

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THÔNG THÔNG TIN

NGUYỄN MINH NHỰT

KHÓA LUẬN TỐT NGHIỆP
**XÂY DỰNG HỆ THỐNG DỰ BÁO GIÁ VÀ KHUYẾN
NGHỊ GIAO DỊCH MUA BÁN TIỀN MÃ HÓA**
**BUILDING A SYSTEM FORECASTING RATE AND
RECOMMENDING TRANSACTION ABOUT CRYPTO
CURRENCY**

KỸ SƯ HỆ THÔNG THÔNG TIN

TP. HỒ CHÍ MINH, 2021

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THÔNG THÔNG TIN

NGUYỄN MINH NHỰT – 17520867

KHÓA LUẬN TỐT NGHIỆP
**XÂY DỰNG HỆ THỐNG DỰ BÁO GIÁ VÀ KHUYẾN
NGHỊ GIAO DỊCH MUA BÁN TIỀN MÃ HÓA**
**BUILDING A SYSTEM FORECASTING RATE AND
RECOMMENDING TRANSACTION ABOUT CRYPTO
CURRENCY**

KỸ SƯ HỆ THÔNG THÔNG TIN

GIẢNG VIÊN HƯỚNG DẪN
PGS. TS NGUYỄN ĐÌNH THUÂN

TP. HỒ CHÍ MINH, 2021

THÔNG TIN HỘI ĐỒNG CHẤM KHÓA LUẬN TỐT NGHIỆP

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

1. – Chủ tịch.
2. – Thư ký.
3. – Ủy viên.
4. – Ủy viên.

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

TRƯỜNG ĐẠI HỌC **Độc lập - Tự do - Hạnh phúc**

CÔNG NGHỆ THÔNG TIN

TP. HCM, ngày... tháng... năm 2021

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP
CỦA CÁN BỘ HƯỚNG DẪN

Tên khóa luận:

**XÂY DỰNG HỆ THỐNG DỰ BÁO GIÁ VÀ KHUYẾN NGHỊ GIAO DỊCH
MUA BÁN TIỀN MÃ HÓA**

Nhóm SV thực hiện:

Nguyễn Minh Nhựt - 17520867

Cán bộ hướng dẫn:

PGS. TS Nguyễn Đình Thuân

Đánh giá Khóa luận

1. Về cuốn báo cáo:

Số trang _____ Số chương _____

Số bảng số liệu _____ Số hình vẽ _____

Số tài liệu tham khảo _____ Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

.....
.....
.....
.....

2. Về nội dung nghiên cứu:

.....
.....
.....

3. Về chương trình ứng dụng:

.....
.....
.....

4. Về thái độ làm việc của sinh viên:

.....
.....
.....

Đánh giá chung:

.....
.....
.....

Điểm từng sinh viên:

Nguyễn Minh Nhựt: _____/10

Người nhận xét

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cảm ơn PGS. TS Nguyễn Đình Thuân. Trong quá trình làm khóa luận tốt nghiệp với thầy, em đã nhận được sự quan tâm huyêt của thầy. Thầy đã giúp em tích lũy thêm nhiều kiến thức về chuỗi thời gian, mô hình dự báo các thuật toán máy học và hồi quy.

Để hoàn thành khóa luận này một tốt nhất, em chân thành cảm ơn các thầy cô giáo Khoa Hệ thống Thông tin cung cấp cho em kiến thức chuyên nghành đã giúp em có kiến thức để góp phần vào sự hoàn thành của khóa luận.

Có lẽ kiến thức là vô hạn mà sự tiếp nhận kiến thức của mỗi người là có giới hạn. Vì vậy, trong quá trình thực hiện khóa luận cũng gặp một chút sai sót nhất định. Chúng em mong sự góp ý của Thầy, Cô giáo để khóa luận em được hoàn chỉnh hơn, song giúp em có nhiều kinh nghiệm hơn trong quá trình học tập và nghiên cứu.

TP. Hồ Chí Minh, ngày 02 tháng 7 năm 2021

Sinh viên thực hiện
Nguyễn Minh Nhựt

MỤC LỤC

Chương 1. GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI	2
1.1. Giới thiệu về dự báo	2
1.1.1. Định nghĩa	2
1.1.2. Các phương pháp dự báo.....	3
1.1.3. Tính chất, đặc điểm của dự báo.....	5
1.1.4. Các ứng dụng của dự báo	5
1.2. Giới thiệu về đối tượng dự báo.....	6
1.2.1. Công nghệ Blockchain	6
1.2.2. Đồng tiền Bitcoin	8
1.2.3. Đồng tiền Ethereum.....	8
1.3. Lý do chọn đề tài	9
1.4. Các công nghệ dự báo, khuyến nghị và môi trường phát triển	10
1.4.1. Các công nghệ dự báo	10
1.4.2. Công nghệ khuyến nghị.....	10
1.4.3. Môi trường phát triển	10
1.5. Kiến trúc đề tài	11
Chương 2. CƠ SỞ LÝ THUYẾT	12
2.1. Lý thuyết chuỗi thời gian.....	12
2.1.1. Định nghĩa chuỗi thời gian	12
2.1.2. Các thành phần của chuỗi thời gian	13
2.2. Mô hình ARIMA	15
2.2.1. Chuỗi dừng (Stationary Time series)	15
2.2.2. Kiểm định DF và Kiểm định ADF [7]	16

2.2.3.	Mô hình tự hồi quy (Autoregression – AR)	17
2.2.4.	Mô hình trung bình trượt (Moving Average – MA)	17
2.2.5.	Sai phân I.....	18
2.2.6.	Mô hình Box-Jenkins	18
2.2.7.	Chỉ số AIC.....	19
2.3.	Support Vector Machine và Module SVR.....	20
2.3.1.	Lý thuyết cơ sở SVM [8].....	20
2.3.2.	Xây dựng bài toán tối ưu sử dụng SVM.....	22
2.3.3.	Kernel SVM [9].....	25
2.3.4.	Module SVR.....	27
2.4.	Neural Network và mô hình LSTM [12], [13], [14] [15].....	28
2.4.1.	Cơ sở lý thuyết Neural Network.....	28
2.4.2.	Mạng RNN và Mạng LSTM	35
2.5.	Mạng tăng cường và Q-Learning	38
2.6.	Phương pháp đánh giá mô hình	43
2.6.1.	Root Mean Square Error (RMSE)	43
2.6.2.	Mean Absolute Percentage Error (MAPE).....	43
Chương 3.	KẾT HỢP MÔ HÌNH ARIMA VỚI HỒI QUY, NEURAL NETWORK	
	44	
3.1.	Lý thuyết kết hợp mô hình [20], [21], [22]	44
3.2.	Ứng dụng mô hình Hybrid ARIMA-SVR và ARIMA-LSTM.....	45
Chương 4.	PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM	46
4.1.	Pharse 1 – Học mô hình dự báo và khuyến nghị	46
4.1.1.	Nguồn dữ liệu, chia train test	46

4.1.2.	Kết quả dự báo từng mô hình dự báo	47
4.1.3.	Đánh giá mô hình mô hình dự báo phù hợp cho Pharse 2	50
4.1.4.	Mô hình khuyến nghị Q-Learning cho Pharse 2	51
4.2.	Pharse 2 – Quy trình xây dựng hệ thống API.....	52
4.2.1.	Thiết kế thành phần chức năng.....	52
4.2.2.	Thiết kế thành phần dữ liệu.....	56
4.2.3.	Thiết kế thành phần xử lý.....	58
4.3.	Pharse 3 – Triển khai hệ thống API lên Server Cloud Linux.....	59
4.3.1.	Cơ chế định thời trong Laravel [22].....	59
4.3.2.	Cơ chế CRON trong Server Linux	63
4.4.	Pharse 4 – Xây dựng giao diện và giao tiếp API.....	65
Chương 5.	KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN	68
5.1.	Kết quả đạt được.....	68
5.2.	Hướng phát triển.....	68
Chương 6.	TÀI LIỆU THAM KHẢO	70

DANH MỤC HÌNH

Hình 1.1 Mô hình chuyển tiền truyền thông qua ngân hàng.....	6
Hình 1.2 Blockchain visualization	7
Hình 1.3 Mô hình chuyển tiền sử dụng Blockchain	7
Hình 1.4 Bảng so sánh đồng Bitcoin và đồng Ethereum [3]	9
Hình 1.5 Kiến trúc hệ thống	11
Hình 2.1 Biểu đồ thể hiện giá Bitcoin theo thời gian.....	12
Hình 2.2 Thành phần xu thế (T).....	13
Hình 2.3 Thành phần mùa vụ (S) [4]	14
Hình 2.4 Thành phần chu kỳ (C) [5]	14
Hình 2.5 Các dạng chuỗi dừng và chuỗi không dừng [6]	15
Hình 2.6 Mô hình Box-Jinkins áp dụng cho mô hình ARIMA	19
Hình 2.7 Mô tả hai lớp trên không gian được phân chia bởi nhiều siêu phẳng	21
Hình 2.8 Mặt phân chia với điểm gần nhất của mỗi class	21
Hình 2.9 Margin của các class	22
Hình 2.10 Phân tích bài toán SVM trên không gian 2 chiều	23
Hình 2.11 Các điểm gần mặt phân cách được biểu diễn khoanh tròn	24
Hình 2.12 Dữ liệu 2 chiều không có Linearly Separable.....	25
Hình 2.13 Dữ liệu 3 chiều đã có Linearly Separable	26
Hình 2.14 Dữ liệu đã được phân lớp	26
Hình 2.15 Một số Kernel thông dụng	27
Hình 2.16 Mô hình hồi quy SVR	27
Hình 2.17 cấu tạo một mạng Neural Network [10].....	28
Hình 2.18 Mô hình Mạng Neural theo Supervised Learning.....	30
Hình 2.19 Mô hình Mạng Neural theo Upsupervised Learning	30
Hình 2.20 Mô hình Mạng Neural theo kiểu học tăng cường	30
Hình 2.21 Hàm chi phí trong không gian 3 chiều	32
Hình 2.22 Recurrent Neural Network có vòng lặp [16].....	35
Hình 2.23 Recurrent Neural Network đơn giản hóa	36

Hình 2.24 Một khối LSTM Block cơ bản	36
Hình 2.25 Các ký hiệu của LSTM block	36
Hình 2.26 Forget Gate.....	37
Hình 2.27 Input Gate.....	37
Hình 2.28 Merge Cell Forget Gate and Input Gate.....	38
Hình 2.29 Output Gate	38
Hình 2.30 Quá trình học Q-Learning	39
Hình 3.1 Mô hình lai giữa Linear Model và Non-Linear Model	45
Hình 4.1 Trang Web dùng để download dữ liệu Bitcoin	46
Hình 4.2 Trang Web dùng để download dữ liệu Ethereum	46
Hình 4.3 Đồ thị về Closing Price theo thời gian của đồng BTC và ETH	47
Hình 4.4 Usecase tổng quan hệ thống dự báo và khuyến nghị BTC và ETH.....	52
Hình 4.5 Class diagram hệ thống	56
Hình 4.6 Sequence Diagram Dự báo giá BTC và ETH	58
Hình 4.7 Sequence Diagram khuyến nghị BTC và ETH	59
Hình 4.8 Ví dụ minh họa về Task Scheduling	59
Hình 4.9 Schedule lấy dữ liệu mới.....	63
Hình 4.10 Schedule lấy dữ liệu mới nhất xuất thành CSV	63
Hình 4.11 File cấu hình crontab	64
Hình 4.12 Danh sách các crontab đang chạy trong server Linux	64
Hình 4.13 Màn hình tổng quan Bitcoin Page.....	65
Hình 4.14 Màn hình tổng quan Ethereum.....	65
Hình 4.15 Màn hình xem lịch sử giá dạng bảng	66
Hình 4.16 Màn hình xem lịch sử giá dạng biểu đồ	66
Hình 4.17 Màn hình khuyến nghị hành động.....	67

DANH MỤC BẢNG

Bảng 2.1 Một số hàm kích hoạt thông dụng	31
Bảng 2.2 Dữ liệu nhập Input Bitocin	33
Bảng 2.3 Dữ liệu đầu vào và đầu ra mẫu	34
Bảng 2.4 Ví dụ về thuật toán Q-Learning	40
Bảng 2.5 Bảng Q-Table ban đầu	40
Bảng 2.6 Bảng Q-Learning chạy lần thứ 1	41
Bảng 2.7 Vị trí của chú chuột tại lần thứ 1	42
Bảng 2.8 Bảng Q-Learning chạy lần thứ 2	42
Bảng 4.1 Kết quả dự báo mô hình ARIMA trên hai loại đồng tiền	47
Bảng 4.2 Kết quả dự báo mô hình SVR trên hai loại đồng tiền.....	48
Bảng 4.3 Kết quả dự báo mô hình LSTM trên hai loại đồng tiền.....	48
Bảng 4.4 Kết quả dự báo mô hình ARIMA_SVR trên hai loại đồng tiền	49
Bảng 4.5 Kết quả dự báo mô hình ARIMA_LSTM trên hai loại đồng tiền	49
Bảng 4.6 Error of individual models with BTC Closing Price	49
Bảng 4.7 Error of individual models with ETH Closing Price	50
Bảng 4.8 Error of hybrid models with BTC Closing Price	50
Bảng 4.9 Error of hybrid models with ETH Closing Price	50
Bảng 4.10 Usecase xem dự báo	53
Bảng 4.11 Usecase xem khuyến nghị	54
Bảng 4.12 Usecase lấy dữ liệu mới.....	54
Bảng 4.13 Usecase xuất file CSV	55
Bảng 4.14 Usecase Dự báo giá	55
Bảng 4.15 Usecase Xuất đồ thị khuyến nghị	56
Bảng 4.16 Bảng thuộc tính Bitcoin	57
Bảng 4.17 Bảng hàm Bitcoin Class	57
Bảng 4.18 Bảng thuộc tính Ethereum	57
Bảng 4.19 Bảng hàm Ethereum Class.....	58
Bảng 4.20 Danh sách các hàm schedule được dựng sẵn Laravel	62

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Dịch sang tiếng anh	Ý nghĩa Tiếng Việt
BTC	Bitcoin	Đồng bitcoin
ETH	Ethereum	Đồng Ethereum
PoW	Proof of Work	
PoS	Proof of Stake	
CSDL		Cơ sở dữ liệu
T	Trend	Xu thế
S	Season	Mùa vụ
DF	Dickey Fuller	Kiểm định Dickey Fuller
ADF	Augmented Dickey Fuller	Kiểm định Augmented Dickey Fuller
SVM	Support Vector Machine	
SVR	Support Vector Regression	
RNN	Recurrent Neural Network	
RMSE	Root Mean Square Error	
MAPE	Mean Absolute Percentage Error	

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

TÓM TẮT KHÓA LUẬN

Hiện nay tiền mã hóa (Crypto Currency) đang tăng mạnh trong quý I năm 2021.

