DỰ ĐOÁN GIÁ ĐỒNG TIỀN ĐIỆN TỬ BITCOIN

1^{st} Nguyễn Mạnh Đức
Đại học CNTT- ĐHQG Tp.HCM
Ngành: Khoa học dữ liệu
MSSV: 20521196
Email: 20521196@gm.uit.edu.vn
Tp.Hồ Chí Minh, Việt Nam

2^{nd} Huỳnh Lê Phương Vy
Đại học CNTT- ĐHQG Tp.HCM
Ngành: Khoa học dữ liệu
MSSV: 20520951
Email: 20520951@gm.uit.edu.vn
Tp.Hồ Chí Minh, Việt Nam

3^{rd} Văn Ngọc Nhật Huy
Đại học CNTT- ĐHQG Tp.HCM
Ngành: Khoa học dữ liệu
MSSV: 20521418
Email: 20521418@gm.uit.edu.vn
Tp.Hồ Chí Minh, Việt Nam

Tóm tắt nội dung—Tài liệu này là bản báo cáo chi tiết về đồ án cuối kì môn Học máy thống kê. Bản báo cáo trình bày nội dung về việc áp dụng ba mô hình học máy là Random Forest, XGBoost, Long Short-term memory sử dụng cho việc dự đoán giá của đồng tiền điện tử Bitcoin trong tương lai dựa trên dữ liệu giá có sẵn trong quá khứ. Đánh giá mô hình và so sánh khả năng dự đoán giữa các mô hình học máy qua các chỉ số Mean Absolute Error, Mean Squared Error, Mean Absolute Percent Error, R2 Score. Kiểm tra dữ liệu trên tập đào tạo và tập kiểm tra, tính toán các chỉ số theo từng khoảng dữ liệu để tìm hiểu nguyên nhân có sự chênh lệch lớn giữa giá dự đoán và giá thực tế. Tinh chỉnh các giá trị tham số đầu vào của ba mô hình học máy bằng cách sử dụng GridSearchCV để giúp mô hình có khả năng dự đoán chính xác hơn. Cũng như đề xuất một số hướng phát triển thêm.

Từ khóa—Random Forest, LSTM, XGBoost, GridSearchCV, dự đoán giá Bitcoin, tinh chỉnh mô hình, MAE, MSE, MAPE, R2 Score.

I. GIỚI THIỆU

Thị trường tài chính là thị trường tiềm ẩn nhiều rủi ro và đầy thách thức. Cryptocurrency bùng nổ trong thời gian qua và đã được đông đảo mọi người xem là kênh đầu tư hấp dẫn. Đầu tư vào Bitcoin có khả năng đem lại lợi nhuận lớn cho nhà đầu tư. Việc sử dụng mô hình học máy giúp dự đoán giá Bitcoin trong tương lai dựa trên dữ liệu giá trong quá khứ giúp nhà đầu tư quyết định mua hay bán, điều chỉnh danh mục đầu tư hợp lý, từ đó tối ưu hóa lợi nhuận và đem lại khoản sinh lời lớn cho bản thân. Trong bản báo cáo này,dữ liệu đầu vào cho mô hình học máy dựa vào dữ liệu cột "Close" trong quá khứ, dữ liệu đầu vào sẽ được mô tả chi tiết trong chương II, phần C. Nhóm lựa chọn dựa đoán giá trong 30 ngày tiếp theo tính từ ngày cuối cùng của bộ dữ liệu

II. BỘ DỮ LIỆU

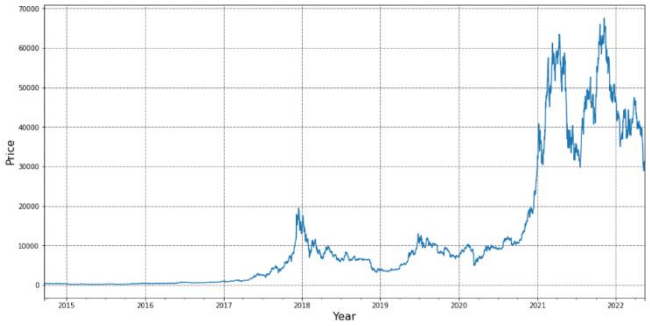
A. *Tổng quan về bộ dữ liệu*

Bitcoin đã xuất hiện từ năm 2009, tuy nhiên dữ liệu chỉ có từ 17-09-2014 nên bộ dữ liệu sẽ được lấy bắt đầu từ ngày này. Bộ dữ liệu khi mới tải về được lưu dưới dạng file .csv, gồm 2800 dòng và 7 cột ứng với 7 thuộc tính là: "Date", "Open", "High", "Low", "Close","Adj Close", "Volume". Tuy nhiên, nhóm chỉ sử dụng dữ liệu từ cột "Close" làm dữ liệu đầu vào, vậy nên chỉ giữ lại hai thuộc tính là "Date" và "Close", sau đó nhóm chuyển cột "Date" thành cột đếm (Index). Bảng I là Codebook mô tả thông tin tổng quan về bộ dữ liệu sau khi nhóm chỉnh sửa từ bộ dữ liệu gốc:

