên (forget gate), và cổng ra (output gate).
- Thích hợp với dữ liệu chuỗi dài và có tính phụ thuộc thời gian như dữ liệu thị trường tài chính.

Hàm mô hình LSTM:

$$\begin{aligned}f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\h_t &= o_t \odot \tanh(C_t)\end{aligned}$$

Kiến trúc mô hình:

- Mạng Sequential gồm:
  - 1 lớp LSTM với số units thay đổi trong khoảng từ 32 đến 128.
  - 1 lớp Dropout với giá trị trong khoảng từ 0.1 đến 0.5.
  - 1 lớp Dense đầu ra với 1 neuron để dự đoán giá Bitcoin (priceUSD).
- Hàm kích hoạt: ReLU.
- Bộ tối ưu: Adam, với tốc độ học learning\_rate  $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$ .
- Đầu vào: (1, num\_features) với 9 đặc trưng (features).

Phạm vi tìm kiếm siêu tham số (Keras Tuner):

- units  $\in \{32, 64, 96, 128\}$ ,
- dropout  $\in \{0.1, 0.2, \dots, 0.5\}$ ,
- learning\_rate  $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$ ,
- epochs = 50, batch\_size = 32

Tham số tối ưu: Mô hình LSTM đạt kết quả tốt nhất với:

- units = 96, dropout = 0.2, learning\_rate = 0.001



Hình 2: Kết quả mô hình LSTM

**BiLSTM** là một phần mở rộng của mô hình LSTM, cho phép mạng học được cả hai chiều thời gian: quá khứ (forward) và tương lai (backward). Nhờ đó, mô hình có thể khai thác được nhiều thông tin ngữ cảnh hơn trong chuỗi thời gian, giúp cải thiện độ chính xác trong dự báo giá Bitcoin.

Đặc điểm chính:

- BiLSTM bao gồm hai lớp LSTM chạy song song: một theo chiều thời gian xuôi và một theo chiều ngược.
- Mô hình có khả năng học thông tin từ cả quá khứ và tương lai, từ đó hiểu rõ hơn các mối quan hệ trong chuỗi.
- Đặc biệt hữu ích trong các bài toán chuỗi thời gian mà mỗi điểm dữ liệu phụ thuộc vào bối cảnh xung quanh (context).

Hàm mô hình BiLSTM: Mỗi lớp BiLSTM có thể được biểu diễn bằng việc gộp đầu ra của hai LSTM:

$$h_t^{\text{Bi}} = \left[ \vec{h}_t, \overleftarrow{h}_t \right]$$

trong đó  $\vec{h}_t$  là trạng thái ẩn từ LSTM chiều xuôi, và  $\overleftarrow{h}_t$  là từ chiều ngược.

Kiến trúc mô hình:

- Mạng Sequential gồm:
  - 1 lớp Bidirectional (LSTM) với số units thay đổi từ 32 đến 128.
  - 1 lớp Dropout để tránh overfitting.
  - 1 lớp Dense đầu ra với 1 neuron để dự đoán priceUSD.
- Hàm kích hoạt: ReLU.
- Bộ tối ưu: Adam, với learning\_rate  $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$ .
- Đầu vào: (1, num\_features), với 9 đặc trưng.

Phạm vi tìm kiếm siêu tham số (Keras Tuner):

- units  $\in \{32, 64, 96, 128\}$ ,
- dropout  $\in \{0.1, 0.2, \dots, 0.5\}$ ,
- learning\_rate  $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$ ,
- epochs = 50, batch\_size = 32

Tham số tối ưu: Mô hình BiLSTM đạt kết quả tốt nhất với:

- units = 64, dropout = 0.1, learning\_rate = 0.001



Hình 3: Kết quả mô hình BiLSTM

**ARIMA** ARIMA (Autoregressive Integrated Moving Average) là một mô hình thống kê phổ biến dùng trong phân tích chuỗi thời gian, đặc biệt hiệu quả trong lĩnh vực tài chính. ARIMA kết hợp ba thành phần: hồi quy tự động (AR), sai phân (I), và trung bình trượt (MA), giúp dự báo dữ liệu có xu hướng và nhiễu.

*Đặc điểm chính:*

- Thành phần AR (Autoregressive): mô hình hóa mối quan hệ giữa giá trị hiện tại và các giá trị quá khứ gần đó.
- Thành phần I (Integrated): giúp làm chuỗi trở nên dừng thông qua quá trình sai phân bậc  $d$ .
- Thành phần MA (Moving Average): mô hình hóa ảnh hưởng của nhiễu trắng từ các bước trước.

*Hàm mô hình:*

$$y_t = c + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

*Cấu hình mô hình:*

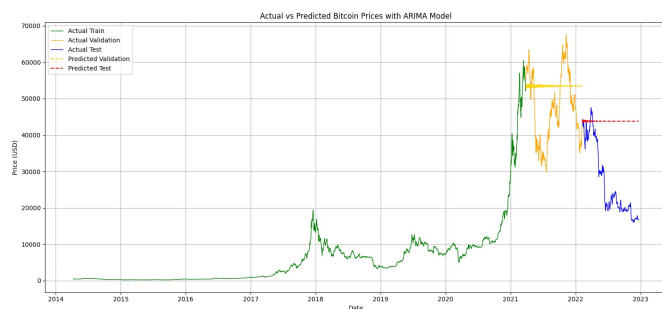
- $p = 2$ : sử dụng hai giá trị trễ để dự đoán giá trị hiện tại (tính tự hồi quy).
- $d = 1$ : sai phân một lần để loại bỏ xu hướng, được xác định qua kiểm định Augmented Dickey-Fuller.
- $q = 4$ : sử dụng bốn sai số quá khứ để điều chỉnh đầu ra hiện tại (trung bình trượt).

*Phạm vi tìm kiếm tham số:*

- $p \in [0, 4]$
- $q \in [0, 4]$
- $d = 1$

*Tiêu chí lựa chọn mô hình:*

- Với mỗi tổ hợp  $(p, d, q)$ , mô hình ARIMA được huấn luyện và tính toán chỉ số Akaike Information Criterion (AIC).
- Mô hình có AIC thấp nhất là **ARIMA(2,1,4)**, được chọn là mô hình tối ưu cho bài toán dự báo giá Bitcoin.



Hình 4: Kết quả mô hình ARIMA

**GRU** Gated Recurrent Unit (GRU) là một biến thể của mạng nơ-ron hồi tiếp (RNN), được thiết kế để giải quyết

các vấn đề về ghi nhớ dài hạn trong chuỗi thời gian. GRU có cấu trúc đơn giản hơn so với LSTM nhờ loại bỏ cổng đầu ra, nhưng vẫn duy trì khả năng học tốt các phụ thuộc dài trong dữ liệu.

*Đặc điểm chính:*

- GRU sử dụng hai cổng chính: *update gate* và *reset gate*, giúp kiểm soát luồng thông tin được lưu trữ hoặc quên đi.
- GRU ít tham số hơn LSTM nhưng hiệu quả vẫn rất cao trong các bài toán dự báo chuỗi như tài chính, dịch vụ thời gian thực, v.v.
- Cơ chế cập nhật ẩn giúp mô hình linh hoạt học được các mẫu dữ liệu có tính thời gian cao.