Dự báo giá tăng, giảm hoặc khuyến nghị chọn trạng thái mua, bán hoặc giữ là điều mà người chơi loại tiền này quan tâm. Việc dự báo đòi hỏi có độ chính xác cao bằng cách kết hợp các phương pháp thống kê, thuật toán máy học để đưa ra kết luận đúng đắn.

Đề tài nghiên cứu về các mô hình dự báo chuỗi thời gian: Mô hình ARIMA, thuật giải SVM, hay các phương pháp máy học: Hồi quy, học tăng cường Q-Learning, mạng Neural, LSTM,...

Nội dung khóa luận gồm 6 chương chính

- Chương 1: GIỚI THIỆU MÔ TẢ ĐỀ TÀI
- Chương 2: CƠ SỞ LÝ THUYẾT
- Chương 3: KẾT HỢP MÔ HÌNH ARIMA VỚI HỒI QUY VÀ NEURAL NETWORK
- Chương 4: PHÂN TÍCH THỰC NGHIỆM
- Chương 5: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN
- Chương 6: TÀI LIỆU THAM KHẢO

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

Chương 1. GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

1.1. Giới thiệu về dự báo

1.1.1. Định nghĩa

Thuật ngữ *dự báo* [1] có nguồn gốc từ khoa từ Hy Lạp cổ đại.

Dự báo là quá trình tiên đoán các quan hệ, xu hướng phát triển của đối tượng nghiên cứu trong tương lai bằng những dẫn chứng, biện chứng dựa trên căn cứ khoa học.

Tiên đoán là hình thức phản ánh vượt trước thời gian về mặt hiện thực, đưa ra kết quả nhận thức chủ quan của con người. Có ba hình thức tiên đoán:

- *Hình thức tiên đoán phản khoa học*: Dự đoán trên cơ sở không khoa học, thường mang tính chất tưởng tượng. Các hình thức: Nằm mộng, bói toán, tiên trí thuộc thể loại hình thức tiên đoán này.
- *Hình thức tiên đoán kinh nghiệm*: Các tiên đoán hình thành qua kinh nghiệm thực tế, trải nghiệm, nghiên cứu các quy luật thông qua kinh nghiệm. Loại tiên đoán này ít nhiều cũng có cơ sở nhưng nó không sử dụng dựa trên khoa học chứng minh.
- *Hình thức tiên đoán khoa học*: Dựa trên cơ sở khoa học, những con số, công thức, lý luận khoa học nhất định. Tiên đoán khoa học là kết quả của sự kết hợp giữa những phân tích định tính và những phân tích định lượng các quá trình cần dự báo. Hình thức tiên đoán này có độ tin cậy cao và mức rủi ro thấp.

Dự báo là quá một quá trình đưa ra một số liệu trong tương về kinh tế, xã hội,... Nhằm mục tiêu phục vụ cho việc đưa ra quyết định của doanh nghiệp hoặc hoạch định chính sách của chính phủ.

Mô hình dự báo = Dữ liệu + Kỹ thuật dự báo

Để một mô hình dự báo tốt cần phải có dữ liệu tốt và kỹ thuật dự báo tốt.

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

1.1.2. Các phương pháp dự báo

Có rất nhiều phương pháp dự báo hiện nay [2]: Các phương pháp điều tra, phương pháp thực nghiệm, phương pháp ngoại suy, phân tích kinh tế lượng, phân tích chuyên gia, phương pháp phân tích đầu vào – đầu ra, phân tích I-O hay phân tích liên ngành,

1.1.2.1. Phương pháp thực nghiệm

Loại dự báo này là đưa ra kết quả dự báo cho sản phẩm mới, bằng cách cho sử dụng sản phẩm thử nghiệm và cải tiến trên sự quan sát phản ứng của khách hàng với sản phẩm thử trong một thị trường nhỏ để hoặc các thị trường lớn bán thử nhằm mục đích xác định nhu cầu của khách hàng đối với sản phẩm.

Phương pháp này bằng cách tập hợp ý kiến khách hàng tại nhiều khu vực khác nhau để có thể dự báo nhu cầu sản phẩm, dịch vụ của doanh nghiệp mang lại.

Phương pháp này có nhược điểm là phụ thuộc đánh giá chủ quan của người bán hàng.

1.1.2.2. Phương pháp ngoại suy

Phương pháp ngoại suy là phương pháp sử dụng dữ liệu quá khứ theo dạng chuỗi thời gian để phân tích, xác định xu thế phát triển của chúng để dự báo cho tương lai.

Phương pháp ngoại suy ngầm giả định quy định dữ liệu lịch sử có quan hệ với nhau theo một quy luật từ đó mà diễn giải giá trị tương lai dựa trên quy luật đó.

Nhược điểm chính của phương pháp này là nếu thị trường biến động mạnh thì quy luật chuỗi thời gian để diễn giải giá trị tương lai có thể không tốt, không đáng tin cậy. Tuy vậy phương pháp này vẫn được sử dụng rộng rãi nhất trong các phương pháp dự báo.

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

Một số phương pháp đặc trưng cho phương pháp này là: bình quân trượt, san só mũ.

1.1.2.3. Phương pháp kinh tế lượng

Phương pháp kinh tế lượng là phương pháp dựa trên kiến thức thống kê và phân tích hồi quy để đưa ra một công thức nhất định có dạng $Y = aX + b$ và phải phù hợp với xu thế và đặc điểm của đối tượng nghiên cứu. Phương pháp dự báo này dùng để dự báo cho ngắn hạn dài hạn.

Giống như phương pháp ngoại suy, *phương pháp kinh tế lượng* được xem là phương pháp được sử dụng rộng rãi bởi các chuyên gia dự báo. Tuy nhiên phương pháp này cũng không tốt đối với thị trường biến động lớn.

Một số phương pháp đặc trưng cho phương pháp này là: Hồi quy đơn biến, hồi quy đa biến, hồi quy phi tuyến, hồi quy logistics,...

1.1.2.4. Phân tích đầu vào-đầu ra

Phương pháp phân tích đầu vào-đầu ra là phương pháp kinh tế vĩ mô dựa trên sự phụ thuộc lẫn nhau giữa các ngành kinh tế. Phương pháp này thường được ước tính tác động của các cú sốc kinh tế tiêu cực hoặc tiêu cực làm lan rộng nền kinh tế.

Phương pháp phân tích kinh tế này ban đầu được phát triển bởi Wassily Leontief¹ đã dành được giải nobel về phương pháp phân tích đầu vào đầu ra.

Nền tảng chính của phương thức đầu vào-đầu ra gồm 2 bảng chính là bảng đầu vào và bảng đầu ra. Dữ liệu trong mỗi cột tương ứng với mức đầu vào được sử dụng trong hàm sản xuất của ngành đó.

¹ Wassily Leontief (1905 -1999) là một nhà kinh tế Nga-Mỹ là người đạt giải Nobel cho lĩnh vực kinh tế về vấn đề phương pháp phân tích đầu vào-đầu ra

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

Ví dụ: Đề sản xuất một chiếc ô (đầu ra) cần các đầu vào là sắt, thép, cao su,... Nếu dự báo ô tô là bao nhiêu chiếc sẽ được bán ra thì có thể dự báo các sản phẩm liên quan.

1.1.3. Tính chất, đặc điểm của dự báo

Tính chất của dự báo là đưa ra xu thế để cạnh tranh, phát sinh lợi nhuận trên cơ sở khoa học.

Các phương pháp dự báo thường rất hiếm đạt mức độ hoàn hảo, vì vậy cần phải có sai số cho phép.

Dự báo cho nhóm đối tượng thường chính xác hơn dự báo từng đối tượng riêng lẻ.

Độ chính xác của dự báo giảm dần khi kéo dài thời gian dự báo.

1.1.4. Các ứng dụng của dự báo

Dự báo thời tiết: Một ứng dụng phổ biến nhất trong các ứng dụng dự báo, lấy các đặc điểm thời tiết của chuỗi thời gian quá khứ để dự báo cho những ngày tiếp theo.

Dự báo giá vàng, cổ phiếu: Lấy những đặc điểm xu thế về giá của thời điểm quá khứ để dự báo cho tương lai.

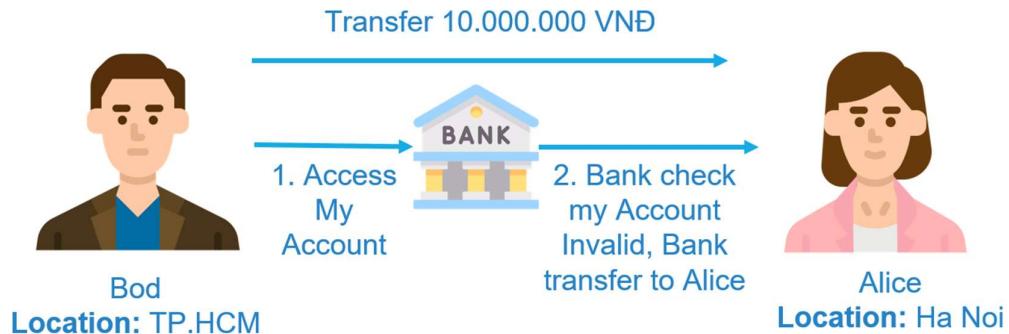
AirVisual Air Quality Forecast: Ứng dụng theo dõi dự báo về chất lượng không khí.

Ngoài ra dự báo còn ứng dụng trên rất nhiều lĩnh vực: Kinh tế, chính trị, văn hóa, xã hội, du lịch, giáo dục,...

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

1.2. Giới thiệu về đối tượng dự báo

1.2.1. Công nghệ Blockchain



Hình 1.1 Mô hình chuyển tiền truyền thống qua ngân hàng

Đầu tiên ta xét mô hình chuyển tiền truyền thống Bod ở vị trí TP.HCM chuyển tiền 10.000.000 VNĐ cho Alice tại địa điểm Hà Nội qua các bước sau đây:

- **Bước 1:** Bod truy cập tài khoản của mình thông qua tài khoản ngân hàng của mình.
- **Bước 2:** Ngân hàng kiểm tra số dư của Bod đủ đáp ứng việc gửi cho Alice hay không? Nếu đủ thì giao dịch thành công thì tiền sẽ gửi đến tài khoản của Alice.
- **Bước 3:** Bod gọi điện thoại/nhắn tin cho Alice để biết tiền của mình đã chuyển thành công.

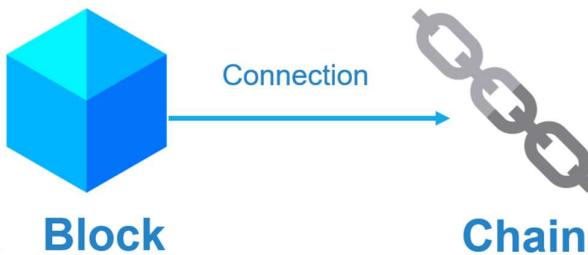
Vấn đề của mô hình truyền thống:

Cả Bod và Alice đều tin tưởng một hệ thống thứ 3 để quản lý tiền của họ, việc chuyển tiền của Bod và Alice chỉ lưu vào một cuốn sổ cái lưu trữ giao dịch của ngân hàng, hay nói cách khác chỉ mục của cuốn sổ đó Alice và Bod không được quản lý.

Phụ thuộc quá nhiều vào bên thứ 3, giả sử bên thứ 3 bị hack hoặc bị quá tải thì giao dịch của Bod và Alice liệu có an toàn không?

Để thay thế mô hình chuyển tiền truyền thống, sự ra đời của công nghệ **Blockchain** là một phương thức để duy trì cuốn sổ cái giữa tất cả các người dùng, thay vì phải phụ thuộc vào bên thứ 3.