STT	Thông tin	Mô tả
1	Tên bộ dữ liệu	BTC-USD
2	Nguồn thu thập	https://finance.yahoo.com/
3	Số thuộc tính	Có 01 thuộc tính
4	Thông tin thuộc tính	Close: Giá đóng cửa

B. *Phân tích sơ bộ về bộ dữ liệu*

Bộ dữ liệu thể hiện giá đóng của Bitcoin trong quá khứ (tính theo $) được lấy bắt đầu từ ngày 17-09-2014 đến ngày 17-05-2022. Gồm 2800 dòng tương ứng với 2800 ngày liên tiếp nhau, mỗi dòng cách nhau 1 ngày, không có giá trị "Null". Hình IIB.1 thể hiện đồ thị giá Bitcoin dựa trên bộ dữ liệu mà nhóm sử dụng.



Thuộc tính "Close" được lưu dưới dạng float64, làm tròn đến 6 chữ số sau dấu phẩy. Dữ liệu giá Bitcoin trong dữ liệu nằm trong khoảng giá trị là min = 178.102997$ và max = 67566.828125$. Độ chênh lệch giữa giá trị min và max khá lớn (xấp xỉ 380 lần) ảnh hưởng trực tiếp đến khả năng dự đoán của mô hình học máy, vấn đề này sẽ được xử lý trong mục tiếp theo (Chương II, phần C).

Bộ dữ liệu mà nhóm sử dụng chính là dữ liệu chuỗi thời gian (Time series data). Dữ liệu chuỗi thời gian, còn được gọi là dữ liệu đóng dấu thời gian, là một chuỗi các điểm dữ liệu được lập chỉ mục theo thứ tự thời gian[1]. Các điểm dữ liệu này thường bao gồm các phép đo liên tiếp được thực hiện từ cùng một nguồn trong một khoảng thời gian và được sử dụng để theo dõi sự thay đổi theo thời gian[1].

Các thành phần trong chuỗi thời gian (Time series Components) hay các mẫu trong chuỗi thời gian (Time series patterns) bao gồm[2]:

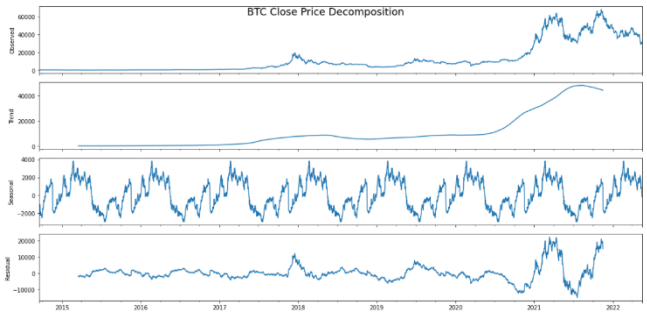
Xu hướng (Trend - T): thể hiện chiều biến động tăng hay giảm của đối tượng nghiên cứu trong một khoảng thời gian dài[2].

Mùa vụ (Seasonal - S): được nhận biết qua các dấu hiệu tăng, giảm của đối tượng nghiên cứu lặp đi lặp lại giống nhau trong các khoảng thời gian liên tiếp[2].

Chu kỳ (Cyclical - C): tồn tại nếu nó hiển thị một chuỗi xen kẽ các điểm bên dưới và bên trên đường xu hướng (tăng và giảm lặp lại)kéo dài hơn một năm[2].

Ngẫu nhiên/Bất thường (Irregular - I): các biến động ngẫu nhiên trong ngắn hạn không lường trước hay dự báo được[2]. Cả 4 thành phần trên tạo thành mô hình chuỗi dữ liệu tổng quan.

Hình dưới mô phỏng về 4 thành phần trong chuỗi thời gian được tạo ra từ dữ liệu mà nhóm sử dụng:



Nhóm sử dụng mô hình Additive với chu kỳ là 365 ngày để mô phỏng dữ liệu. Có thể thấy dữ liệu có xu hướng tăng theo thời gian, tuy nhiên dữ liệu vẫn mang yếu tố ngẫu nhiên cao, và yếu tố ngẫu nhiên tăng phức tạp ở năm 2021, 2022.