*Hàm cập nhật trạng thái:*

$$r_t = \sigma(W_r[h_{t-1}, X_t] + b_r) \quad (\text{Reset gate})$$

$$z_t = \sigma(W_z[h_{t-1}, X_t] + b_z) \quad (\text{Update gate})$$

$$\tilde{h}_t = \tanh(W_h[r_t * h_{t-1}, X_t] + b_h) \quad (\text{Candidate hidden})$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (\text{Final hidden})$$

*Kiến trúc mô hình:*

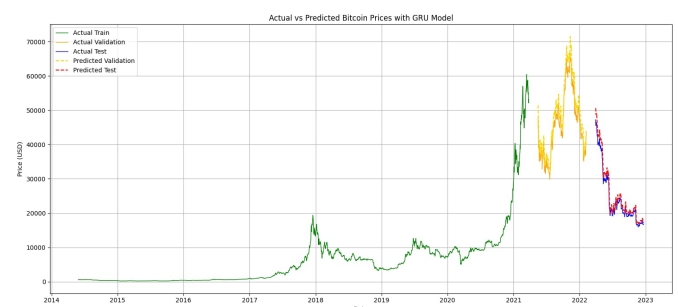
- 1 lớp GRU với `units = 64`
- 1 lớp Dropout với `dropout = 0.2` để giảm overfitting
- 1 lớp Dense đầu ra
- Optimizer: Adam với `learning rate = 0.001`
- Loss function: Mean Squared Error (MSE)
- `window_size = 50` ngày

*Phạm vi tìm kiếm tham số:*

- `units`  $\in \{32, 64, 96, 128\}$
- `batch_size`  $\in \{16, 32\}$
- `epochs = 30`
- `learning_rate`  $\in \{0.001, 0.0005\}$
- `dropout`  $\in \{0.0, 0.2, 0.5\}$

*Tham số tối ưu:*

- `units = 64`, `batch_size = 32`, `epochs = 30`, `learning_rate = 0.001`, `dropout = 0.2`



Hình 5: Kết quả mô hình GRU

## GPR

**Đặc điểm chính :** Mô hình phi tham số dựa trên quá trình Gauss, không chỉ cho ra giá trị dự đoán mà còn cả độ không chắc chắn (variance) của dự báo.

**Hàm mục tiêu:** Tối đa hoá log-likelihood của dữ liệu dưới giả thuyết quá trình Gauss

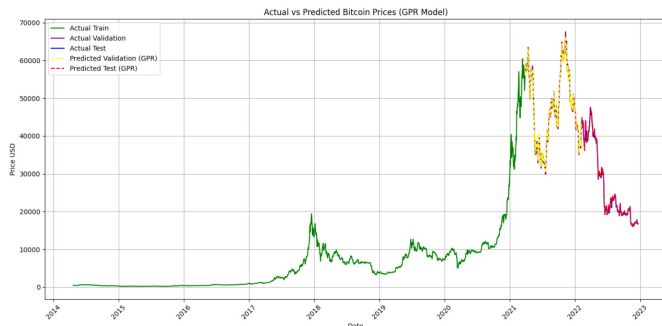
$$\log p(\mathbf{y} | \mathbf{X}, \theta) = -\frac{1}{2} \mathbf{y}^\top K_\theta^{-1} \mathbf{y} - \frac{1}{2} \log |K_\theta| - \frac{n}{2} \log(2\pi),$$

**Kiến trúc mô hình:** Kernel hỗn hợp  $k(\mathbf{x}, \mathbf{x}') = C \text{RBF}(\ell) + \sigma_{\text{noise}}^2 \delta_{\mathbf{x}\mathbf{x}'}$ , gồm:

- **ConstantKernel**  $C$  – biên độ tín hiệu.
- **RBF** với độ dài  $\ell$  – giả định hàm mục tiêu trơn.
- **WhiteKernel**  $\sigma_{\text{noise}}^2$  – mô hình hoá nhiễu quan sát.

**Phạm vi tìm kiếm tham số:**

- **n\_restarts\_optimizer = 10:** khởi động tối ưu 10 lần để tránh kẹt local-minimum.
- **alpha = 0.05:** số cộng thêm đường chéo (ổn định tính toán).
- **normalize\_y = True:** chuẩn hoá mục tiêu trước khi fit.
- **random\_state = 42**



Hình 6: Kết quả mô hình GPR

**LightGBM-LSTM** Để tận dụng đồng thời khả năng học quan hệ thời gian và phi tuyến, nhóm đã kết hợp hai mô hình mạnh là LSTM và LightGBM nhằm xây dựng hệ thống dự báo giá Bitcoin chính xác và ổn định hơn.

**Mô hình LSTM**

- Phạm vi tìm kiếm tham số:
  - Số nút LSTM: 32–128
  - Dropout: 0.1–0.5
  - Learning rate:  $10^{-4}$  đến  $10^{-2}$
  - Batch size: {16, 32, 64}
- Tham số tối ưu: Mô hình đạt kết quả tốt nhất với: **n\_units = 111, dropout  $\approx 0.2$ , learning\_rate  $\approx 0.0029$ , batch\_size = 16.**

**Mô hình LightGBM**

- Đặc điểm chính: Thuật toán boosting cây quyết định tốc độ cao, xây cây theo hướng *leaf-wise* và dùng kỹ thuật histogram để giảm bộ nhớ, nhờ đó học các quan hệ phi tuyến phức tạp mà vẫn rất nhanh.
- Hàm mục tiêu: Tối thiểu hóa hàm mất mát L2 (RMSE) kèm điều chuẩn:

$$\mathcal{L} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda_1 \sum_k |w_k| + \lambda_2 \sum_k w_k^2,$$

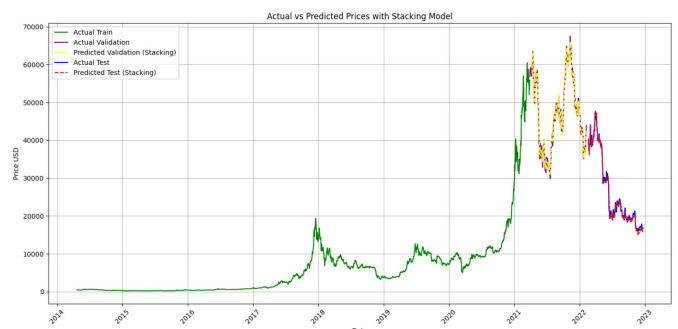
- Kiến trúc mô hình: Gồm nhiều cây quyết định tuần tự. Mỗi cây mới học từ sai số còn lại của toàn bộ mô hình trước đó. Các đặc trưng đầu vào là bảng tabular gồm giá trị, MA, Momentum, Volatility và chỉ báo on-chain.
- Phạm vi tìm kiếm tham số:
  - **num\_leaves:** 20–150
  - **feature\_fraction, bagging\_fraction:** 0.6–1.0
  - **bagging\_freq:** 1–10
  - **min\_child\_samples:** 5–100
  - **learning\_rate:** 0.005–0.05
  - **lambda\_l1, lambda\_l2:**  $10^{-8}$ –10
- Tham số tối ưu: Mô hình đạt kết quả tốt nhất với **num\_leaves = 94, feature\_fraction  $\approx 0.95$ , bagging\_fraction  $\approx 0.74$ , learning\_rate  $\approx 0.046$ , lambda\_l1  $\approx 0.0186$ , lambda\_l2  $\approx 0.000156$ .**