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI



Hình 1.2 Blockchain visualization

- **Block** là một khối hình vuông, hoặc hình chữ nhật.
- **Chain** là một chuỗi.
- **Blockchain** là một chuỗi hình vuông, hoặc một chuỗi hình chữ nhật được nối với nhau thành một hệ thống chuỗi lớn.
- Công nghệ này được phát minh bởi ông [Satoshi Nakamoto](#)²
- Được phát minh vào năm 2008
- Công nghệ **Blockchain** được đóng vai trò như một cuốn sổ lớn ghi lại tất cả các hoạt động và lưu trữ thông tin giao dịch



Hình 1.3 Mô hình chuyển tiền sử dụng Blockchain

Một tính năng ưu việt của công nghệ Blockchain là không đòi hỏi một hệ thống trung gian để xác nhận thông tin, hệ thống Blockchain hoạt động độc lập một thành phần bị hỏng, bị rò rỉ không ảnh hưởng đến các thành phần khác trong hệ thống Blockchain qua cơ chế đồng thuận Proof of Work (PoW) và cơ chế cỗ phần

² Satoshi Nakamoto: Người phát minh ra công nghệ Blockchain và ứng dụng công nghệ Blockchain vào đồng tiền Bitcoin

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

Proof of Stake (PoS). Điểm mạnh khác của công nghệ Blockchain là hệ thống mã hóa đặc biệt, đảm bảo các tính an toàn bảo mật: Tính bảo mật, tính toàn vẹn, tính xác thực, tính phân quyền, tính không thể chối từ.

Hiện tại công nghệ Blockchain có 3 phiên bản chính

- Blockchain 1.0 – Blockchain dành cho tiền tệ (Bitcoin)
- Blockchain 2.0 – Smart Contract – Ethereum (ETH)
- Blockchain 3.0 – Dapps

1.2.2. Đồng tiền Bitcoin

Bitcoin là loại Blockchain 1.0 dành cho tiền tệ:

- Bitcoin blockchain là nguồn gốc cho các loại tiền mã hóa khác ra đời.
Nó được thiết kế cho việc chuyển đổi ngang hàng các giá trị và nó đang hoạt động hiệu quả.
- Là một hệ thống giao dịch: an toàn, bảo mật, ẩn danh

1.2.3. Đồng tiền Ethereum

Đồng Ethereum là Blockchain 2.0 Smart Contract và đồng Ethereum

Vào khoảng năm 2013, một khung làm việc để thực thi mã là Ethereum đã được giới thiệu bởi các sáng lập viên. Trung tâm và sự đột phá của Ethereum blockchain chính là hợp đồng thông minh (smart contract).

So sánh về đồng Bitcoin và đồng Ethereum

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI



	bitcoin	ethereum
concept	digital money	smart contracts
transaction	send from alice to bob	send from alice to bob if.. • date = jan 1, 2018 • bob's balance < 10 eth
market cap	~\$18 billion	~\$1 billion
founder	satoshi nakamoto (unknown)	vitalik buterin and team
release date	jan 2009	july 2015
release method	early mining	presale raised \$18M in bitcoin

Hình 1.4 Bảng so sánh đồng Bitcoin và đồng Ethereum [3]

1.3. Lý do chọn đề tài

Trong những năm gần đây, Sự phát triển của Internet dẫn đến công nghệ Blockchain phát triển theo điều này làm cho Cryptocurrencies hay gọi là tiền mã hóa phát triển mạnh mẽ, vào tháng 6 năm 2021, giá 1 đồng BTC tương đương 34,040.92 đô la, và 1 đồng ETH tương đương 2,097.96 đô la. Việc một đồng BTC tương đương một chiếc xe hơi, hay một căn nhà làm cho nhiều cá nhân, doanh nghiệp sử dụng và giao dịch loại tiền làm cho nó trở nên phổ biến trong những năm gần đây trên mạng xã hội. Khi tìm kiếm trên trang tìm kiếm google.com từ khóa “bitcoin” đạt 1.030.000.000 kết quả tìm kiếm. Tại Việt Nam một đồng Bitcoin tương đương gần 795 triệu đồng. Do đó, việc sử dụng và giao dịch tiền ảo một là một vấn đề rất quan trọng. Vì vậy, việc phân tích giá xu hướng của 2 loại đồng tiền mã hóa hot nhất hiện nay là đồng BTC và ETH để dự báo về giá của loại tiền này cần được nghiên cứu song việc khuyến nghị hành động giao dịch trên 2 loại tiền này cũng phải được tính toán và đưa ra khuyến nghị một cách cẩn thận. Nên vấn đề nghiên cứu và thử nghiệm dự báo và khuyến nghị là một vấn đề tất yếu.

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

1.4. Các công nghệ dự báo, khuyến nghị và môi trường phát triển

1.4.1. Các công nghệ dự báo

Trong phạm vi đề tài thử nghiệm trên các công nghệ dự báo như sau:

- Mô hình kinh tế lượng: ARIMA
- Mô hình máy học: LSTM
- Mô hình hồi quy: SVR
- Mô hình kết hợp: ARIMA-LSTM, ARIMA-SVR

Đánh giá mô hình dự báo: MAPE, RMSE

1.4.2. Công nghệ khuyến nghị

Hoạt động khuyến nghị dựa trên mô hình máy học cụ thể hơn là học tăng cường (Reinforcement Learning-Thuật toán Q-Learning)

Hoạt động khuyến nghị sẽ tìm ra 3 hành động: Sell/Buy/Hold

1.4.3. Môi trường phát triển

Môi trường phát triển hệ thống được chia làm 3 phần:

- Phần I: Xây dựng mô hình dự báo bằng ngôn ngữ lập trình Python 3.6
- Phần II: Xây dựng hệ thống API bằng ngôn ngữ lập trình PHP 7.4 và framework Laravel
- Phần III: Xây dựng giao diện ứng dụng bằng ngôn ngữ HTML và Javascript Framework (VueJS)

Sever Cloud: Hệ điều hành Linux

Hệ quản trị Cơ sở dữ liệu: MySQL

Công cụ thiết kế: Visio, Photoshop

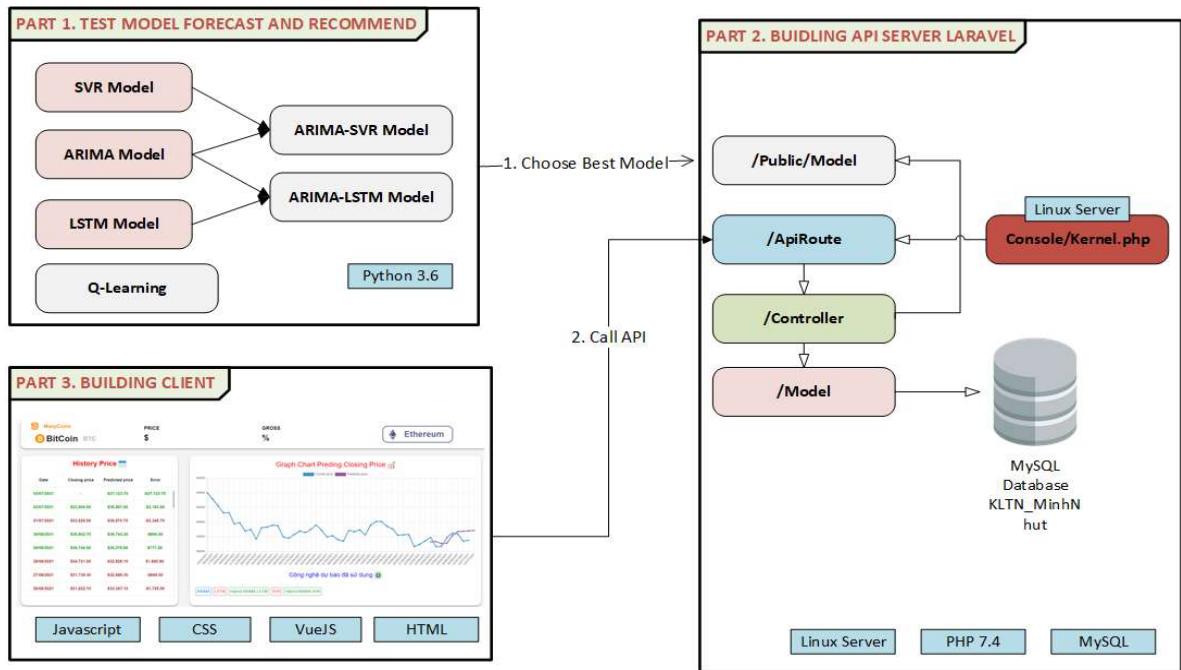
Công cụ quản lý Source Code: Smart Git, Github

Tên miền: <http://minhnhat.xyz>

Kỹ thuật định thời: CRON Schedule của hệ điều hành Linux

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

1.5. Kiến trúc đề tài



Hình 1.5 Kiến trúc hệ thống

Mô tả kiến trúc đề tài gồm 4 pharse cơ bản như sau:

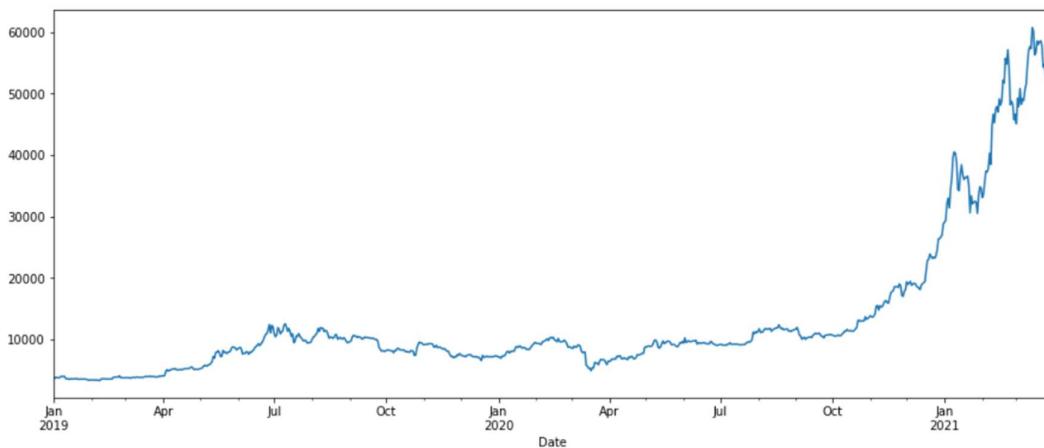
- *Triển khai Pharse 1*: Học các mô hình dự báo SVR, ARIMA, LSTM, Hybrid Model sau đó chọn mô hình tốt nhất dự báo tốt nhất để **dự báo**, riêng mô hình Q-Learning sẽ được chọn và đưa vào dùng để **khuyến nghị**. Tất cả mô hình máy học được đưa vào PART 2 thư mục public.
- *Triển khai Pharse 2*: Xây dựng các API clone dữ liệu từ Website Bitcoin, Ethereum về, Xây dựng các Controller, kiến trúc CSDL.
- *Triển khai Pharse 3*: Giai đoạn đưa API lên Cloud Server Linux, dùng cơ chế CRON Schedule của hệ điều hành Linux cho các hoạt động định thời chạy các mô hình máy học hoặc lấy dữ liệu từ Web site Data Bitcoin/Ethereum.
- *Triển khai Pharse 4*: Xây dựng Website Client để người dùng có thể thấy giá trị dự báo, khuyến nghị mua bán theo dạng bảng hoặc dạng đồ thị.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Lý thuyết chuỗi thời gian

2.1.1. Định nghĩa chuỗi thời gian

Time Series (Chuỗi thời gian) được xem là bao gồm các phép đo một giá trị theo thời gian ví dụ như giá BTC từ ngày 6/1/2019 – 6/1/2021 trong một khoảng thời gian. Phân tích chuỗi thời gian có mục đích nhận dạng và tập hợp lại các yếu tố, ảnh hưởng mà thời gian có ảnh hưởng đến giá trị mà ta quan sát.



Hình 2.1 Biểu đồ thể hiện giá Bitcoin theo thời gian

Chuỗi thời gian gian là một chuỗi mà trực tung y biểu diễn giá trị, trực hoành x biểu diễn thời gian. Và dữ liệu phải mang tính *liên tục, không ngắt quãng*.

Một số ví dụ của chuỗi thời gian:

- Chuỗi thời gian giá vàng, giá chứng khoán, giá bitcoin.
- Chuỗi thời gian nhiệt độ trung bình theo từng ngày.
- Chuỗi thời gian điện tâm đồ.
- Chuỗi thời gian chất lượng không khí.
- Chuỗi thời gian số lượng người bệnh Covid-19 từng ngày.
- Chuỗi thời gian về đánh giá chất lượng giấc ngủ mỗi người theo ngày.