C. *Dữ liệu đầu vào và đầu ra cho mô hình học máy*

Để đảm bảo việc so sánh, đánh giá giữa các mô hình với nhau, nhóm sẽ sử dụng chung một kiểu dữ liệu đầu vào. Bộ dữ liệu được chia làm 2 phần:

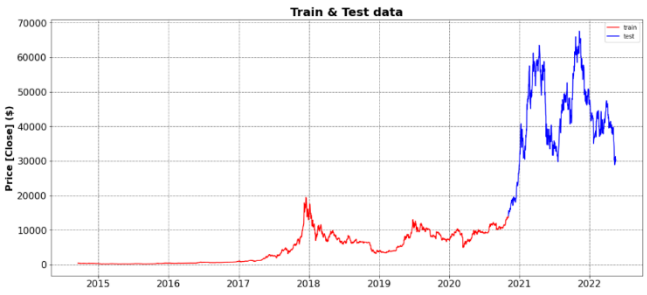
- Train (80%): là tập huấn luyện cho mô hình học máy, gồm 2240 dòng, từ ngày 17-09-2014 đến ngày 03-11- 2020.

- Test (20%): là tập kiểm tra khả năng dự đoán của mô hình, gồm 560 dòng, từ ngày 04-11-2020 đến ngày 17-05-2020.

Các biến được đo lường ở các tỷ lệ khác nhau không đóng góp như nhau vào chức năng đã học của mô hình và có thể tạo ra sai lệch của dự đoán [3]. Vì vậy, để đối phó với vấn đề tiềm ẩn này, nhóm sử dụng hàm MinMaxScaler để đưa dữ liệu về phạm vi [0,1] trước khi đẩy dữ liệu vào mô hình học máy, sau khi dự đoán sẽ chuyển về lại giá trị ban đầu [3]. Công thức toán học hàm chuẩn hóa:

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Đối với dữ liệu đầu vào cho mô hình, nhóm sử dụng tham số time_step = 30, thì ứng với một mảng gồm giá đóng trong 30 ngày liên tiếp, sẽ dự đoán giá đóng trong ngày tiếp theo của ngày thứ 30 trong mảng. Ví dụ: Với time_step = 3, X_train = [[1,2,3],[2,3,4],[3,4,5],[4,5,6]], y_train = [4,5,6,7], sử dụng [1,2,3] dự đoán ra 4, sử dụng [2,3,4] dự đoán ra 5,... Dưới đây là hình ảnh tập train-test sau khi nhóm chia với tỉ lệ 80-20:



III. PHƯƠNG PHÁP MÁY HỌC

Như đã đề cập ở trên, trong báo cáo này nhóm sẽ tập trung vào ba mô hình học máy là Random Forest, XGBoost và Long short term memory và các giá trị tham số cho từng mô hình. Sử dụng bốn độ đo để đánh giá các mô hình học máy là Mean Absolute Error, Mean Squared Error, Mean Absolute Percent Error, R2 Score.

A. *Mô hình học máy*

1) *Mô hình Random Forest*

Trước khi đi vào thuật toán Random Forest, nhóm sẽ trình bày cơ bản về Decision Trees (Cây quyết định) để có thể hiểu về thuật toán Random Forest. Decision Tree là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật [4]. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary) , Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal [4].

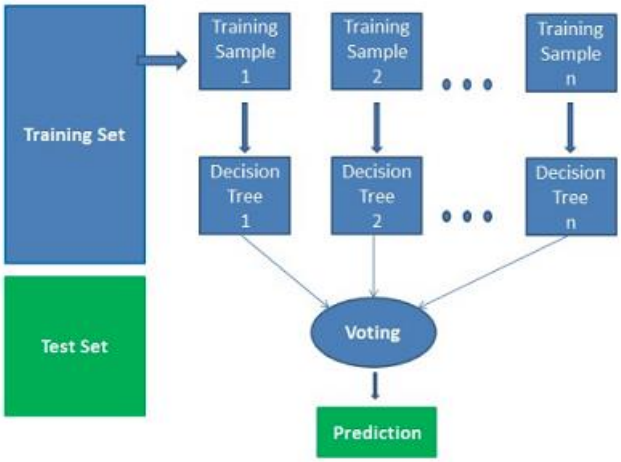
Random Forest là thuật toán học có giám sát, có thể giải quyết cả bài toán phân loại và hồi quy. Thuật toán Random Forest sử dụng nhiều cây quyết định (Decision Trees) tổng hợp để giúp đưa ra dự đoán ổn định và chính xác hơn [5]. Nó hoạt động theo bốn bước:

Bước 1. Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.

Bước 2. Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi quyết định cây.

Bước 3. Hãy bỏ phiếu cho mỗi kết quả dự đoán.

Bước 4. Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng.