**ElasticNetCV (Stacking)**

- Đặc điểm chính: Hồi quy tuyến tính có điều chuẩn kết hợp L1 (Lasso) và L2 (Ridge), đóng vai trò meta-model để phối hợp dự đoán của LSTM và LightGBM.
- Hàm mục tiêu: Tối thiểu hoá sai số bình phương kèm hai thành phần phạt

$$\min_{\beta} \frac{1}{2n} \sum_{i=1}^n (y_i - X_i \beta)^2 + \alpha \left[ \lambda \|\beta\|_1 + (1 - \lambda) \|\beta\|_2^2 \right],$$

- Kiến trúc mô hình: Pipeline 2 bước: (i) chuẩn hoá dữ liệu đầu vào, (ii) tìm mô hình ElasticNet tối ưu ( $\alpha$ , **l1\_ratio**) bằng 5-fold CV. Kết quả là meta-model học cách kết hợp có trọng số hai dự đoán con, đồng thời điều chuẩn để tránh quá khớp.



Hình 7: Kết quả mô hình LightGBM-LSTM



**SVR** Support Vector Regression (SVR) là một biến thể của SVM được thiết kế cho bài toán hồi quy. Mô hình tìm một siêu phẳng sao cho sai số nằm trong khoảng dung sai  $\epsilon$ , đồng thời tối ưu hóa độ đơn giản của mô hình.

*Đặc điểm chính:*

- Hỗ trợ mô hình phi tuyến thông qua các hàm kernel (RBF, poly, sigmoid).
- Khoảng sai số  $\epsilon$  xác định vùng dự đoán chấp nhận được.
- Tham số  $C$  điều chỉnh mức độ phạt với điểm nằm ngoài vùng  $\epsilon$ .

*Hàm mục tiêu:*

$$\min \frac{1}{2} \|w\|^2 + C \sum (\xi_i + \xi_i^*) \quad \text{với} \quad \begin{cases} y_i - f(x_i) \leq \epsilon + \xi_i \\ f(x_i) - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

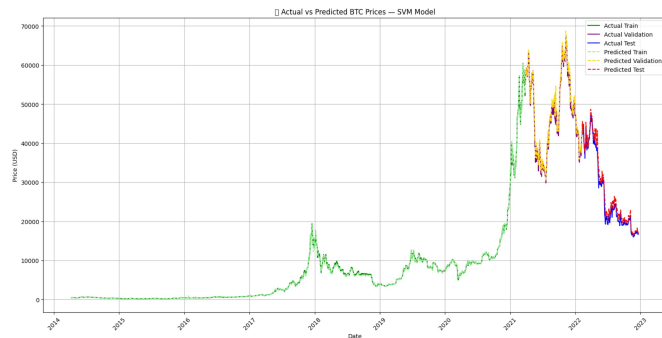
Trong đó:  $f(x) = w^T x + b$ , và  $\xi_i, \xi_i^*$  là biến trượt cho sai số vượt  $\epsilon$ .

*Kiến trúc mô hình:*

- Sử dụng **RandomizedSearchCV** kết hợp với **TimeSeriesSplit (3 folds)** để tìm tham số tối ưu.
- Phạm vi tìm kiếm tham số:
  - $C \in [10^{-2}, 10^3]$  (logspace, 100 giá trị),
  - $\epsilon \in [0.001, 1]$ ,
  - kernel: ['linear', 'rbf', 'poly', 'sigmoid'],
  - degree: [2, 3, 4, 5] (áp dụng cho kernel 'poly'),
  - gamma: ['scale', 'auto']  $\cup$  logspace(-4, 1, 6).

*Tham số tối ưu:*

- $C = 890.22$ ,  $\epsilon = 0.6165$ , kernel = rbf, gamma = scale



Hình 8: Kết quả mô hình SVR

**RNN** Recurrent Neural Network (RNN) là một mô hình học sâu phù hợp cho bài toán dự báo chuỗi thời gian, đặc biệt là khi dữ liệu có tính liên kết theo thời điểm. Mô hình có khả năng ghi nhớ thông tin trong quá khứ nhờ cơ chế hồi tiếp, giúp cải thiện chất lượng dự báo trong các bài toán như dự đoán giá Bitcoin.

*Đặc điểm chính:*

- RNN (Recurrent Neural Network) là mạng nơ-ron hồi tiếp, được thiết kế để xử lý dữ liệu tuần tự, đặc biệt phù hợp với chuỗi thời gian và dữ liệu ngữ cảnh.
- Mạng RNN có khả năng lưu trữ thông tin từ các bước thời gian trước đó thông qua trạng thái ẩn (*hidden state*), giúp mô hình nắm bắt được các phụ thuộc theo thời gian.
- Trái ngược với mạng truyền thống (feedforward), RNN cho phép các kết nối quay ngược, từ đầu ra về lại chính nó, tạo thành vòng lặp nội tại.

*Hàm mục tiêu:*

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

*Kiến trúc mô hình:*

- Một mạng **Sequential** gồm:
  - 1 lớp SimpleRNN với số lượng units thay đổi,
  - 1 lớp Dropout để tránh hiện tượng overfitting,
  - 1 lớp Dense đầu ra.
- Bộ tối ưu: Adam với tốc độ học learning rate được điều chỉnh.
- Cửa sổ thời gian: window\_size = 30.

*Phạm vi tìm kiếm tham số:*

- units  $\in \{32, 64, 96, 128\}$ ,
- batch\_size  $\in \{16, 32\}$ ,
- epochs = 50,
- learning\_rate  $\in \{0.001, 0.0005\}$ ,
- dropout  $\in \{0.0, 0.2, 0.5\}$ .

*Tham số tối ưu:*

- units = 96
- batch\_size = 16
- epochs = 50
- learning\_rate = 0.0005
- dropout = 0.0



Hình 9: Kết quả mô hình RNN

**Prophet** Facebook Prophet là một mô hình dự báo chuỗi thời gian được thiết kế để xử lý dữ liệu có xu hướng và yếu tố mùa vụ rõ ràng. Mô hình giả định cấu trúc cộng tuyến gồm thành phần xu hướng, mùa vụ và nhiễu.

*Đặc điểm chính:*

- Là mô hình chuỗi thời gian bán tham số, dễ giải thích, có khả năng dự đoán linh hoạt.
- Hỗ trợ thêm các biến ngoại sinh nhằm nâng cao độ chính xác dự báo.
- Tham số `changepoint_prior_scale` điều chỉnh độ nhạy với điểm đổi xu hướng.
- Tham số `seasonality_prior_scale` kiểm soát mức ảnh hưởng của các thành phần mùa vụ.
- Có thể tự động xử lý dữ liệu thiếu và nhiễu nhỏ.