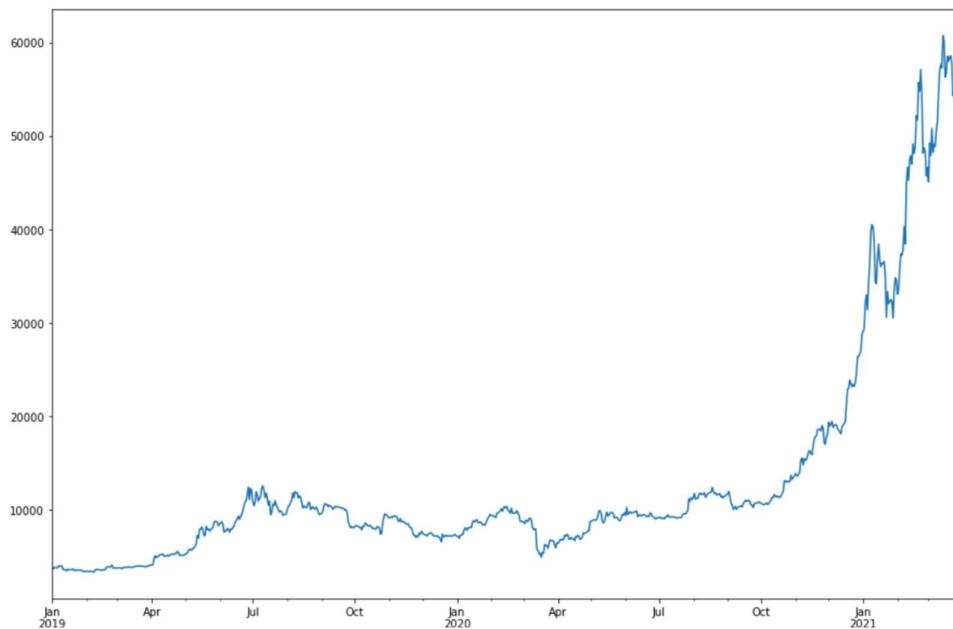
CƠ SỞ LÝ THUYẾT

2.1.2. Các thành phần của chuỗi thời gian

Một chuỗi thời gian có thể có một thành phần trong các thành phần hoặc có thể có nhiều thành phần trong các thành phần sau

2.1.2.1. Thành phần xu thế (T)

Thành phần xu thế là thành phần để chỉ ra xu thế tăng hoặc giảm của giá trị Y theo thời gian X. Đường xu thế có thể là một đường cong hoặc đường thẳng. Thành phần xu thế là thành phần thường được xuất hiện nhất trong các thành phần.

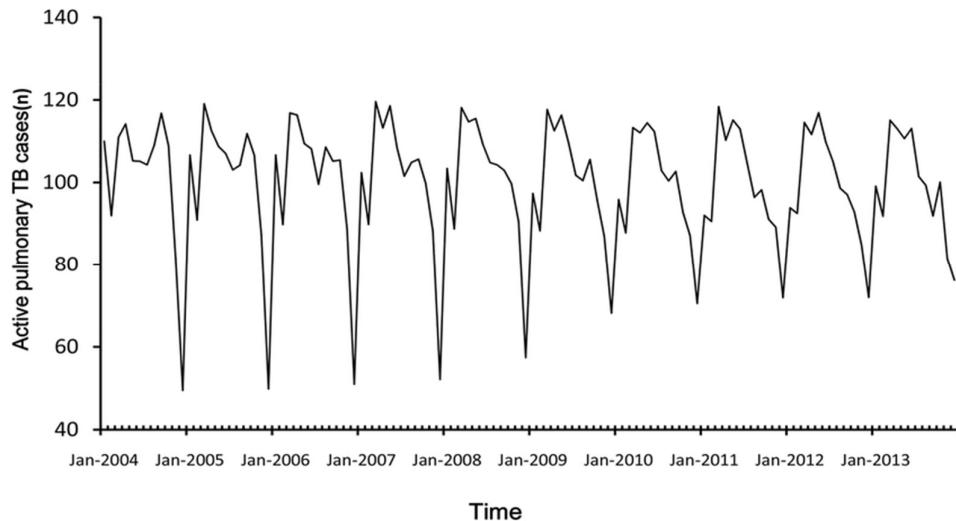


Hình 2.2 Thành phần xu thế (T)

2.1.2.2. Thành phần mùa vụ (S)

Thành phần mùa vụ là sau khoảng thời gian T là tập hợp nhiều giá trị thời gian X liên tiếp nhau thì giá trị Y lại quay lại vị trí X đầu tiên. Và quá trình lặp này có thể lặp đi lặp lại nhiều lần nhưng trong khoảng thời gian ngắn.

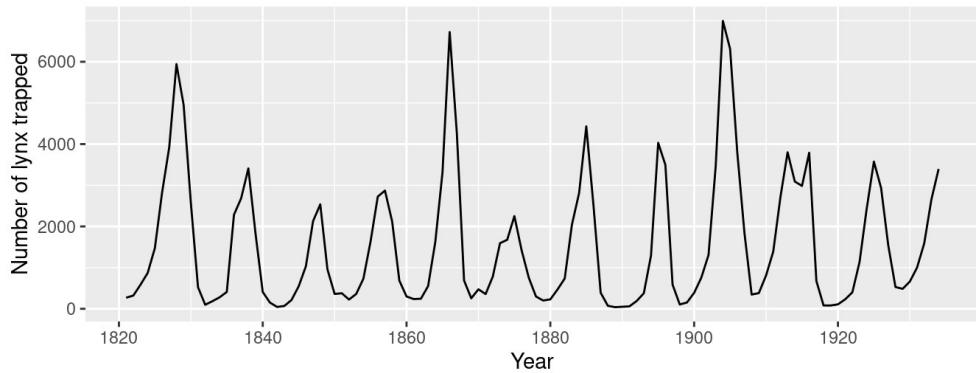
CƠ SỞ LÝ THUYẾT



Hình 2.3 Thành phần mùa vụ (S) [4]

2.1.2.3. Thành phần chu kỳ (C)

Thành phần chu kỳ là đặc điểm cũng giống như là thành phần mùa vụ nhưng khoảng chu kỳ dài hơn mùa vụ.



Hình 2.4 Thành phần chu kỳ (C) [5]

2.1.2.4. Thành phần bất thường (I)

Thành phần bất thường là những điểm mà nó không mang tính chất trên, tăng giảm không theo một quy luật, hệ quả nào cả. Thành phần này không thể dự đoán được bằng kinh nghiệm quá khứ. Những thị trường mà có thành phần bất thường I nhiều thì gọi là thị trường bất động mạnh.

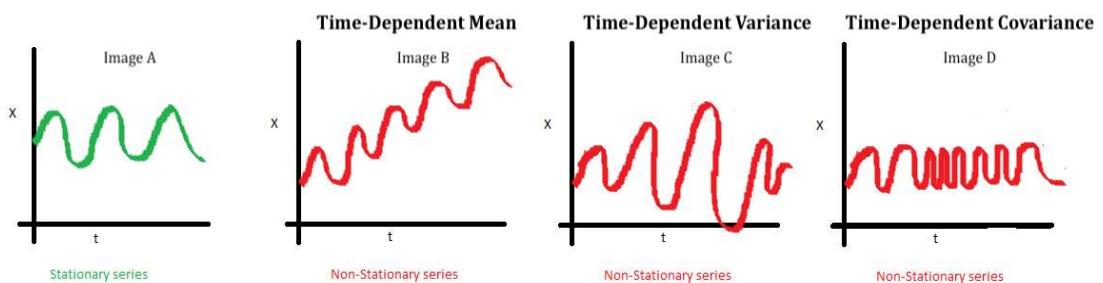
CƠ SỞ LÝ THUYẾT

2.2. Mô hình ARIMA

2.2.1. Chuỗi dừng (Stationary Time series)

Một chuỗi thời gian có tính dừng là một chuỗi các giá trị mean, variance, autocorrelation không thay đổi theo thời gian và nó không bao hàm yếu tố xu thế. Với hầu hết các phương pháp thống kê dự báo, ta đều phải đảm bảo tính dừng của chuỗi dữ liệu vì thế việc kiểm tra tính dừng là rất quan trọng.

The Principles of Stationarity



Hình 2.5 Các dạng chuỗi dừng và chuỗi không dừng [6]

Một số phương pháp kiểm tra chuỗi thời gian là chuỗi dừng hay không:

- Phương pháp nhìn vào đồ thị của chuỗi thời gian để phán đoán. Phương pháp này mang tính chất cảm tính không có cơ sở khoa học.
- Kiểm định Dickey Fuller³ (DF) và Dickey Fuller cải tiến (ADF⁴)

Phương pháp biến đổi chuỗi thời gian dừng về chuỗi dừng:

- Biến đổi thời gian bằng hàm logarit vì hàm logarit có thể mang lại chuỗi dừng. Việc lấy logarit làm giảm đi bậc liên kết.
- Mô hình ARIMA có thành phần I khác 0 thì mô hình ARIMA nó có biến đổi sai phân. Quá trình biến đổi sai phân là quá trình đưa một chuỗi không dừng thành một chuỗi thời gian dừng.

Nếu đã đưa I khác 0 mà mô hình ARIMA vẫn chưa về chuỗi thì quá trình này phải tiếp tục với $I = I_{\text{cũ}} + 1$ để đạt chuỗi dừng.

³ https://en.wikipedia.org/wiki/Dickey%20Fuller_test

⁴ https://en.wikipedia.org/wiki/Augmented_Dickey%20Fuller_test

CƠ SỞ LÝ THUYẾT

2.2.2. Kiểm định DF và Kiểm định ADF [7]

Kiểm định Dickey Fuller

Giả sử từ chuỗi thời gian ta có phương trình hồi quy:

$$y_t = p_0 + p_1 y_{t-1} + \varepsilon \quad (1)$$

Với các giá trị như sau

- y_t : Là giá trị tại thời điểm t
- p_0 : Là bậc tự do của phương trình hồi quy
- p_1 : Tham số thứ 1
- y_{t-1} : Là giá trị tại thời điểm t-1
- ε : Giá trị lỗi sai số

Từ công thức (1) trên ta trùa hai vế cho y_{t-1}

$$\Delta y_t = p_0 + \delta y_{t-1} + \varepsilon \quad (2)$$

Giả thiết $H_0: \delta = 0$ (Δy_t là chuỗi không dừng)

Đối thiêt $H_1: \delta < 0$ (Δy_t là chuỗi dừng)

Từ công thức ta có thể tính được tìm được giá test-value của phương trình hồi quy

$$t_{\hat{\delta}} = \frac{\hat{\delta}}{SE(\hat{\delta})}$$

Nếu test-value $< DF_{critical}$ từ chối giả thiêt $H_0 \rightarrow$ Tức là chuỗi thời gian dừng

Nếu test-value $> DF_{critical}$ chấp nhận giả thiêt $H_0 \rightarrow$ Tức là chuỗi thời gian không dừng.

Với $DF_{Critical}^5 = 0.95$

Kiểm định Augmented Dickey Fuller

Tương tự như kiểm định Dickey Fuller nhưng không lấy giá trị chỉ thời điểm trước đó 1 đơn vị mà lấy tất cả các thời điểm trước đó. Và được biểu diễn như sau:

$$\Delta y_t = p_0 + \delta y_{t-1} + \sum_{i=1}^{t-2} \beta_i y_{t-i} + \varepsilon$$

⁵ https://en.wikipedia.org/wiki/Augmented_Dickey%E2%80%93Fuller_test

CƠ SỞ LÝ THUYẾT

2.2.3. Mô hình tự hồi quy (Autoregression – AR)

Mô hình AR(p) là mô hình tự hồi quy dựa vào các chu kỳ trước

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \dots + a_p y_{t-p} + \varepsilon_t$$

Trong đó

- y_t : Là giá trị quan sát tại thời điểm hiện tại
- a_0, a_1, \dots, a_p : Các hệ số hồi quy
- $y_{t-1}, y_{t-2}, \dots, y_{t-p}$: Là giá trị quan sát tại quá khứ từ thời điểm t-p đến t-1
- ε_t : Sai số ngẫu nhiên của giai đoạn hiện tại. Giá trị trung bình được mong đợi là 0.

Mô hình AR(p) là một mô hình hồi quy tuyến tính của những quan sát dùng t-p đến t-1 trong quá khứ.

Số quan sát dùng quá khứ sử dụng trong mô hình hàm tự tương quan là bậc p của mô hình AR. Nếu ta sử dụng **n** *quan sát* quá khứ, ta có mô hình tương quan bậc hai AR(n).

Điều kiện dùng của mô hình AR(p) là: $\sum_{i=0}^p a_i < 1$

2.2.4. Mô hình trung bình trượt (Moving Average – MA)

Quan sát dùng hiện tại y_t là một hàm tuyến tính phụ thuộc các biến sai số dự báo quá khứ và hiện tại. Mô hình bình quân di động là một trung bình trọng số của những sai số mới nhất.

$$y_t = \beta_0 + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \dots + \beta_q \varepsilon_{t-q} + \mu_t$$

Trong đó

- y_t : Là giá trị quan sát tại thời điểm hiện tại
- b_0, b_1, \dots, b_p : Các hệ số hồi quy
- $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-p}$: Là sai số quan sát tại quá khứ từ thời điểm t-p đến t-1
- ε_t : Sai số ngẫu nhiên của giai đoạn hiện tại. Giá trị trung bình được mong đợi là 0.

CƠ SỞ LÝ THUYẾT

Mô hình MA(q) là một mô hình hồi quy tuyến tính của những quan sát dùng $t-q$ đến $t-1$ trong quá khứ.

Số quan sát dùng quá khứ sử dụng trong mô hình hàm tự tương quan là bậc q của mô hình MA. Nếu ta sử dụng **n quan sát** quá khứ, ta có mô hình tương quan bậc hai MA(n).