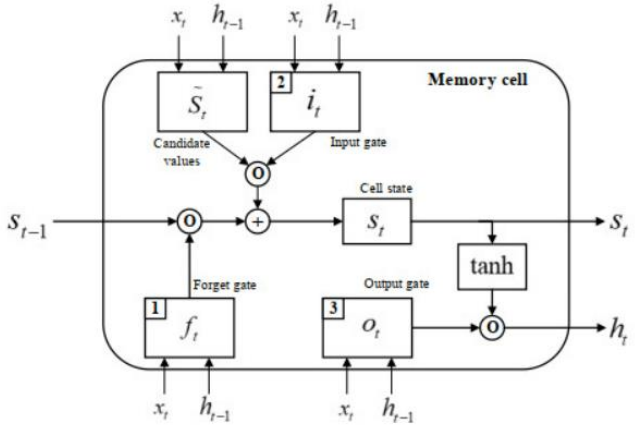


2) *Mô hình XGBoost*

XGBoost là phiên bản có thể mở rộng và được cải tiến của thuật toán tăng độ dốc được thiết kế cho hiệu quả, tốc độ tính toán và hiệu suất mô hình[6]. XGBoost là sự pha trộn hoàn hảo giữa các khả năng phần mềm và phần cứng được thiết kế để nâng cao các kỹ thuật tăng cường hiện có với độ chính xác trong thời gian ngắn nhất [6].

3) *Mô hình Long Short-term memory:*

Là một loại mạng thần kinh tái phát có khả năng học tập phụ thuộc vào thứ tự trong các vấn đề dự đoán trình tự [7].Long short term memory là một phiên bản mở rộng của mạng Recurrent Neural Network (RNN), nó được thiết kế để giải quyết các bài toán về phụ thuộc xa (long-term dependencies) [8].



Mạng LSTM có thể bao gồm nhiều tế bào LSTM liên kết với nhau. Ý tưởng của LSTM là bổ sung thêm trạng thái bên trong tế bào (cell internal state) st và ba cổng sàng lọc thông tin đầu vào và đầu ra cho tế bào bao gồm cổng quên ft, cổng đầu vào it và cổng đầu ra ot [8]. Tại mỗi bước thời gian t, các cổng lần lượt nhận giá trị đầu vào xt đại diện cho một phần tử trong chuỗi đầu vào) và giá trị ht-1 có được từ đầu ra của các ô nhớ từ bước thời gian trước đó t – 1 [8]. Các cổng đều có chức năng sàng lọc thông tin với mỗi mục đích khác nhau. Các cổng được định nghĩa như sau:

Cổng quên: Có chức năng loại bỏ những thông tin không cần thiết nhận được khỏi trạng thái tế bào bên trong [8].

Cổng đầu vào: Giúp sàng lọc những thông tin cần thiết để được thêm vào trạng thái tế bào bên trong [8].

Cổng đầu ra: Có chức năng xác định những thông tin nào từ các trạng thái tế bào bên trong được sử dụng như đầu ra [8].

Trong quá trình thực hiện, st và các giá trị đầu ra ht được tính toán như sau:

Ở bước đầu tiên, tế bào LSTM quyết định những thông tin cần được loại bỏ từ các trạng thái tế bào bên trong ở bước thời gian trước đó st-1. Giá trị ft của cổng quên tại bước thời gian t được tính toán dựa trên giá trị đầu vào hiện tại xt, giá trị đầu ra ht-1 từ tế bào LSTM ở bước trước đó và độ lệch (bias) bf của cổng quên. Hàm sigmoid biến đổi tất cả các giá trị kích hoạt (activation value) về miền giá trị trong khoảng từ 0 và 1 theo công thức [8]:

ft = σ(Wf, xxt + Wf, hht-1 + bf)

Ở bước thứ 2, tế bào LSTM xác định những thông tin nào cần được thêm vào các trạng thái tế bào bên trong st. Bước này bao gồm hai quá trình tính toán đối với st và ft. st biểu diễn những thông tin có thể được thêm vào các trạng thái tế bào bên trong [8]:

st = tanh(Ws,xxt +Ws, hht-1 + bs)

Giá trị it của cổng đầu vào tại bước thời gian t được tính [8]:

it = σ(Wi,xxt + Wi, hht-1 + bi)

Ở bước tiếp theo, giá trị mới của trạng thái tế bào bên trong st được tính toán dựa trên kết quả thu được từ các bước trên [8]:

st = ft*st - 1 + it*st

Cuối cùng, giá trị đầu ra ht: ot = σ(Wo,xxt + Wo, hht-1 + bo) ht = ot*tanh(st) Trong đó: Ws,x, Ws,h, Wf,x Wf,h, Wi,x, Wi,h là các ma trận trọng số trong mỗi tế bào LSTM, bf, bs, bi, bo là các vector bias [8].