*Hàm mô hình:*

$$y(t) = g(t) + s(t) + \sum_k \beta_k x_k(t) + \epsilon_t$$

Trong đó:

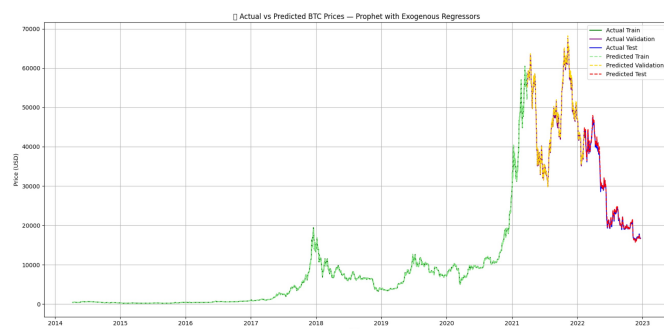
- $g(t)$ : thành phần xu hướng (linear hoặc logistic),
- $s(t)$ : thành phần mùa vụ (theo tuần, năm),
- $x_k(t)$ : các biến ngoại sinh,
- $\epsilon_t$ : nhiễu ngẫu nhiên.

*Kiến trúc mô hình:*

- Các biến ngoại sinh được chuẩn hóa bằng `StandardScaler`.
- Tìm kiếm tham số tối ưu bằng phương pháp **Grid search thủ công** với:
  - `changepoint_prior_scale`  $\in \{0.001, 0.01, 0.05, 0.1, 0.2\}$
  - `seasonality_prior_scale`  $\in \{0.1, 1.0, 5.0, 10.0, 20.0\}$
- Mùa vụ theo năm và tuần được bật:
  - `yearly_seasonality=True`,
  - `weekly_seasonality=True`,
  - `daily_seasonality=False`.

*Tham số tối ưu:*

- `changepoint_prior_scale` = 0.001
- `seasonality_prior_scale` = 10.0



Hình 10: Kết quả mô hình Prophet

**SARIMAX** Mô hình SARIMAX là mở rộng của mô hình ARIMA bằng cách tích hợp các thành phần theo mùa và các biến bên ngoài (ngoại sinh). Nó giải quyết cả xu hướng theo mùa và không theo mùa bằng cách sử dụng các tham số ARIMA và tham số mùa vụ:

*Order (p, d, q):* Đây là các tham số của mô hình ARIMA dùng để mô tả các đặc tính cơ bản của chuỗi thời gian:

- **p (Auto-regressive terms):** Số lượng các giá trị trễ (lag) của chính chuỗi thời gian được sử dụng để dự đoán.
- **d (Differencing order for stationarity):** Số lần lấy sai phân (differencing) để làm cho chuỗi trở nên dừng (stationary).
- **q (Moving average terms):** Số lượng giá trị sai số (error terms) trễ được sử dụng trong mô hình.

**ARIMA Equation:**

$$y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \epsilon_t - \sum_{j=1}^q \theta_j \epsilon_{t-j}$$

Trong đó:

- $y_t$ : Biến mục tiêu tại thời điểm  $t$ .
- $c$ : Hệ số hằng số.
- $\phi_i$ : Hệ số tự hồi quy.
- $\theta_j$ : Hệ số trung bình trượt.
- $\epsilon_t$ : Sai số (nhiều trắng).

Differencing

$$y'_t = y_t - y_{t-1} \quad (\text{if } d = 1)$$

**Trong đó:**

- $y'_t$ : Chuỗi sai phân để đạt tính dừng
- $d$ : Bậc sai phân

**Seasonal Order (P, D, Q, m):** Đây là các tham số mở rộng để mô hình hóa tính chất mùa vụ của chuỗi thời gian:

- **p (P, D, Q):** Tương ứng với các tham số ARIMA (p, d, q) nhưng dành riêng cho thành phần mùa vụ.
- **m (Seasonal period length):** Độ dài chu kỳ mùa vụ (ví dụ:  $m = 12$  cho chu kỳ 12 tháng nếu dữ liệu theo tháng).

**Seasonal Component:**

$$y_t = \sum_{k=1}^P \Phi_k y_{t-km} + \sum_{l=1}^Q \Theta_l \epsilon_{t-lm}$$

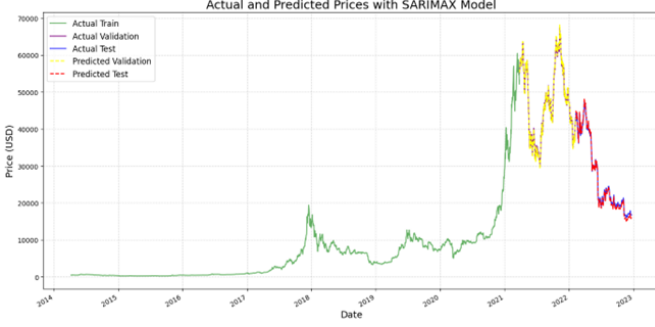
**Trong đó:**

- $\Phi_k, \Theta_l$ : Các hệ số tự hồi quy theo mùa (Seasonal AR) và trung bình trượt theo mùa (Seasonal MA).
- $m$ : Độ dài chu kỳ theo mùa.

Đánh giá nhiều sự kết hợp (p, d, q, P, D, Q, m) để xác định cấu hình tối ưu. Dùng GridSearch để tìm các tham số tốt nhất với phạm vi tham số: p, d, q: 0, 1, 2, P, D, Q:

0, 1, 2, m: 12 (tính theo mùa hàng tháng). RMSE được sử dụng làm số liệu đo hiệu suất.

Tham số của mô hình tốt nhất: order=(1, 2, 2), seasonal\_order=(1, 0, 2, 12)



Hình 11: Kết quả mô hình SARIMAX

**SARIMAX-LSTM** Ghi lại sự phụ thuộc tuyến tính và tính theo mùa trong dữ liệu chuỗi thời gian.

Order (p, d, q): tham số của mô hình ARIMA mô tả các đặc tính cơ bản của chuỗi thời gian.

Seasonal Order (P, D, Q, m): tham số mở rộng mô hình hóa tính chất mùa vụ của chuỗi thời gian.

SARIMAX được huấn luyện với các tham số tốt nhất đã tìm được: order=(1, 2, 2), seasonal\_order=(1, 0, 2, 12).

Dữ liệu phần dư từ dự đoán của SARIMAX được sử dụng làm đầu vào cho mô hình LSTM.

Residuals:

$$\text{Residual}_t = y_t - \hat{y}_t^{\text{SARIMAX}}$$

**Trong đó:**

- $\text{Residual}_t$ : Chênh lệch giữa giá trị dự đoán và thực tế.
- $y_t$ : Giá trị thực tế ở thời gian  $t$ .
- $\hat{y}_t^{\text{SARIMAX}}$ : Giá trị dự đoán bằng SARIMAX ở thời gian  $t$ .