Điều kiện dùng của mô hình MA(q) là: $\sum_{i=0}^q \beta_i < 1$

2.2.5. Sai phân I

Sai phân chỉ sự khác nhau giữa giá trị hiện tại và giá trị trước đó. Phân tích sai phân nhằm làm cho ổn định giá trị trung bình của chuỗi dữ liệu, giúp cho việc chuyển đổi chuỗi thành một chuỗi dừng.

Sai phân lần 1 I(1): $\Delta y_t = y_t - y_{t-1}$

Sai phân lần 2 I(2): $\Delta(\Delta y_t) = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$

Sai phân lần d được ký hiệu là I(d)

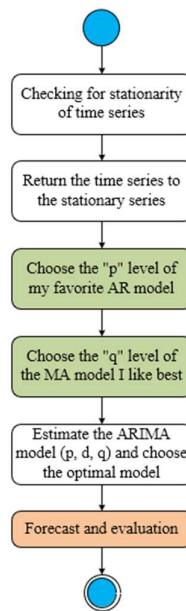
2.2.6. Mô hình Box-Jenkins⁶

Mô hình Box-Jenkins là mô hình gồm 6 bước và áp dụng cho mô hình ARIMA

- **Bước 1:** Kiểm tra tính dừng bằng phương pháp ADF hoặc KPSS.
- **Bước 2:** Nếu là chuỗi dừng thì bỏ qua bước 2. Nếu là chuỗi dừng thì thực hiện bước biến đổi chuỗi không dừng thành chuỗi dừng.
- **Bước 3:** Chọn bậc p trong mô hình AR(p) để tối ưu
- **Bước 4:** Chọn bậc q trong mô hình MA(q) để tối ưu
- **Bước 5:** Ước lượng p, d, q sao cho mô hình ARIMA(p,d,q) sau đó chọn mô hình tối ưu.
- **Bước 6:** Dự báo

⁶ https://en.wikipedia.org/wiki/Box%E2%80%93Jenkins_method

CƠ SỞ LÝ THUYẾT



Hình 2.6 Mô hình Box-Jinkins áp dụng cho mô hình ARIMA

2.2.7. Chỉ số AIC

Một trong những tiêu chí thường được sử dụng để lựa chọn mô hình đó là chỉ số AIC (Akaike Information Criteria). Tiêu chí thông tin này là một công cụ ước tính lỗi dự báo và do đó đánh giá chất lượng tương đối của các mô hình thống kê trên một tập hợp dữ liệu nhất định. AIC được hình thành dựa trên lý thuyết thông tin (information theory). Khi một mô hình thống kê được sử dụng để dự báo, kết quả sẽ gần như không bao giờ chính xác hoàn toàn. Vì vậy một số thông tin sẽ bị mất do không thể dự báo từ mô hình. AIC ước tính lượng thông tin tương đối bị mất bởi một mô hình nhất định: mô hình mất càng ít thông tin thì chất lượng của mô hình đó càng cao. Giả sử rằng chúng ta có một mô hình thống kê tương ứng với một bộ dữ liệu. Gọi là số lượng tham số ước tính trong mô hình. Đặt là giá trị tối đa của hàm hợp lý (maximum likelihood function) của mô hình.

$$AIC = 2k - 2 \ln(Likelihood)$$

Với k là số tham số quan sát hồi quy, $likelihood^7$ là hàm hợp lý cực đại

⁷ <https://viblo.asia/p/so-luoc-ve-maximum-likelihood-estimation-1Je5EvrYKnL>

CƠ SỞ LÝ THUYẾT

2.3. Support Vector Machine và Module SVR

2.3.1. Lý thuyết cơ sở SVM [8]

Khoảng cách từ một điểm đến một mặt phẳng trong không gian

Trong không gian 2 chiều Oxy ta có khoảng cách từ một điểm $(x_0; y_0)$ đến một đường thẳng $w_1x + w_2y + b = 0$ được xác định bởi công thức:

$$\frac{|w_1x_0 + w_2y_0 + b|}{\sqrt{w_1^2 + w_2^2}}$$

Trong không gian 3 chiều Oxyz ta có khoảng cách từ một điểm (x_0, y_0, z_0) đến một mặt phẳng $w_1x + w_2y + w_3z + b = 0$ được xác định bởi công thức:

$$\frac{|w_1x_0 + w_2y_0 + w_3z_0 + b|}{\sqrt{w_1^2 + w_2^2 + w_3^2}}$$

Tương tự vậy trong không gian nhiều chiều T ta có khoảng cách từ một điểm X_0 đến một siêu phẳng $w^T X + b = 0$ được xác định bởi công thức:

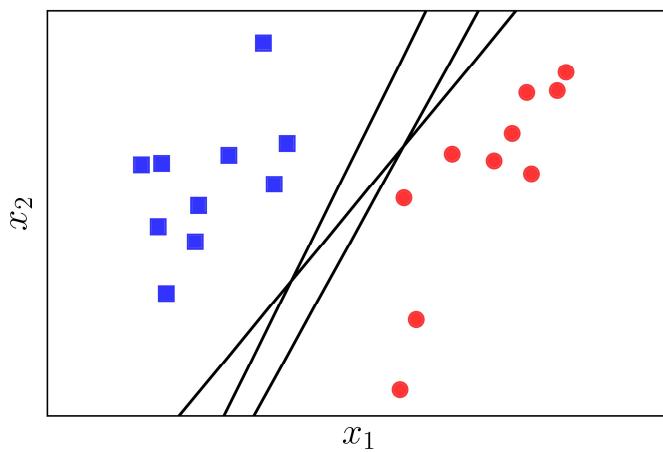
$$\frac{|w^T X + b|}{\sqrt{\sum_{t=1}^T w_t^2}} \quad (\text{Với } T \text{ là số chiều})$$

Bài toán phân chia hai lớp sử dụng SVM

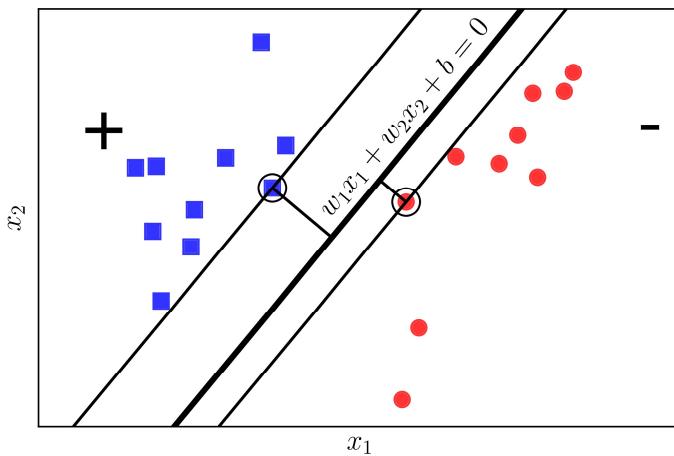
Giả sử có hai lớp khác nhau được mô tả bởi không gian nhiều chiều, hai lớp này tồn tại một linearly separable (tức là tồn tại ít nhất một siêu phẳng phân chia hai lớp)

Dựa vào hình mô tả trên không gian (Hình 2.7), đối với hai class màu xanh và màu đỏ có vô số siêu phẳng phân chia hai lớp.

CƠ SỞ LÝ THUYẾT



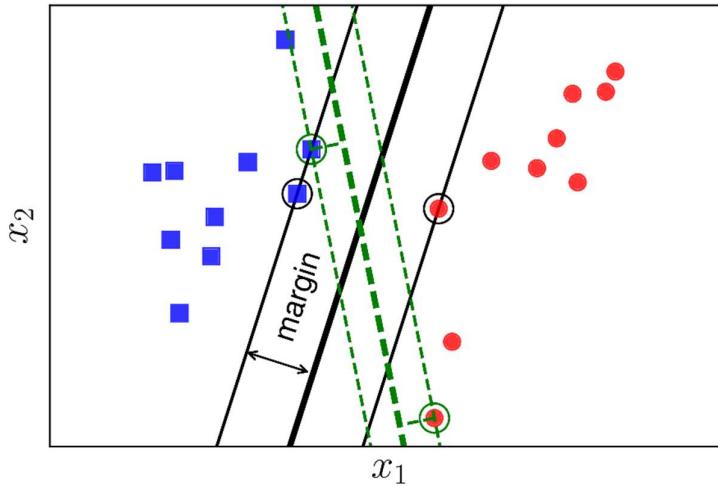
Hình 2.7 Mô tả hai lớp trên không gian được phân chia bởi nhiều siêu phẳng
Câu hỏi đặt ra là trong các siêu phẳng, đâu là mặt siêu phẳng phân chia tốt nhất



Hình 2.8 Mặt phân chia với điểm gần nhất của mỗi class

Đối với Hình 2.8, mức **độ hạnh phúc** của một điểm tỉ lệ thuận với khoảng cách gần nhất từ một điểm của một class đến mặt phân chia. Nếu khoảng cách càng gần thì **độ hạnh phúc** sẽ càng cao. Ngoài độ hạnh phúc ra thì mặt phân chia giữa hai class phải là **công bằng** tức là **độ hạnh phúc** của mỗi class phải là như nhau. Khoảng cách như nhau được gọi là **margin (lề)**.

CƠ SỞ LÝ THUYẾT



Hình 2.9 Margin của các class

Với hai lớp được phân chia bởi một siêu phẳng thì có vô số siêu phẳng để phân chia hai lớp có margin bằng nhau. Nhìn vào Hình 2.9 ta thấy đường đen có độ phân chia hai lớp với margin cao hơn đường màu xanh nét đứt, điều này làm cho việc phân lớp của đường màu đen trở nên rõ ràng hơn đường màu xanh.

Bài toán tối ưu trong SVM tức là tìm đường phân chia giữa hai class sao cho margin đạt giá trị lớn nhất.

2.3.2. Xây dựng bài toán tối ưu sử dụng SVM

Giả sử rằng các cặp dữ liệu *training set* được biểu diễn như sau:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

Với $x_i \in \mathbb{R}^d$ thể hiện d dữ liệu đầu vào (Input)

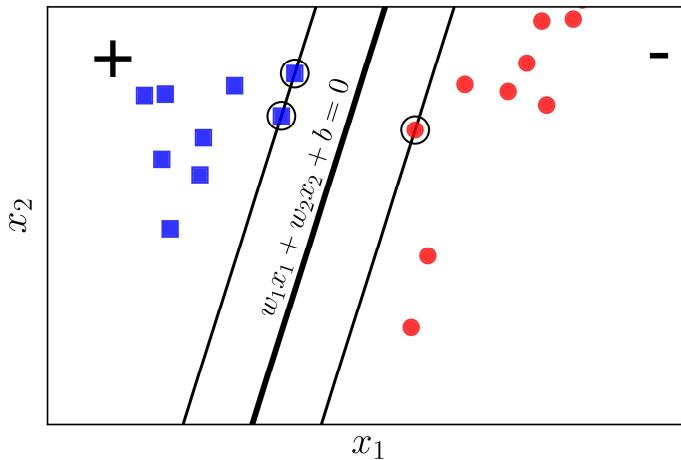
y_i thể hiện nhãn (label) hay dữ liệu đầu ra (Output)

N là số tập dữ liệu *training set*

Giá trị nhãn của mỗi điểm được xác bởi $y_i = 1$ (class 1) hoặc $y_i = -1$ (class 2)

CƠ SỞ LÝ THUYẾT

Để dễ dàng mô tả cho thuật toán trên nhiều chiều, đầu tiên ta xét trường hợp trên không gian hai chiều như hình minh họa (Hình 2.10)



Hình 2.10 Phân tích bài toán SVM trên không gian 2 chiều

Giả sử hình vuông màu xanh thuộc class 1 (+), hình tròn màu đỏ thuộc class 2 (-)

$$\text{Và mặt phẳng } w^T X + b = w_1 x_1 + w_2 x_2 + b = 0$$

Với cặp dữ liệu (x_n, y_n) bất kì, khoảng cách từ điểm dữ liệu đó đến mặt phân cách được xác định bởi công thức

$$\frac{y_n (w^T x_n + b)}{\sqrt{\sum_{t=1}^T w_t^2}}$$

Điều này cho thấy rằng (x_n, y_n) luôn cùng nằm một phía nên tử số không âm

Với mặt phân chia như trên, margin được xác định từ điểm gần nhất tới mặt (bất kể điểm nào trong hai class và được xác định bởi công thức:

CƠ SỞ LÝ THUYẾT

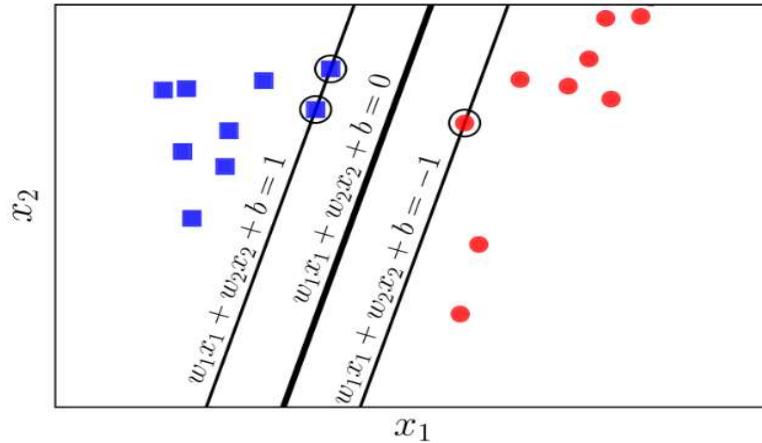
$$m \arg \min_{(w,b)} = \min_n \frac{y_n(w^T x_n + b)}{\sqrt{\sum_{i=1}^T w_i^2}}$$