4) Các giá trị tham số cho mô hình học máy sử dụng để tinh chỉnh mô hình

Các giá trị tham số dùng cho mô hình học máy sẽ được sử dụng là mặc định. Các giá trị này được sửa đổi sẽ được trình bày tại phần sau (Chương IV-Các thử nghiệm tinh chỉnh mô hình)

- Random Forest:

n_estimators: Số lượng cây trong rừng ngẫu nhiên [5].

max_features: Số lượng các tính năng để xem xét khi tìm kiếm sự phân chia tốt nhất [5].

max_depth: Độ sâu tối đa của cây. Nếu không có, thì các nút được mở rộng cho đến khi tất cả các lá nguyên chất hoặc cho đến khi tất cả các lá chứa nhỏ hơn các mẫu [5].

min_samples_split: Số lượng mẫu tối thiểu cần thiết để chia một nút nội bộ[5].

min_samples_leaf: Số lượng mẫu tối thiểu cần có ở nút lá. Một điểm phân tách ở bất kỳ độ sâu nào sẽ chỉ được xem xét nếu nó để lại ít nhất các mẫu đào tạo min_samples_leaf ở mỗi nhánh trái và phải [5].

- XGBoost: Ngoài việc sử dụng tham số n_estimators, max_depth, nhóm sử dụng thêm 3 tham số là:

learning_rate: Tốc độ học tập xác định kích thước bước ở mỗi lần lặp trong khi mô hình tối ưu hóa theo mục tiêu của nó [9].

colsample_bytree: Đại diện cho phần cột được lấy mẫu ngẫu nhiên cho mỗi cây[9].

subsample: Đại diện cho phần quan sát được lấy mẫu cho mỗi cây[9].

- Long short term memory:

epochs: là một hyperparameter trong ANN, được dùng để định nghĩa số lần learning algorithm hoạt động trên model, một epoch hoàn thành là khi tất cả dữ liệu training được đưa vào mạng neural network một lần (đã bao gồm cả 2 bước forward và backward cho việc cập nhật internal model parameters)[10].

batch_size: một tập training dataset có thể được chia nhỏ thành các batches (sets, parts). Một batch sẽ chứa các training samples, và số lượng các samples này được gọi là batch_size[10].

hidden_layers: được gọi là fully connected layer, tên gọi theo đúng ý nghĩa, mỗi node trong hidden layer được kết nối với tất cả các node trong layer trước[11].

B. Phương pháp đánh giá

Để đánh giá, tỉnh chỉnh giúp cải thiện khả năng dự đoán của mô hình học máy, nhóm sử dụng sai số dự báo (Forecast error) qua các chỉ số Mean Squared Error, Mean Absolute Error, Mean Absolute Percent Error và R2 Score. Sai số dự báo là chênh lệch giữa giá trị thực (dữ liệu) và giá trị dự báo nhằm đánh giá chất lượng hay sự phù hợp của mô hình dự báo tại cùng một thời điểm [12]. Công thức tính sai số dự báo:

$$\varepsilon_t = Y_t - \hat{Y}_t$$

Trong đó:

ε_t: sai số dự báo tại thời điểm t.

Y_t: giá trị thực tế tại thời điểm t.

$\hat{Y}_t$: giá trị dự đoán tại thời điểm t (ứng với quan sát t).

1) Mean Squared Error:

Trong thống kê, Mean Squared Error (MSE - sai số bình phương trung bình) của công cụ ước tính (của thủ tục ước tính số lượng không quan sát được) đo trung bình bình phương của các lỗi – nghĩa là chênh lệch bình phương trung bình giữa các giá trị dự đoán và giá trị gốc[13]. MSE là một hàm rủi ro, tương ứng với giá trị dự kiến của mất lỗi bình phương. Công thức tính MSE:

$$MSE = \frac{\sum_{t=1}^{n} \varepsilon_t^2}{n} = \frac{\sum_{t=1}^{n} (Y_t - \hat{Y}_t)^2}{n}$$

2) Mean Absolute Error:

Mean Absolute Error (MAE) - sai số tuyệt đối trung bình là một phương pháp đo lường sự khác biệt giữa hai biến liên tục. Chúng ta có độ đo MAE được tính theo công thức sau:

$$MAE = \frac{\sum_{t=1}^{n} |\varepsilon_t|}{n} = \frac{\sum_{t=1}^{n} |Y_t - \hat{Y}_t|}{n}$$

3) Mean Absolute Percent Error:

Mean Absolute Percent Error (MAPE - sai số tỷ lệ phần trăm tuyệt đối trung bình) là trung bình của các lỗi tuyệt đối chia cho các giá trị quan sát thực tế. Chúng ta có độ đo MAPE được tính theo công thức sau:

$$MAPE = \frac{\sum_{t=1}^{n} \frac{|\varepsilon_t|}{Y_t}}{n} = \frac{\sum_{t=1}^{n} \frac{|Y_t - \hat{Y}_t|}{Y_t}}{n}$$

4) R2 Score:

R2 là một số liệu rất quan trọng được sử dụng để đánh giá hiệu suất của mô hình học máy dựa trên hồi quy, hoạt động bằng cách đo lường số lượng phương sai trong các dự đoán được giải thích bởi bộ dữ liệu. Nói một cách đơn giản, đó là sự khác biệt giữa các mẫu trong bộ dữ liệu và các dự đoán được thực hiện bởi mô hình [14].