Mô hình LSTM: Ghi lại các mối phụ thuộc phi tuyến tính và các mô hình dài hạn trong phần dư. Nhiều lớp LSTM với các đơn vị có thể điều chỉnh và tỷ lệ dropout. Điều chỉnh siêu tham số bằng cách sử dụng RandomSearch. LSTM Forget Gate:

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

**Trong đó:**

- $f_t$ : Đầu ra của Forget gate.
- $W_f$ : Ma trận trọng số cho forget gate.
- $[h_{t-1}, x_t]$ : Kết hợp giữa trạng thái ẩn trước đó và đầu vào hiện tại..
- $b_f$ : Tham số chệch (Bias) cho forget gate.
- $\sigma$ : hàm kích hoạt Sigmoid.

Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

**Trong đó:**

- $i_t$ : Đầu ra của cổng đầu vào (Input gate).
- $\tilde{C}_t$ : Trạng thái tế bào ứng viên (Candidate cell state).
- $W_i, W_C$ : Ma trận trọng số cho cổng đầu vào và trạng thái candidate cell.
- $b_i, b_C$ : Các tham số chệch (Bias terms).
- $\tanh$ : Hàm kích hoạt hyperbolic tangent..

Cell State Update:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

**Trong đó:**

- $C_t$ : Trạng thái tế bào hiện tại.
- $C_{t-1}$ : Trạng thái tế bào trước đó.
- $\odot$ : Phép nhân từng phần tử (Element-wise multiplication).

Output Gate:

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

$$h_t = o_t \odot \tanh(C_t)$$

**Trong đó:**

- $o_t$ : Cổng đầu ra (Output gate).
- $h_t$ : Trạng thái ẩn (Hidden state).
- $W_o$ : Ma trận trọng số cho cổng đầu ra.
- $b_o$ : Tham số chệch (Bias term) cho cổng đầu ra.

Các tham số của mô hình LSTM tốt nhất: num\_lstm\_layers: 1; lstm\_units\_0: 32; dropout\_rate\_0: 0.5; learning\_rate: 0.0004080644937450095; weight\_decay: 6.17885235880975e-05

Phương pháp kết hợp: SARIMAX được huấn luyện trước tiên để dự đoán và tính toán giá trị dư. SARIMAX xử lý các thành phần tuyến tính và theo mùa, trong khi LSTM mô hình hóa các phần dư phi tuyến tính còn lại để cải thiện độ chính xác của dự báo. Combined Prediction:

$$\hat{y}_t = \hat{y}_t^{\text{SARIMAX}} + \hat{y}_t^{\text{LSTM}}$$

**Trong đó:**

- $\hat{y}_t$ : Dự đoán cuối cùng
- $\hat{y}_t^{\text{SARIMAX}}$ : Dự đoán từ SARIMAX.
- $\hat{y}_t^{\text{LSTM}}$ : Dự đoán từ LSTM.



Hình 12: Kết quả mô hình SARIMAX-LSTM



**Transformer** sử dụng cơ chế tự chú ý để mô hình hóa các phụ thuộc toàn cục trong dữ liệu tuần tự một cách hiệu quả. Các thành phần chính bao gồm:

**Self-Attention Mechanism:** Tính toán mức độ liên quan giữa các phần tử của một chuỗi.

**Multi-Head Attention:** Nhiều cơ chế chú ý cho phép thể hiện nhiều tính năng khác nhau.

**Feed-Forward Network (FFN):** Xử lý đầu ra sự chú ý thông qua các lớp dày đặc.

**Layer Normalization and Residual Connections:** Tăng cường tính ổn định và hội tụ.

Cơ chế tự chú ý được tính như sau:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Trong đó:

- $Q$ : Ma trận truy vấn (Query matrix).
- $K$ : Ma trận khóa (Key matrix).
- $V$ : Ma trận giá trị (Value matrix).
- $d_k$ : Kích thước của các khóa (dimensionality of keys)
- Thay vì sử dụng một khối attention, Transformer dùng nhiều attention heads song song:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O$$

Trong đó:

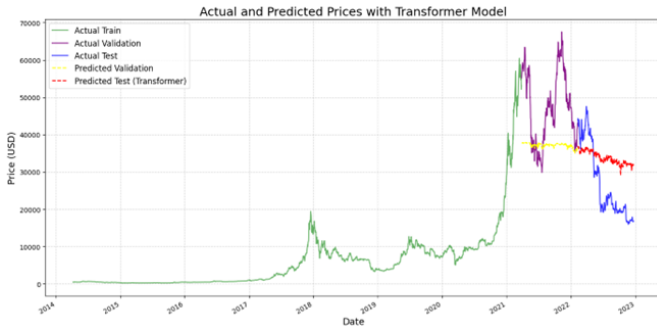
$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$W_i^Q, W_i^K, W_i^V$  là các ma trận tham số có thể học được. Mỗi khối attention được tiếp nối bởi feed-forward network:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$$

$d_k$ : Kích thước của các khóa (dimensionality of keys)

Dùng RandomSearch để thử nghiệm nhiều tham số của mô hình. Mô hình tốt nhất: units: 256, dropout\_rate: 0.5, num\_transformer\_blocks: 2, num\_heads: 3, key\_dim: 48, units\_ff: 256, learning\_rate: 0.00027492722234476815, weight\_decay: 2.7151975248047397e-05



Hình 13: Kết quả mô hình Transformer

## SARIMAX-Transformer

**Order (p, d, q):** Đây là các tham số của mô hình ARIMA dùng để mô tả các đặc tính cơ bản của chuỗi thời gian.

**Seasonal Order (P, D, Q, m):** Đây là các tham số mở rộng để mô hình hóa tính chất mùa vụ của chuỗi thời gian.

SARIMAX được huấn luyện với các tham số tốt nhất đã tìm được: order=(1, 2, 2), seasonal\_order=(1, 0, 2, 12).

Dữ liệu phần dư từ dự đoán của SARIMAX được sử dụng làm đầu vào cho mô hình Transformer.

Residuals:

$$\text{Residual}_t = y_t - \hat{y}_t^{\text{SARIMAX}}$$

**Trong đó:**

- $\text{Residual}_t$ : Chênh lệch giữa giá trị dự đoán và thực tế.
- $y_t$ : Giá trị thực tế ở thời gian  $t$ .
- $\hat{y}_t^{\text{SARIMAX}}$ : Giá trị dự đoán bằng SARIMAX ở thời gian  $t$ .

Mô hình Transformer được sử dụng để huấn luyện các phần dư từ mô hình SARIMAX để nắm bắt các phụ thuộc phi tuyến tính.