Bài toán tối ưu SVM chính là bài toán tìm w, b sao cho margin đạt giá trị lớn nhất

$$(w, b) = \arg \max_{(w,b)} \left\{ \min_n \frac{y_n(w^T x_n + b)}{\sqrt{\sum_{i=1}^T w_i^2}} \right\} = \arg \max_{(w,b)} \left\{ \frac{1}{\sqrt{\sum_{i=1}^T w_i^2}} \min_n y_n(w^T x_n + b) \right\}$$

Nếu thay w thành kw và b thành kb , đó k là một hằng số dương thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức *margin* không đổi. Dựa trên tính chất này, ta có thể giả sử:

$$\min_n y_n(w^T x_n + b) = 1$$



Hình 2.11 Các điểm gần mặt phân cách được biểu diễn khoanh tròn

Vậy ta có với mọi n thì $y_n(w^T x_n + b) \geq 1$

Trong python thư viện hỗ trợ SVM là gói sklearn với thư viện con là SVM

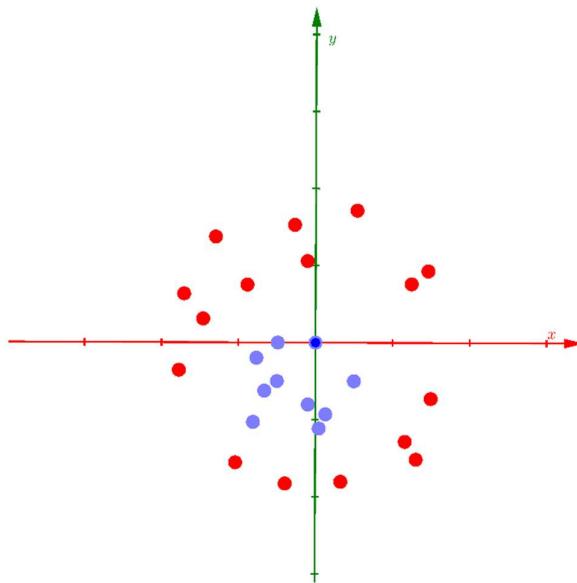
CƠ SỞ LÝ THUYẾT

2.3.3. Kernel SVM [9]

Đối với bài toán SVM không có linearly separable thì có một phương pháp khác để phân chia class, phương pháp đó gọi là Kernel SVM.

Ý tưởng cơ bản của Kernel SVM và các phương pháp kernel nói chung là tìm một phép biến đổi sao cho dữ liệu ban đầu là *không phân biệt tuyến tính* được biến sang không gian mới. Ở không gian mới này, dữ liệu trở nên *phân biệt tuyến tính*.

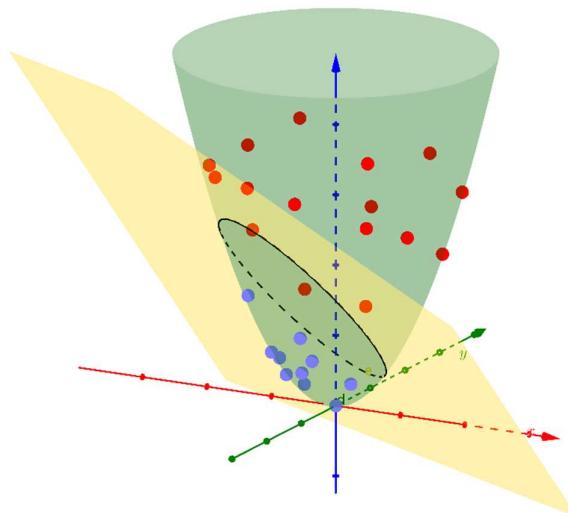
Xét ví dụ dưới đây với việc biến dữ liệu không phân biệt tuyến tính trong không gian hai chiều thành phân biệt tuyến tính trong không gian ba chiều bằng cách giới thiệu thêm một chiều mới:



Hình 2.12 Dữ liệu 2 chiều không có Linearly Separable

Dữ liệu 2 chiều ở Hình 1.5 không có tính phân chia tuyến tính được vì vậy sử dụng phương pháp Kernel SVM bằng cách thêm một chiều mới

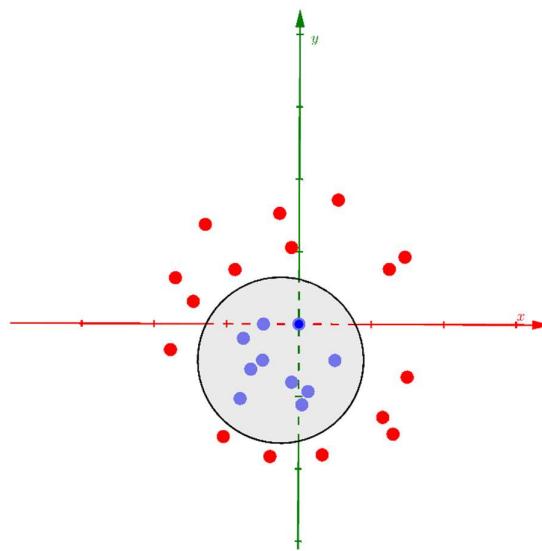
CƠ SỞ LÝ THUYẾT



Hình 2.13 Dữ liệu 3 chiều đã có Linearly Separable

Nếu có chiều thứ 3 với dạng $z = x^2 + y^2$ các điểm dữ liệu sẽ được phân bố parabolic và dữ liệu trở nên tuyến tính. Mặt phẳng màu vàng là mặt phân chia hai lớp.

Giao điểm của mặt phẳng tìm được và mặt parabolic là một đường ellipse, khi chiếu toàn bộ dữ liệu cũng như đường ellipse này xuống không gian hai chiều ban đầu, ta đã tìm được đường phân chia hai classes.



Hình 2.14 Dữ liệu đã được phân lớp

CƠ SỞ LÝ THUYẾT

Một số dạng Kernel thông dụng

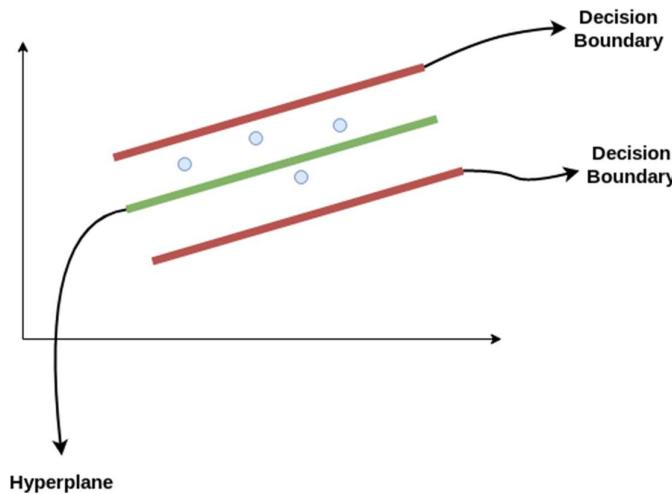
Tên	Công thức	kernel	Thiết lập hệ số
linear	$\mathbf{x}^T \mathbf{z}$	'linear'	không có hệ số
polynomial	$(r + \gamma \mathbf{x}^T \mathbf{z})^d$	'poly'	$d: \text{degree}$, $\gamma: \text{gamma}$, $r: \text{coef0}$
sigmoid	$\tanh(\gamma \mathbf{x}^T \mathbf{z} + r)$	'sigmoid'	$\gamma: \text{gamma}$, $r: \text{coef0}$
rbf	$\exp(-\gamma \ \mathbf{x} - \mathbf{z}\ _2^2)$	'rbf'	$\gamma > 0: \text{gamma}$

Hình 2.15 Một số Kernel thông dụng

2.3.4. Module SVR

Support Vector Regression (SVR) là một thuật toán hồi quy sử dụng SVM, để dự báo cho giá trị một biến liên tục (dữ liệu thời gian).

Ý tưởng của Support Vector Regression, tìm ra một hàm số xấp xỉ ứng với dữ liệu số nhập đầu vào



Hình 2.16 Mô hình hồi quy SVR

Dựa vào Hình 2.16 ta thấy hai đường màu đỏ là hai đường biên của mỗi lớp, đường biên Hyperplane là một siêu phẳng dùng để ngăn cách hai lớp.

Giả sử siêu phẳng Hyper plane có phương trình $y = wx + b$

CƠ SỞ LÝ THUYẾT

Gọi a là khoảng cách từ siêu phẳng đến các điểm giới hạn Decision Boundary với “ a ” thì ta nói điểm đó thuộc giới hạn trên, và Decision Boundary là “ $-a$ ” thì ta nói điểm đó thuộc giới hạn dưới.

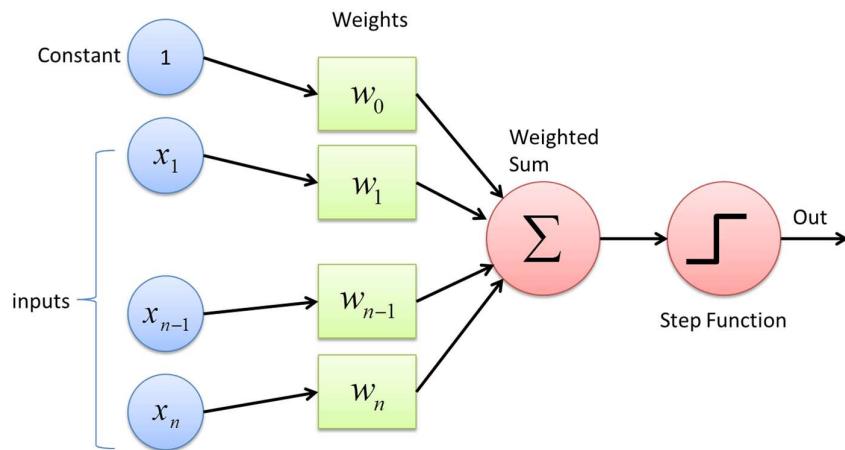
Mục tiêu của phương pháp SVR là tìm kiếm một hàm f có dạng như trên sao cho độ lệch giữa giá trị của hàm f và giá trị thực tế tại các thời điểm không vượt quá độ lệch ε và f là “phẳng” nhất có thể (as flat as possible). Hàm f càng phẳng thì số lượng các điểm dữ liệu có độ lệch lớn hơn ε càng tăng và ngược lại.

Trong đó thuật giải SVM thường không chia một cách tuyến tính được mà nó xen kẽ lẫn nhau, do đó SVR cũng phải sử dụng Kernel SVM để biến đổi các không gian không phân biệt tuyến tính (Non-Linearly Separable) thành các không gian phân biệt tuyến tính (Linearly Separable) bằng cách sử dụng các hàm số thông dụng. Vì vậy biểu diễn SVR trong mô hình chuỗi thời gian được biểu diễn bởi công thức sau đây:

$$y = K(wx + b)$$

2.4. Neural Network và mô hình LSTM [12], [13], [14] [15]

2.4.1. Cơ sở lý thuyết Neural Network



Hình 2.17 cấu tạo một mạng Neural Network [10]

CƠ SỞ LÝ THUYẾT

Một mạng Neural Network có n đầu vào mỗi ngõ vào, mỗi ngõ vào x_i có trọng số w_i .

Các trọng số w_1, w_2, \dots, w_n thay đổi trong quá trình học của mạng Neural Network

Weighted Sum của một lớp ẩn được xác định như sau:

$$\sum_{i=1}^n x_i w_i$$

Một hệ số quan trọng là w_0 được gọi là bias, hệ số này rất quan trọng vì nếu phương trình Weighted Sum không có hệ số tự do w_0 thì phương trình của sẽ trở thành $\sum_{i=1}^n x_i w_i = 0$, phương trình này luôn qua gốc tọa độ nên không tổng quát của một phương trình đường thẳng nên ta cộng thêm một hệ số tự do để phương trình Weighted Sum có dạng sau đây:

$$\sum_{i=1}^n x_i w_i + w_0$$

Vì hàm Weighted Sum là một hàm tuyến tính, không đúng với thực tế trong quá trình dự báo vì mô hình dự báo không bao giờ ở dạng tuyến tính mà nó tăng hoặc giảm theo một hàm phi tuyến vì vậy cần có một hàm hardlim để biến hàm tuyến tính thành hàm phi tuyến. Nên ta có giá trị y dự báo được xác định như sau:

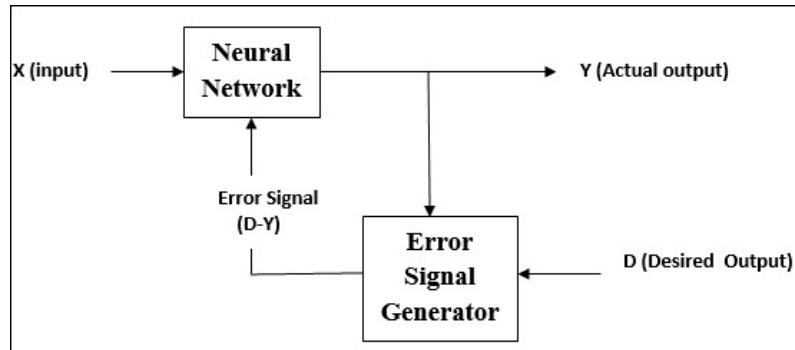
$$\hat{y} = f(\sum_{i=1}^n x_i w_i + w_0)$$

Mạng Neural nhân tạo được huấn luyện theo 3 kiểu học của máy học là học có giám sát (Supervised learning), học không giám sát (Unsupervised learning) và học tăng cường (Reinforcement learning)

Học có giám sát (Supervised Learning): Mạng được cung cấp sẵn các lập mẫu $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ trong đó x_i là giá trị đầu vào (Input), y_i là giá trị

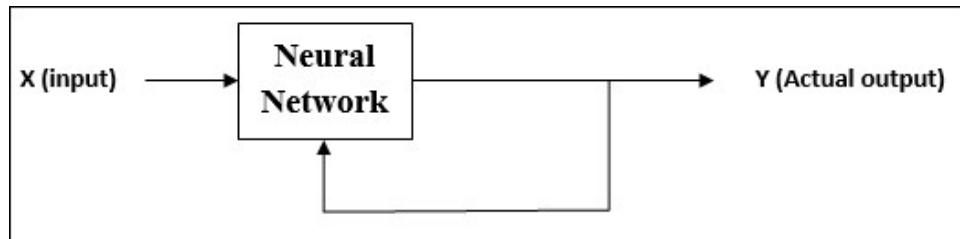
CƠ SỞ LÝ THUYẾT

đầu ra hay còn gọi là nhãn (Output). Một số bài toán điển hình cho kiểu học này là nhận diện khuôn mặt.