IV. CÁC THỬ NGHIỆM TINH CHỈNH MÔ HÌNH

GridSearchCV là quá trình thực hiện điều chỉnh hyperparameter (siêu tham số) để xác định các giá trị tối ưu cho một mô hình nhất định [15]. Hiệu suất của mô hình học máy phụ thuộc khá lớn đến các hyperparameter, tuy nhiên không có cách nào biết trước các giá trị tốt nhất, vì vậy phải thử các giá trị để tìm giá trị tối ưu. Làm điều này theo cách thủ công có thể mất một lượng thời gian và tài nguyên đáng kể và do đó chúng tôi sử dụng GridSearchCV để tự động điều chỉnh hyperparameters [15]. Bởi vì có giới hạn về tài nguyên và thời gian, nhóm đã không thử được thêm nhiều tham số hơn, các hyperparameter tối ưu nhất cho mô hình học máy mà nhóm tìm được là:

- Đối với mô hình Random Forest: 'max_features': 'auto', 'max_depth': None, 'min_sample_leaf':10, 'n_estimators': 400, 'min_samples_split':5

- Đối với mô hình Long short term memory: 'batch_size':1, 'epochs':1, 'hidden_layers':1, 'neurons':400

- Đối với mô hình XGBoost: 'colsample_bytree':1, 'learning_rate':0.2,'max_depth':7,'n_estimators':700, 'subsample': 0.6

Dưới đây là hình ảnh trực quan các giá trị dự đoán trên tập test và tập train dựa vào mô hình học máy mà nhóm thu được. Tên các đường chú thích trong ảnh tương ứng với giá trị dự đoán dựa trên tên mô hình tương ứng.

Các chỉ số Mean Squared Error(MSE), Mean Absolute Error(MAE), Mean Absolute Percent Error(MAPE), R2 Scrore thu được từ các dự đoán từ các mô hình học máy, các chỉ số được tính toán bằng hàm có sẵn trong thư viện sklearn, dưới đây là hình ảnh kết quả thu được từ file Source code của nhóm:

```
TẬP TEST                    TẬP TRAIN
Random Forest
MSE  : 4333682.8989  ||  64802.273654
MAE  : 1524.4629730  ||  111.00616910
MAPE : 3.4304558497  ||  2.1108255991
R2   : 0.9594346142  ||  0.9963654106

LSTM
MSE  : 4778076.4431  ||  197208.66237
MAE  : 1734.1195652  ||  344.61435457
MAPE : 3.9963009853  ||  44.382326052
R2   : 0.9552748739  ||  0.9889390840

XGBoost
MSE  : 6345492.7416  ||  74.117625530
MAE  : 1805.3842265  ||  6.4298873114
MAPE : 3.9868527692  ||  0.6379346409
R2   : 0.9406030928  ||  0.9999958429
```

Kết quả thu được chỉ là tương đối, với mỗi lần chạy lại mô hình sẽ đưa ra những chỉ số khác. Cả 4 chỉ số cho thấy khả năng dự đoán của mô hình LSTM trên tập train kém hơn so với mô hình Random Forest và XGBoost. Đối với mô hình Random Forest thì ngược lại, có biểu hiện tốt nhất trên tập train, nhưng lại biểu hiện kém nhất trên tập test. Cả 3 mô hình nhóm thu được đều thu được chỉ số R2 khá cao (cả 3 chỉ số R2 đều > 0.94 ở tập test, và đều lớn hơn 0.98 trên tập train). Chỉ số MAPE trên tập train của mô hình Long short-term memory cho chỉ số khá lớn (~44.382).