**Attention Mechanism:**

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

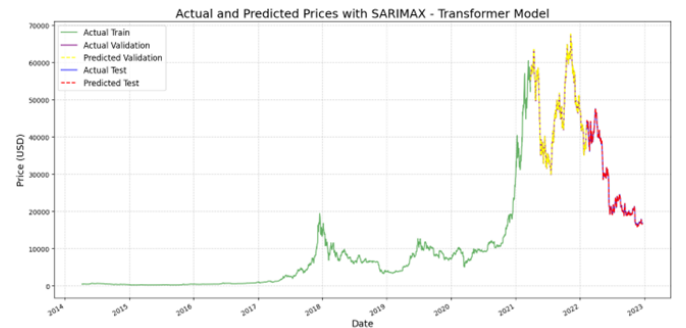
**Feedforward Network:**

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$$

Dùng RandomSearch để thử nghiệm qua các tham số của mô hình Transformer. Tham số tốt nhất của mô hình Transformer dự đoán phần dư: units: 192, dropout\_rate: 0.5, num\_transformer\_blocks: 1, num\_heads: 4, key\_dim: 32, units\_ff: 128, learning\_rate: 0.0006932040937053005, weight\_decay: 1.2171709952301372e-05

Dự đoán cuối cùng kết hợp dự đoán SARIMAX và dự đoán giá trị dư của Transformer. Dự đoán kết hợp:

$$\hat{y}_t = \hat{y}_t^{\text{SARIMAX}} + \hat{y}_t^{\text{Transformer}}$$



Hình 14: Kết quả mô hình SARIMAX-Transformer

**Diffusion-VAE** VAE nhằm mục đích mã hóa dữ liệu vào không gian tiềm ẩn và giải mã dữ liệu trở lại không gian dữ liệu, trong khi đảm bảo phân phối không gian tiềm ẩn xấp xỉ phân phối tiên nghiệm, thường là  $\mathcal{N}(0, 1)$ .

Mục tiêu: Giảm thiểu giá trị ELBO (Evidence Lower Bound):

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}))$$

#### Giải thích:

- $q_\phi(\mathbf{z}|\mathbf{x})$ : Bộ mã hóa (encoder), ước lượng phân phối hậu nghiệm của biến tiềm ẩn  $\mathbf{z}$ .
- $p_\theta(\mathbf{x}|\mathbf{z})$ : Bộ giải mã (decoder), tái tạo lại  $\mathbf{x}$  từ  $\mathbf{z}$ .
- $p(\mathbf{z})$ : Phân phối tiên nghiệm (prior), thường là  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ .
- $D_{\text{KL}}$ : Độ đo phân kỳ Kullback-Leibler đảm bảo  $q_\phi(\mathbf{z}|\mathbf{x})$  gần với  $p(\mathbf{z})$ .

Một quá trình khuếch tán tiến dần thêm nhiễu vào dữ liệu theo từng bước  $t = 0, 1, \dots$ , dần dần chuyển đổi dữ liệu  $\mathbf{x}_0$  thành nhiễu Gaussian đẳng hướng  $\mathbf{x}_T$ . Quá trình đảo ngược nhằm mục đích phục hồi  $\mathbf{x}_0$  từ  $\mathbf{x}_T$  từng bước một.

Quá trình chuyển tiếp được định nghĩa như sau:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

#### Giải thích:

- $x_t$ : Dữ liệu nhiễu tại bước  $t$
- $\bar{\alpha}_t$ : Tích lũy của  $\alpha_t$ , với  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$
- $\alpha_t = 1 - \beta_t$ : Hệ số giảm nhiễu
- $\beta_t$ : Lịch trình nhiễu, tăng tuyến tính từ  $\beta_{\text{start}}$  đến  $\beta_{\text{end}}$

#### Mối quan hệ giữa các bước:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

**Giải thích:** Nhiễu  $\epsilon$  được thêm vào theo tỷ lệ  $\sqrt{1 - \bar{\alpha}_t}$ , đảm bảo quá trình nhiễu diễn ra dần dần.

Kết hợp khả năng biểu đạt của VAE với tính linh hoạt của quá trình khuếch tán giúp mô hình hóa phân phối dữ liệu phức tạp. Bộ mã hóa đưa ra không gian tiềm ẩn  $\mathbf{z}$  có điều kiện trên đầu vào nhiễu  $\mathbf{x}_t$ , trong khi bộ giải mã dự đoán  $\mathbf{x}_0$  từ  $\mathbf{z}$  và bước thời gian  $t$ .

#### Encoder:

- **Input:**  $\mathbf{x}_t$  (nhiễu tại bước  $t$ ), thời gian  $t$ , và đặc trưng  $\mathbf{f}$
- **Output:** Trung bình  $\boldsymbol{\mu}$  và log-phương sai  $\log \boldsymbol{\sigma}^2$

$$\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2 = f_{\text{encoder}}([\mathbf{f}, \mathbf{x}_t, t])$$

#### Reparameterization Trick:

- Lấy mẫu  $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$  bằng cách:
- $$\mathbf{z} = \boldsymbol{\mu} + \epsilon \cdot \boldsymbol{\sigma}, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\sigma} = \exp(0.5 \cdot \log \boldsymbol{\sigma}^2)$$

#### Decoder:

- **Input:** Biểu diễn tiềm ẩn  $\mathbf{z}$ , đặc trưng  $\mathbf{f}$ , và thời gian  $t$
- **Output:** Dữ liệu dự đoán  $\hat{\mathbf{x}}_0$

$$\hat{\mathbf{x}}_0 = f_{\text{decoder}}([\mathbf{z}, \mathbf{f}, t])$$

#### Hàm mất mát của Diffusion-VAE:

$$\mathcal{L} = \|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|^2 + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}_t) \| p(\mathbf{z}))$$

#### Giải thích:

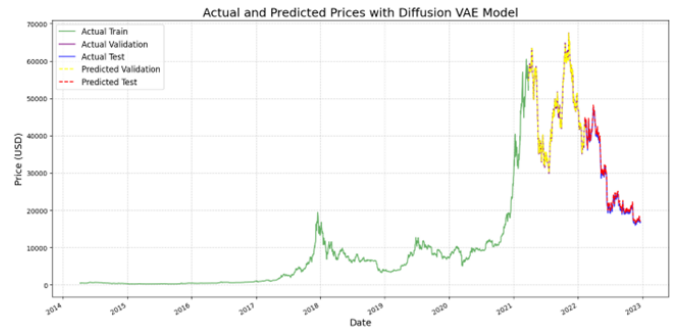
- $\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|^2$ : Mất mát tái tạo (Reconstruction Loss - MSE)
- $D_{\text{KL}}$ : Đảm bảo  $q_\phi(\mathbf{z}|\mathbf{x}_t)$  gần với  $p(\mathbf{z})$

#### Thông số khuếch tán:

- Lịch trình nhiễu:  $\beta_t$  tăng tuyến tính từ  $\beta_{\text{start}}$  đến  $\beta_{\text{end}}$
- Số bước khuếch tán:  $T = 1000$

#### Cấu hình huấn luyện thử nghiệm tốt nhất:

- Learning rate: 0.001
- Batch size: 32
- Latent dimension:  $\dim(\mathbf{z}) = 16$
- Hidden dimensions: 64
- Optimizer: Adam



Hình 15: Kết quả mô hình Diffusion-VAE

### 3.3 Phương pháp đánh giá

Trong nghiên cứu này, các mô hình dự đoán được đánh giá theo ba tiêu chí: **MAE**, **MAPE** và **RMSE**.

Trong các công thức sau,  $X_i$  là giá trị dự đoán tại thời điểm  $i$ , và  $Y_i$  là giá trị thực tế tại thời điểm  $i$ . Phương pháp hồi quy sẽ dự đoán giá trị  $X_i$  tương ứng với giá trị thực tế  $Y_i$  trong tập dữ liệu gốc [?].