V. PHÂN TÍCH LỖI

Sau khi chia dữ liệu trên tập test 10 khoảng thời gian, 9 khoảng thời gian đầu mỗi khoảng thời gian gồm 53 ngày liên tiếp nhau, khoảng thời gian thứ 10 (cuối cùng) là 52 ngày liên tiếp nhau. Với mỗi khoảng dữ liệu sẽ in ra các chỉ số R2 để

tìm khoảng dữ liệu có các chỉ số kém, thuận tiện cho việc phân tích lỗi. Dưới đây là hình ảnh trực quan:

Nhóm xác định được khoảng thời gian có các chỉ số R2 thấp (mô hình hoạt động kém hiệu quả trên đoạn dữ liệu này) là từ ngày 19/03/2021 đến ngày 03/07/2021, từ ngày 17/10/2021 đến ngày 09/12/2021, từ ngày 31/01/2022 đến ngày 25/03/2022. Có thể thấy đây là đoạn dữ liệu có xu hướng giảm từ đỉnh ngay trước đó là 1 xu hướng tăng khá mạnh(gấp 3 lần giá trong vòng 3 tháng từ 20000$ lên 60000$). Chỉ số R2 cho thấy hiện tượng quá khớp của tập dữ liệu.

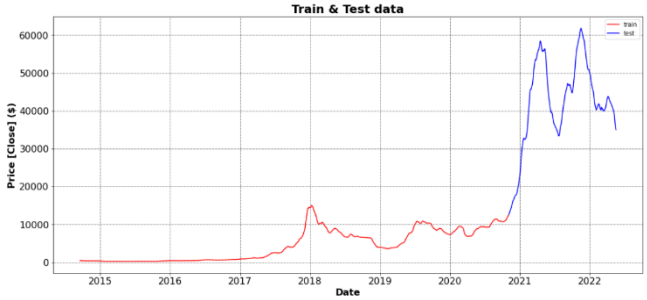
VI. HƯỚNG PHÁT TRIỂN

Nhóm sẽ xử lý dữ liệu bằng cách làm mịn. Mục tiêu của việc làm mịn là để giảm các yếu tố ngẫu nhiên và nhiễu khỏi dữ liệu, cũng như tạo biểu đồ mượt mà hơn, từ đó giúp mô hình dễ dàng xác định các xu hướng dài hạn hơn. Việc làm mịn được xác định bằng cách sau:

$$S_0 = Y_0$$

$$\text{for t} > 0,\ S_t = \alpha * Y_t + (1 - \alpha) * S_{t-1}$$

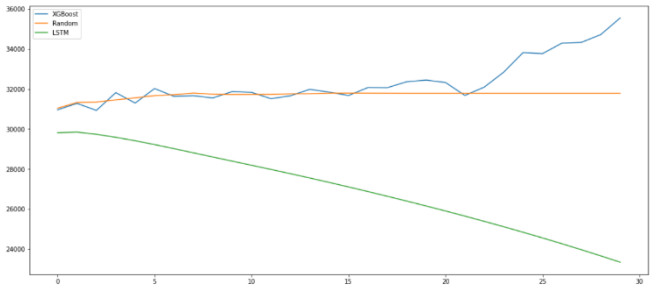
Trong đó, α là hệ số làm mịn và $0 < \alpha < 1$. Sau khi làm mịn, ta có được đồ thị từ dữ liệu như sau:



Tuy nhiên trong đồ án này, nhóm sẽ không sử dụng dữ liệu sau khi làm mịn để làm dữ liệu đầu vào cho mô hình học máy, nhóm kiểm tra lại dữ liệu và các điểm Outlier đã giảm. Nếu có thêm thời gian, nhóm sẽ tìm hiểu thêm các phương pháp tiền xử lý dữ liệu, phương pháp xử lý các điểm outlier. Phân tích dữ liệu time series, đánh giá và tính toán các chỉ số từ dữ liệu. Tìm hiểu thêm và chạy thêm các tham số đầu vào của mô hình để tìm các tham số giúp mô hình hoạt động tốt nhất. Tìm hiểu kĩ hơn về mô hình LSTM. Sử dụng thêm các chỉ báo kinh tế khác làm dữ liệu đầu vào cho mô hình học máy, từ đó giúp cải thiện mô hình dự đoán tốt hơn. Sử dụng các dữ liệu time series theo 1 giờ, 6 giờ, 12 giờ,3 ngày, 1 tuần,.... để dự đoán trong ngắn và dài hạn. Tạo một ứng dụng sử dụng cho việc dự đoán thêm các đồng tiền điện tử khác như ETH, BNB, SOL,.... Ngoài 3 phương pháp học máy mà nhóm đã sử dụng trong bài báo cáo này, nhóm sẽ tìm hiểu và chạy thử thêm các mô hình học máy khác.

VII. KẾT LUẬN

Trong bài báo cáo này, nhóm đã trình bày một số nội dung lý thuyết về dữ liệu nhóm sử dụng và ba mô hình học máy là Random Forest, XGBoost, Long short term memory cũng như lý thuyết và công thức tính toán các chỉ số Mean Absolute Error, Mean Squared Error và Mean Absolute Percent Error. Việc áp dụng GridSearchCV để tìm các hyperparameters đã cho thấy hiệu quả trong việc giảm độ lệch của giá trị dự đoán so với giá trị thật (xét trên tập train và test). Tuy nhiên, nhóm vẫn gặp khó khăn và chưa đi sâu được vào việc phân tích dữ liệu, cũng như xử lý các vấn đề liên quan đến các điểm dữ liệu. Nhóm đã trình bày và sử dụng phương pháp làm mịn dữ liệu tuy nhiên vẫn chưa áp dụng mô hình học máy. Có thể thấy mô hình xxx hoạt động tốt hơn mô hình xxx...Dưới đây là hình ảnh dự đoán giá đồng tiền điện tử Bitcoin của 3 mô hình học máy mà nhóm sử dụng mô hình học máy sau khi được tối ưu:



VIII. BẢNG PHÂN CÔNG CÔNG VIỆC

Công việc thực hiện	**Đức 20521196**	**Vy 20520951**	**Huy 20521418**
Lên ý tưởng và phân công	x		
Khám phá, chỉnh sửa dữ liệu		x	x
Mô hình Random Forest	x		
Mô hình XGBoost	x		x
Mô hình Long Short-term memory		x	x
Đánh giá mô hình	x	x	x
Phân tích lỗi	x		
Hoàn thiện source code	x	x	x
Bản báo cáo	x		
Slide		x	
Thuyết trình			x

TÀI LIỆU

[1] Influxdata, "What is time series data?", Influxdata, [Online], Available: What is Time Series Data? | Definition, Examples, Types & Uses (influxdata.com)

[2] Bigdatauni, "Tìm hiểu về Time series(phân tích chuỗi thời gian)(P1)", Bigdatauni, [Online], Available: Tìm hiểu về Time series (phân tích chuỗi thời gian) (P.1) - Big Data Uni

[3] Serafeim Loukas, "Everything you need to know about Min-Max normalization: A Python tutorial", Towards Data Science, [Online], Available: Everything you need to know

about Min-Max normalization: A Python tutorial | by Serafeim Loukas | Towards Data Science

[4] "Cây Quyết Định (Decision Tree)," Trí tuệ nhân tạo,[Online], Available: Cây Quyết Định (Decision Tree) - Trí tuệ nhân tạo (trituenhantao.io)

[5] Alex Reed, "Predicting Stock Movement with Random Forest", Github, [Online], Available: sigma_coding_youtube/random_forest_price_prediction.ipynb at master · areed1192/sigma_coding_youtube (github.com)

[6] Võ Mỹ Quyên, "XGBoost: Một cuộc lặn sâu để thúc đẩy", Helpex, [Online], Available: XGBoost: Một cuộc lặn sâu để thúc đẩy (helpex.vn)

[7] Jason Brownlee, "A Gentle Introduction to Long Short-Term Memory Networks by the Experts", Machinelearningmastery, [Online], Available: A Gentle Introduction to Long Short-Term Memory Networks by the Experts (machinelearningmastery.com)

[8] "SỬ DỤNG MẠNG LSTM (LONG SHORT TERM MEMORY) ĐỂ DỰ ĐOÁN SỐ LIỆU HƯỚNG THỜI GIAN", trituevietvn, [Online], Available: TTV-GIÁO DUC ỨNG DUNG (trituevietvn.com)

[9] "XGBoost: Hướng dẫn hoàn chỉnh để tinh chỉnh và tối ưu hóa mô hình của bạn", Ichi, [Online], Available: XGBoost: Hướng dẫn hoàn chỉnh để tinh chỉnh và tối ưu hóa mô hình của bạn (ichi.pro)

[10] "Epoch là gì", Sydneyowenson.com, [Online], Available: Tham Số Epoch Là Gì - What Is Batch Size In Neural Network (sydneyowenson.com)

[11] "Sai số dự báo (Forecast Error) là gì?", vietnambiz, [Online], Available: Sai số dự báo (Forecast Error) là gì? (vietnambiz.vn)

[12] "Bài 6: Convolutional neural network", Nttuan8, [Online], Available: Bài 6: Convolutional neural network | Deep Learning cơ bản (nttuan8.com)

[13] "MEAN SQUARE ERROR LÀ GÌ", diymcwwm, [Online], Available: " Mean Square Error Là Gì - Sai Số Toàn Phương Trung Bình (diymcwwm.com)

[14] "R2 Score in Machine Learning",Aman Kharwal, [Online],Available: R2 Score in Machine Learning (thecleverprogrammer.com)

[15] "Hyperparameter Tuning with GridSearchCV", Great Learning Team, [Online], Available: An Introduction to GridSearchCV | What is Grid Search | Great Learning (mygreatlearning.com)