**MAE** (Mean Absolute Error - Sai số tuyệt đối trung bình) là thước đo hiệu suất hoặc độ chính xác của mô hình dự đoán, được tính bằng giá trị trung bình của sai số tuyệt đối giữa giá trị thực tế và giá trị dự đoán:

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |X_i - Y_i|$$

**MAPE** (Mean Absolute Percentage Error - Sai số phần trăm tuyệt đối trung bình) là thước đo hiệu suất hoặc độ chính xác của mô hình dự đoán, được tính bằng phần trăm trung bình của sai số tuyệt đối so với giá trị thực tế:

$$\text{MAPE} = \frac{1}{m} \sum_{i=1}^m \left| \frac{Y_i - X_i}{Y_i} \right|$$

*RMSE (Root Mean Square Error - Sai số bình phương trung bình căn bậc hai)* là thước đo hiệu suất hoặc độ chính xác của mô hình dự đoán, được tính bằng căn bậc hai của trung bình sai số bình phương giữa giá trị dự đoán và giá trị thực tế:

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (X_i - Y_i)^2}$$

## 4 Kết quả

Bảng 1: So sánh hiệu suất các mô hình dự đoán giá Bitcoin

Mô hình	MAE	RMSE	MAPE (%)
LSTM	596.77	835.25	2.2800
BiLSTM	457.78	682.70	1.7900
ARIMA	16 942.97	19 339.87	81.79
GRU	554.37	747.65	2.3300
GPR	194.01	255.70	0.7000
LightGBM-LSTM	502.36	723.19	1.9300
SVR	1 168.29	1 436.59	4.6100
RNN	998.81	1 316.21	3.7417
Prophet	308.77	417.52	1.1300
SARIMAX	498.66	545.51	2.2600
SARIMAX-LSTM	0.64	0.6627	0.0028
Transformer	9 712.76	10 550.84	44.9062
SARIMAX-Transformer	86.64	94.11	0.3746
Diffusion-VAE	552.71	602.58	2.3268

LSTM: MAE là 596,77, RMSE là 835,25 và MAPE là 2,28%. LSTM xử lý hiệu quả các phụ thuộc tuần tự, phù hợp với dữ liệu chuỗi thời gian. Tuy nhiên, nó gặp khó khăn với độ biến động và nhiễu cực độ vốn có trong giá Bitcoin. Nó hoạt động khá tốt nhưng vẫn không thể vượt qua BiLSTM và các mô hình hybrid.

BiLSTM: MAE là 457,78, RMSE là 682,70 và MAPE là 1,79%. BiLSTM tăng cường độ hiểu biết ngữ cảnh bằng cách xử lý dữ liệu theo cả hướng thuận và hướng ngược, cải thiện LSTM. Hiệu suất của nó làm nổi bật lợi thế của việc sử dụng kiến trúc song hướng để dự báo chuỗi thời gian.

ARIMA: MAE là 16.942,97, RMSE là 19.339,87 và MAPE là 81,79%. Việc ARIMA không thể nắm bắt được các mô hình phi tuyến tính và sự thay đổi giá đột ngột trong các tập dữ liệu có tính biến động cao khiến nó không phù hợp để dự báo Bitcoin. Việc dựa vào các mối quan hệ tuyến tính hạn chế độ chính xác của nó.

GRU: MAE là 554,37, RMSE là 747,65 và MAPE là 2,33%. GRU, đơn giản hơn về mặt tính toán so với LSTM, có hiệu suất tương đương nhưng vẫn không thể sánh được với BiLSTM hoặc các mô hình kết hợp.

GPR: MAE là 194,01, RMSE là 255,70 và MAPE là 0,70%. GPR nổi bật giữa các mô hình độc lập do khả

năng mô hình hóa sự không chắc chắn và thích ứng với các xu hướng phi tuyến tính phức tạp. Tuy nhiên, nó có thể không mở rộng tốt cho các tập dữ liệu lớn hơn.

LightGBM-LSTM: MAE là 502,36, RMSE là 723,19 và MAPE là 1,93%. Kết hợp sức mạnh của LightGBM trong việc học tính năng với mô hình tuần tự của LSTM cải thiện hiệu suất. Tuy nhiên, mô hình kết hợp này kém hiệu quả hơn so với các kết hợp dựa trên SARIMAX.

SVR: MAE là 1.168,29, RMSE là 1.436,59 và MAPE là 4,61%. SVR gặp khó khăn trong việc khái quát hóa dữ liệu chuỗi thời gian không ổn định.

RNN: MAE là 998,81, RMSE là 1.316,21 và MAPE là 3,74%. Kiến trúc RNN cơ bản gặp phải vấn đề về độ dốc biến mất và thiếu sự tinh vi để mô hình hóa các mối phụ thuộc dài hạn, dẫn đến các dự đoán không tối ưu.

Prophet: MAE là 308,77, RMSE là 417,52 và MAPE là 1,13%. Prophet được thiết kế cho mô hình theo mùa và xu hướng, Prophet hoạt động tốt cho các tác vụ chuỗi thời gian chung. Tuy nhiên, nó vẫn chưa thể mang lại độ chính xác cao cho dự đoán khi đối mặt với sự biến động phi tuyến tính của Bitcoin.

SARIMAX: MAE là 498,66, RMSE là 545,51 và MAPE là 2,26%. SARIMAX mô hình hóa hiệu quả các thành phần theo mùa và xu hướng, mang lại độ chính xác khá. Bản chất tuyến tính của nó hạn chế khả năng nắm bắt các biến động giá phức tạp.

SARIMAX-LSTM: MAE là 0,64, RMSE là 0,6627 và MAPE là 0,0028%. Mô hình này kết hợp mô hình xu hướng của SARIMAX với khả năng phi tuyến tính của LSTM, đạt được độ chính xác gần như hoàn hảo. Sự thành công của nó xác nhận sức mạnh của các phương pháp tiếp cận kết hợp.

Transformer: MAE là 9.712,76, RMSE là 10.550,84 và MAPE là 44,91%. Hiệu suất kém của Transformer là điều đáng ngạc nhiên, xét đến sức mạnh của nó trong các tác vụ tuần tự. Transformer hoạt động tốt trên các chuỗi dữ liệu dài và giàu quan hệ, chẳng hạn như văn bản hoặc chuỗi phân loại. Tuy nhiên, dữ liệu giá Bitcoin là chuỗi thời gian thường bị nhiễu cao, có tính không dừng và có xu hướng thay đổi đột ngột.

Transformer-SARIMAX: MAE là 86,64, RMSE là 94,11 và MAPE là 0,3746%. Bằng cách kết hợp mô hình tuyến tính có cấu trúc của SARIMAX với khả năng học tập còn lại của Transformer, mô hình lai này đạt được kết quả đặc biệt so với mô hình Transformer đơn lẻ.

Diffusion-VAE (Bộ mã hóa tự động biến đổi): MAE là 552,71, RMSE là 602,58 và MAPE là 2,33%. Diffusion-VAE, mặc dù mang tính sáng tạo, nhưng không vượt trội hơn các mô hình lai đơn giản hơn. Tiềm năng của nó nằm ở khả năng tinh chỉnh và tối ưu hóa tốt hơn.

## 5 Kết luận

ARIMA hoạt động dựa trên giả định về mối quan hệ tuyến tính giữa các quan sát qua thời gian. Mô hình này chỉ có thể nắm bắt được các pattern tuyến tính trong dữ liệu, bao gồm xu hướng (trend) và sự tự tương quan giữa các điểm dữ liệu liên tiếp. Tuy nhiên, giá Bitcoin thể hiện

những đặc tính phức tạp và biến động mạnh mà ARIMA không thể xử lý hiệu quả. SARIMAX, với khả năng tích hợp biến ngoại sinh (exogenous variables), giúp mô hình có thể tận dụng thêm thông tin như khối lượng giao dịch hoặc dữ liệu kinh tế vĩ mô. Tuy nhiên, khả năng phi tuyến của nó vẫn hạn chế, khiến nó không đủ mạnh trong môi trường tài chính biến động cao.

LSTM và các biến thể rất phù hợp với dữ liệu chuỗi thời gian, đặc biệt là khi có các mẫu phi tuyến phức tạp. Khả năng ghi nhớ trạng thái giúp chúng nắm bắt được các quan hệ dài hạn trong dữ liệu.

Prophet được thiết kế cho dữ liệu chuỗi thời gian có tính xu hướng và mùa vụ rõ ràng. Nó hoạt động hiệu quả với dữ liệu tài chính có chu kỳ, nhưng không đủ linh hoạt để xử lý các thay đổi đột ngột hoặc phi tuyến cao.

SVR hoạt động tốt trên dữ liệu nhỏ và tương đối đơn giản. Tuy nhiên, nó gặp khó khăn khi dữ liệu có tính phi tuyến cao hoặc biến động lớn, như giá Bitcoin.

GPR mạnh mẽ trong việc mô hình hóa phi tuyến, đặc biệt là trên dữ liệu nhỏ. Tuy nhiên, GPR không mở rộng tốt cho dữ liệu lớn và đòi hỏi tính toán cao, làm nó ít phù hợp với các tập dữ liệu lớn hoặc có nhiều đặc trưng.

Diffusion-VAE sử dụng phương pháp khuếch tán để mô hình hóa dữ liệu, phù hợp với các bài toán phi tuyến và có nhiễu. Phương pháp khuếch tán (diffusion process) được sử dụng để thêm một lượng nhiễu có kiểm soát vào dữ liệu, làm cho dữ liệu trở nên ngẫu nhiên hơn ở mỗi bước. Quá trình này được mô phỏng bằng cách học cách tái cấu trúc lại dữ liệu ban đầu từ trạng thái nhiễu hoàn toàn. Tuy nhiên, mô hình này phức tạp và đòi hỏi lượng dữ liệu lớn để phát huy tối đa hiệu quả.

Mô hình Transformer phụ thuộc vào cấu trúc dữ liệu và điều chỉnh thông số phù hợp, mô hình mang lại kết quả không tốt khi hoạt động riêng lẻ, điều này cho thấy nó không phù hợp cho bài toán này nếu không có sự cải tiến hoặc lai tạo. SARIMAX-Transformer thể hiện tiềm năng tăng cường tính hiệu quả cho Transformer bằng các mô hình chuỗi thời gian có cấu trúc.

Việc sử dụng các mô hình kết hợp (SARIMAX-LSTM và SARIMAX-Transformer) rất quan trọng để nắm bắt cả khía cạnh tuyến tính và phi tuyến tính của dữ liệu giá Bitcoin. Các mô hình kết hợp luôn vượt trội hơn các phương pháp độc lập bằng cách tích hợp các điểm mạnh của các thuật toán.

Kết quả các mô hình nhấn mạnh tầm quan trọng của kiến trúc hybrid trong việc giải quyết các tập dữ liệu phức

tạp, phi tuyến tính và dễ biến động như giá Bitcoin. Các mô hình kết hợp mang lại kết quả vượt trội bằng cách tận dụng các điểm mạnh bổ sung, đem lại hướng đề xuất mới trong các phương pháp dự báo dữ liệu có độ biến động cao như giá tiền điện tử, giá cổ phiếu. Nghiên cứu trong tương lai tập trung vào việc tinh chỉnh các mô hình kết hợp, khám phá khả năng mở rộng và khái quát hóa của chúng đối với các tập dữ liệu tiền ảo nói riêng và tài chính nói chung.

## Tài liệu

1. Hu, R., Wang, X.: Linkage Analysis Between Bitcoin and Nasdaq Index Based on ARIMAX Model. In: Proc. ICMEIM 2022 (2022). <https://www.atlantis-press.com/proceedings/icmeim-22/125980282>
2. Seabe, B., Smith, R., Doe, T.: Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. *Future Internet* **7**(2), 203–214 (2023). <https://www.mdpi.com/2504-3110/7/2/203>
3. Pabuccu, H., Ongan, S., Ongan, A.: Forecasting the Movements of Bitcoin Prices: An Application of Machine Learning Algorithms. *arXiv preprint* (2023). <https://arxiv.org/abs/2303.04642>
4. Achmadi, G.R., Saikhu, A., Amaliah, B.: Cryptocurrency Price Movement Prediction Using the Hybrid SARIMAX-LSTM Method. In: Proc. 2023 Int. Conf. on Advanced Machine Intelligence and Machine Interaction Applications (ICAMIMIA) (2023). <https://scholar.its.ac.id/en/publications/cryptocurrency-price-movement-prediction-using-the-hybrid-sarimax>
5. Tripathy, N., Dash, P.P., Mishra, S.K.: Performance Analysis of Bitcoin Forecasting Using Deep Learning Techniques. *Indonesian Journal of Electrical Engineering & Computer Science*, **31**(3), 1515–1522 (2023). [https://www.researchgate.net/publication/372786826\\_Performance\\_analysis\\_of\\_bitcoin\\_forecasting\\_using\\_deep\\_learning\\_techniques](https://www.researchgate.net/publication/372786826_Performance_analysis_of_bitcoin_forecasting_using_deep_learning_techniques)
6. Angelovski, D., Velichkovska, B., Jakimovski, G., Efnusheva, D., Kalendar, M.: Machine Learning-Based Forecasting of Bitcoin Price Movements. In: Proc. 12th Int. Conf. on Applied Innovation in IT (ICAIIIT), pp. 76–80 (2024). [https://repo.bibliothek.uni-halle.de/bitstream/1981185920/117598/1/2\\_1\\_ICAIIIT\\_2024\\_Part\\_2\\_paper\\_19.pdf](https://repo.bibliothek.uni-halle.de/bitstream/1981185920/117598/1/2_1_ICAIIIT_2024_Part_2_paper_19.pdf)
7. Koa, T., et al.: Diffusion Variational Autoencoder for Tackling Stochasticity in Multi-Step Regression Stock Price Prediction. *arXiv preprint* (2023). <https://arxiv.org/abs/2309.00073>