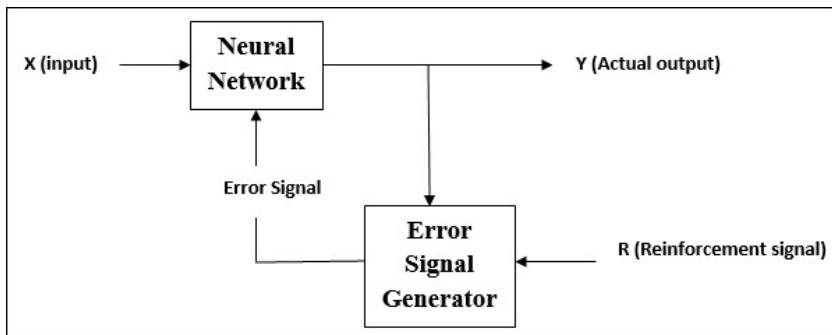
Hình 2.18 Mô hình Mạng Neural theo Supervised Learning

Học không giám sát (Unsupervised Learning): Không có thông tin phản hồi nào từ môi trường (không có sẵn *label*). Mạng chỉ nhận dữ liệu đầu vào tự tìm ra các quy tắc, đặc tính, sự tương quan có trong mẫu để đưa ra dữ liệu đầu ra (Output)



Hình 2.19 Mô hình Mạng Neural theo Unsupervised Learning

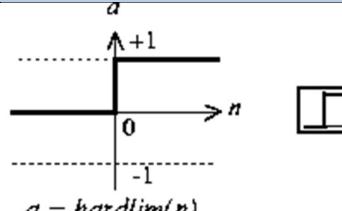
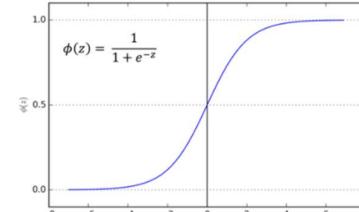
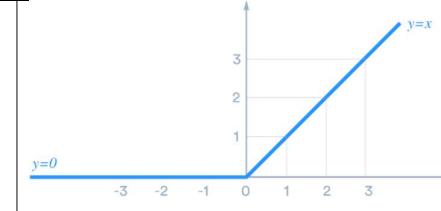
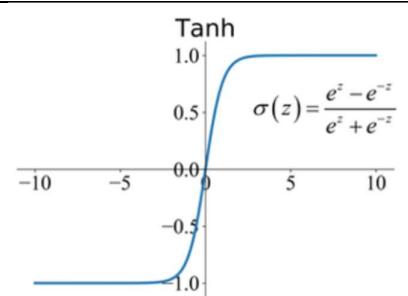
Học tăng cường (Reinforcement Learning): Mạng Neural học để tối ưu hóa hàm lỗi sau nhiều lần để đạt kết quả tối ưu.



Hình 2.20 Mô hình Mạng Neural theo kiểu học tăng cường

CƠ SỞ LÝ THUYẾT

Một số hàm kích hoạt thông dụng

Tên hàm	Công thức hàm	Biểu diễn hàm
Hàm bước nhảy đơn vị hay có tên khác là hàm hardlim	$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$	 <p>$a = \text{hardlim}(n)$</p> <p>Hard Limit Transfer Function</p>
Hàm sigmod (logistic)	$f(x) = \frac{1}{1 + e^{-x}}$	
Hàm ReLu	$f(x) = \max(0, x)$	
Hàm tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	

Bảng 2.1 Một số hàm kích hoạt thông dụng

Quá trình truyền xuôi

Là quá trình tính giá trị $\hat{y} = f(\sum_{i=1}^n x_i w_i + w_0)$, giá trị \hat{y} này thay đổi lại sau

mỗi quá trình truyền xuôi do w_i và **bias** thay đổi.

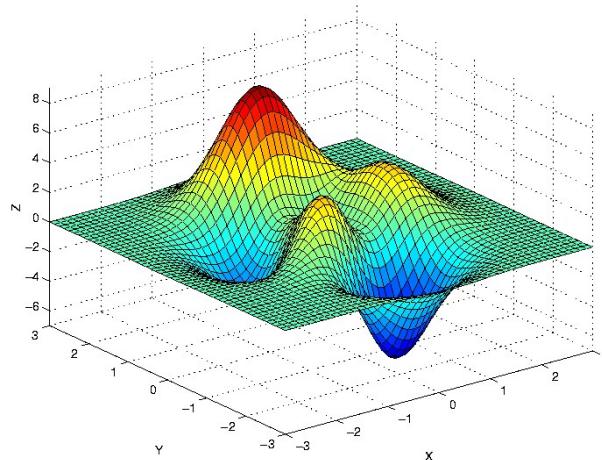
CƠ SỞ LÝ THUYẾT

Quá trình truyền ngược

Mục tiêu của quá trình truyền ngược là điều chỉnh weight trong mạng theo tỷ lệ với mức độ đóng góp lỗi chung trong mạng, nếu quá trình lặp đi lặp lại và cập nhật weight giúp việc một mạng có việc có thể dự báo được một kết quả tốt.

Ta có hàm chi phí (value function) để đánh giá mô hình dự báo MSE.

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - (mx_i + b))^2$$



Hình 2.21 Hàm chi phí trong không gian 3 chiều

Để tối ưu hàm chi phí (value function) thì ta tối ưu bằng cách tính đạo hàm của hàm chi phí $f'(m, b)$ đặt m là trọng số của node x_i thay gì viết là w_i

Ta có

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - (w_i x_i + b))^2 = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i^2 - 2\hat{y}_i(w_i x_i + b) + (w_i x_i)^2 + 2w_i x_i b + b^2)$$

- Đạo hàm riêng theo m: $f'_m MSE = \frac{1}{N} \sum_{i=1}^N -2x_i(\hat{y}_i - (w_i x_i + b))$
- Đạo hàm riêng theo b: $f'_b MSE = \frac{1}{N} \sum_{i=1}^N -2(\hat{y}_i - (w_i x_i + b))$

CƠ SỞ LÝ THUYẾT

- Đạo hàm của hàm $f'(m,b) = \begin{bmatrix} f'_m(m,b) \\ f'_b(m,b) \end{bmatrix}$
- Để tối ưu hàm chi phí ta tối ưu trọng số weight và tối ưu bias
- Weight mới = weight hiện tại - $f'_m(\text{MSE}) * \text{learning_rate}$
- Bias mới = bias hiện tại - $f'_b(\text{MSE}) * \text{learning_rate}$

Ý tưởng đưa liệu đầu vào neural network vào bài toán dự báo giá Bitcoin [11]

Từ dữ liệu file CSV lấy 31 ngày trong tháng 1 năm 2021 được thể hiện dưới bảng sau đây:

Ngày	Closing Price	Ngày	Closing Price
1/1/2021	29111.52157	1/17/2021	36016.77961
1/2/2021	29333.60512	1/18/2021	36375.81138
1/3/2021	32154.16736	1/19/2021	36346.6095
1/4/2021	33002.53643	1/20/2021	36577.51965
1/5/2021	31431.61228	1/21/2021	35004.53263
1/6/2021	34433.60651	1/22/2021	30606.18268
1/7/2021	36275.75635	1/23/2021	33368.36593
1/8/2021	39713.50786	1/24/2021	32070.09743
1/9/2021	40519.4486	1/25/2021	32285.72613
1/10/2021	40258.92399	1/26/2021	32500.25596
1/11/2021	38709.76537	1/27/2021	32324.55565
1/12/2021	34409.64238	1/28/2021	30534.99937
1/13/2021	34214.61026	1/29/2021	33408.21834
1/14/2021	37017.0075	1/30/2021	34842.55739
1/15/2021	38435.86351	1/31/2021	34622.37323
1/16/2021	36751.58497		

Bảng 2.2 Dữ liệu nhập Input Bitocin

CƠ SỞ LÝ THUYẾT

Giai đoạn 1: Pha học

Phát biểu bài toán lấy 10 ngày đầu tiên để từ ngày 1/1 đến ngày 5/1 để dự báo cho ngày 6/1. Sau đó tiến lên một ngày để dự báo từ ngày 2/1 đến ngày 6/1 để dự báo cho ngày 7/1. Cứ tiếp tục đến vậy cho đến ngày 23/1, những ngày còn lại để kiểm tra độ chính xác của thuật toán

Ta sẽ được bảng dữ liệu học như sau:

Đầu vào	X ₁	X ₂	X ₃	X ₄	X ₅	Đầu ra
K = 1	29111.521 57	29111.521 57	29111.521 57	29111.521 57	29111.521 57	34433.606 51
K = 2	29333.605 12	29333.605 12	29333.605 12	29333.605 12	29333.605 12	36275.756 35
K = 3	32154.167 36	32154.167 36	32154.167 36	32154.167 36	32154.167 36	39713.507 86
....						
K = 18	36375.811 38	36375.811 38	36375.811 38	36375.811 38	36375.811 38	33368.365 93

Bảng 2.3 Dữ liệu đầu vào và đầu ra mẫu

Ta có thể mô tả bài toán với bộ tập dữ liệu học như sau:

$$\text{Với } K=1 \text{ ta có: } Y_1 = w_{11}X_1 + w_{12}X_2 + w_{13}X_3 + w_{14}X_4 + w_{15}X_5$$

$$\text{Tương tự } K=2 \text{ ta có: } Y_2 = w_{21}X_1 + w_{22}X_2 + w_{23}X_3 + w_{24}X_4 + w_{25}X_5$$

....

Tổng quát với K = 18 ta có:

$$Y_{18} = w_{181}X_1 + w_{182}X_2 + w_{183}X_3 + w_{184}X_4 + w_{185}X_5$$

CƠ SỞ LÝ THUYẾT

Biểu diễn $Y = W \cdot X$ với $Y = [y_1, y_2, y_3, \dots, y_{18}]^T$, $X = [X_1, X_2, X_3, X_4, X_5]^T$

Từ đây ta tìm được ma trận $W = YX^{-1}$

Giai đoạn 2: Pha chạy

Đã có tập trọng số W

Cho tập dữ liệu từ ngày [1/1/2021 ... 31/1/2021]

Cho tập giá trị Closing Price tương ứng với từng ngày

[29111.52157... 34622.37323]

Xác định giá Closing Price y tức là dự báo giá Closing Price ngày
1/2/2021

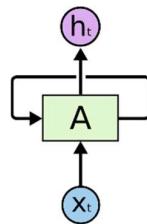
Để dự báo ngày 2/2/2021 thực hiện hai bước:

Bước 1: Lấy dữ liệu ngày 1/2/2021 cho học lại

Bước 2: Lấy dữ liệu từ ngày 28/1/2021 – 1/2/2021 để dự báo
cho ngày 2/2/2021

2.4.2. Mạng RNN và Mạng LSTM

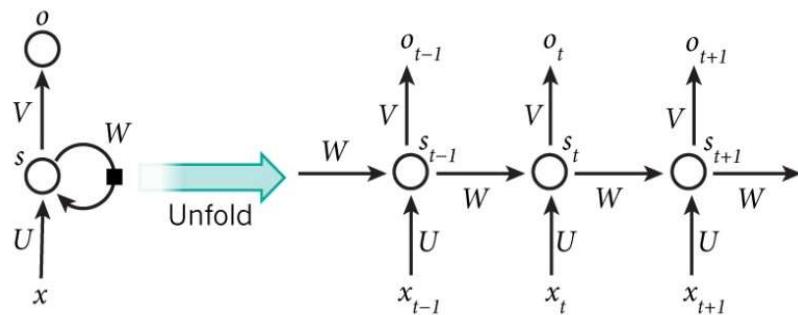
Mạng nơ-ron truyền thống không có khả năng ghi nhớ các khoảng thời gian trước đó gây ra nhược điểm của nó, với các mô hình chuỗi thời gian, việc lưu giữ bộ nhớ là rất quan trọng đối với các giá trị tại thời điểm trước đó để tìm ra quy luật của chuỗi thời gian. Từ đó, một Recurrent Neural Network (RNN) ra đời để giải quyết vấn đề này. Kiến trúc của một RNN đơn giản được hiển thị như sau:



Hình 2.22 Recurrent Neural Network có vòng lặp [16]

CƠ SỞ LÝ THUYẾT

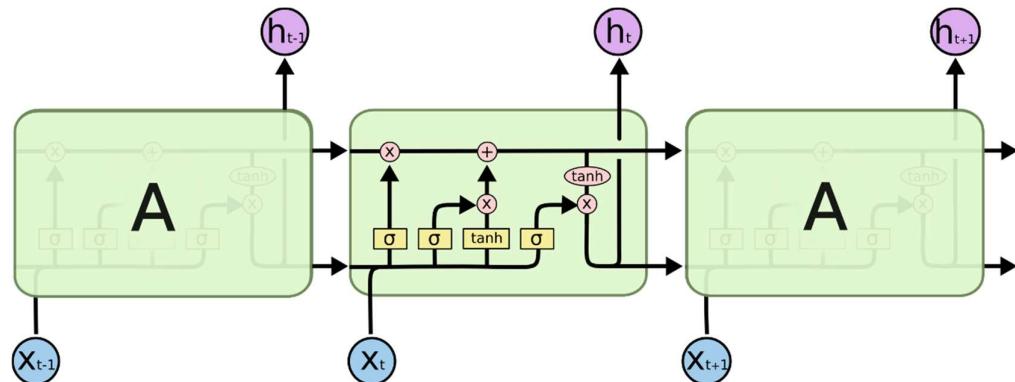
Kiến trúc Recurrent Neural Network được biểu diễn đơn giản như sau:



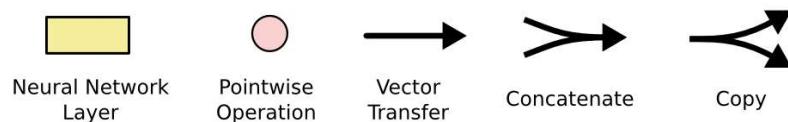
Hình 2.23 Recurrent Neural Network đơn giản hóa

RNN tự nhiên không thể học được trong thời gian dài. Về mặt lý thuyết, RNN có thể giải quyết vấn đề phụ thuộc dài hạn bằng cách thêm các tham số, nhưng RNN không thể học các tham số này thông qua nghiên cứu. [17]

May mắn thay, mạng LSTM là một loại RNN đặc biệt và giải quyết được vấn đề phụ thuộc dài hạn. Một khối LSTM [18] được có cấu trúc như sau:



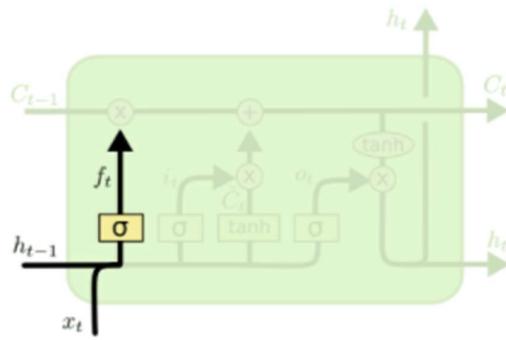
Hình 2.24 Một khối LSTM Block cơ bản



Hình 2.25 Các ký hiệu của LSTM block

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

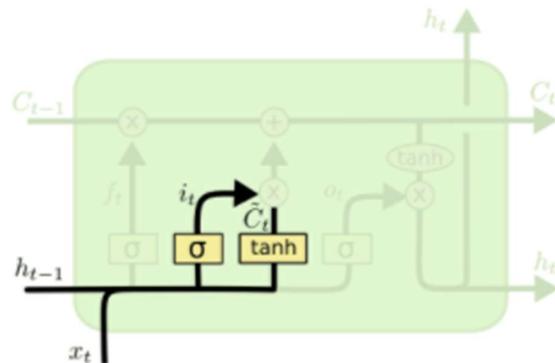
CƠ SỞ LÝ THUYẾT



Hình 2.26 Forget Gate

Tại cổng này nếu giá trị đầu ra có giá trị là 1 thì giữ tất cả thông tin và nếu là 0 thì loại bỏ các thông tin.

- Input Gate : $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ The input value gate of the LSTM block consists of the current and the previous value.

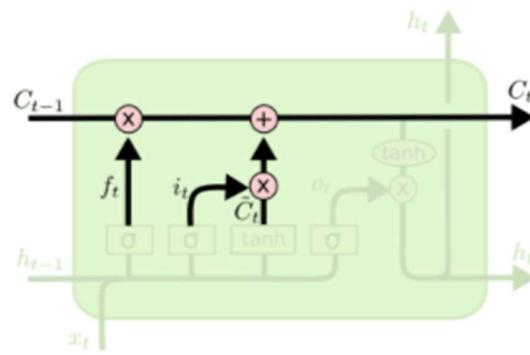


Hình 2.27 Input Gate

Tanh layer with input gate: $\tilde{C} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$

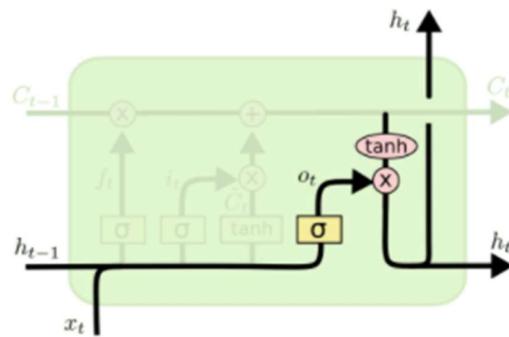
- Cell State for next input: $C_t = f_t * C_{t-1} + i_t * \tilde{C}$

CƠ SỞ LÝ THUYẾT



Hình 2.28 Merge Cell Forget Gate and Input Gate

- Output Gate: $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$ The output value gate of the LSTM block consists of the current and the previous value.



Hình 2.29 Output Gate

- New State: $h_t = o_t * \text{tanh}(\hat{C}_t)$

Mô hình LSTM phù hợp với mô hình chuỗi thời gian vì nó khám phá mối quan hệ giữa quy luật của chuỗi thời gian. Vì mô hình LSTM phù hợp với các mô hình tuần tự như mô hình chuỗi thời gian.

2.5. Mạng tăng cường và Q-Learning

Việc học tập của con người là quá trình tích lũy liên tục từ lúc học tiểu học, trung học, đại học, ở mỗi mức học con người sẽ có một phần thưởng cho sự tích lũy kiến thức của mình.

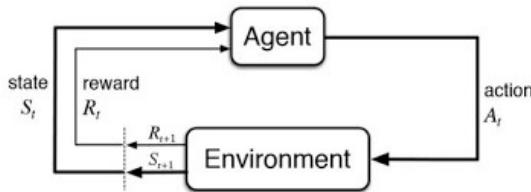
CƠ SỞ LÝ THUYẾT

Quá trình học của con người được phân làm 3 nhân tố

- Agent: Đối tượng thực thể nghiên cứu ở đây là con người.
- Environment: Môi trường học tập (tiểu học, trung học, đại học).
- Reward: Phần thưởng sau mỗi lần học tập.
- Action: Hành động cho mỗi lần học tập. Hành động này là quá trình học random.

Quá trình trial-and-error (thử và lỗi) và sau mỗi lần thử chi phí đến phần thưởng thì được gọi là học tăng cường. Việc thử lặp lại tốn rất nhiều thời gian nhưng giúp chúng ta có thêm kinh nghiệm.

Việc thăm dò càng lâu thì kinh nghiệm học càng nhiều. Có một số chiến lược để điều chỉnh sự cân bằng và thăm dò không gian tìm kiếm.



Hình 2.30 Quá trình học Q-Learning

Công thức Temporal Difference or TD-Update

$$Q(S, A) \leftarrow Q(S, A) + \alpha * (R + \gamma \max_a Q(S', A') - Q(S, A))$$

S: State hiện tại của tác nhân

A: Action hành động được đặt

S': State tiếp theo

A'': Hành động tốt nhất tiếp theo

R: Phần thưởng hiện tại được quan sát từ phản hồi của hành động hiện tại

γ thuộc $(0;1]$ chiết khấu Q-Learning Discounting Factor cho các phần thưởng thường lấy từ $0.95 - 0.99$.

α : Tỉ lệ học của Q-Learning

CƠ SỞ LÝ THUYẾT

Thuật toán Q-Learning [19]

- **Bước 1:** Init Q-Table chỉ lần đầu
- **Bước 2:** Choose Action
- **Bước 3:** Perform Action
- **Bước 4:** Measure Reward
- **Bước 5:** Update Q-Table Tới Bước 2

Ví dụ minh họa thuật toán



Bảng 2.4 Ví dụ về thuật toán Q-Learning

Bước 1: Khởi tạo bản Q-Table

State/Action	Lên	Xuống	Phải	Trái
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0

Bảng 2.5 Bảng Q-Table ban đầu

CƠ SỞ LÝ THUYẾT

Chọn những hằng số:

Tỉ lệ học α : 0.1

Tỉ lệ chiết khấu δ : 0.99

Phân thưởng

- 0: Nếu ô đó không có gì
- (+) 1: Nếu ăn được một mảnh format nhỏ
- (+) 10: Nếu ăn được bánh format lớn
- (-) 10: Nếu uống thuốc độc

Bước 2: Lấy Random 1 hành động có thể đi, chuột đang ở vị trí (1,1) nên chỉ đi được ở (1,2) và vị trí (2,1). Chọn hành động random = Right.

$$Q(11, \text{Right}) = 0 + 0.1(1 + 0.99 * 0 - 0) = 0.1$$

Bước 3, Bước 4, Bước 5: Update Q-Table

State/Action	Lên	Xuống	Phải	Trái
11	0	0	0.1	0
12	0	0	0	0
13	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0

Bảng 2.6 Bảng Q-Learning chạy lần thứ 1

Vị trí hiện tại của con chuột khi ăn xong miếng format lần 1 được cập nhật như sau:

CƠ SỞ LÝ THUYẾT

11		12	13
			
21		22	23

Bảng 2.7 Vị trí của chú chuột tại lần thứ 1

Quay lại Bước 2: Lấy Random 1 hành động có thể đi, chuột đang ở vị trí (1,2) nên chỉ đi được ở (1,1) và vị trí (2,2) và (1,3). Chọn hành động random = Down.

$$Q(12, \text{Down}) = 0 + 0.1(-10 + 0.99 * 0 - 0) = -1$$

Bước 3, Bước 4, Bước 5: Update Q-Table

State/Action	Lên	Xuống	Phải	Trái
11	0	0	0.1	0
12	0	-1	0	0
13	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0

Bảng 2.8 Bảng Q-Learning chạy lần thứ 2

Vị trí hiện tại của con chuột khi uống phải thuốc độc được cập nhật như sau:

CƠ SỞ LÝ THUYẾT

11		12		13
				
21		22		23

Vì Mouse = Deed hoặc ăn hết số format nên ta kết thúc, nên sẽ bắt đầu một sinh mạng mới tại vị trí (1,1) tại lần chạy thứ 2. Có thể cho chú chuột học 100 lần hay 1000 lần tùy theo mô hình học. Đến khi học xong chúng ta có thể dựa trên Q-Table để tìm ra hành động tốt nhất cho mỗi bước đi dựa trên chỉ số Q-Value.

2.6. Phương pháp đánh giá mô hình

Các giá trị đánh giá **RMSE**, **MAPE**, **AIC** đạt giá trị càng nhỏ càng tốt.

2.6.1. Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\sum_i^n (f_i - y_i)^2}{n}}$$

- f_i là giá trị dự báo tại thời điểm i , y_i là giá trị thực tế tại thời điểm i .
- Giá trị n là số mẫu mà ta quan sát.

2.6.2. Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{\sum_i^n \left(\frac{|y_i - f_i|}{y_i} \right)}{n}$$

- f_i là giá trị dự báo tại thời điểm i , y_i là giá trị thực tế tại thời điểm i .
- Giá trị n là số mẫu mà ta quan sát.

KẾT HỢP MÔ HÌNH ARIMA VỚI HỒI QUY, NEURAL NETWORK

Chương 3. KẾT HỢP MÔ HÌNH ARIMA VỚI HỒI QUY, NEURAL NETWORK

3.1. Lý thuyết kết hợp mô hình [20], [21], [22]

Kết hợp giữa mô hình ARIMA và ANN để dự báo chuỗi thời gian. Ý tưởng chính của đề xuất này dựa trên các thành phần tuyến tính (ARIMA) và các thành phần phi tuyến (ANN). Mô hình dự báo được hiển thị bên dưới:

$$Y_t = L_t + N_t$$

Trong đó Y_t là giá trị trong chuỗi thời gian. L_t là thành phần tuyến tính và N_t là thành phần phi tuyến.

Đầu tiên, chúng tôi sử dụng mô hình ARIMA để dự đoán các thành phần tuyến tính và các thành phần phi tuyến tính. Chúng tôi gọi chúng là phần dư lỗi, ký hiệu là e_t . Nó được tính như sau:

$$e_t = Y_t - \hat{L}_t$$

Trong đó \hat{L}_t giá trị dự đoán của mô hình ARIMA tại thời điểm t .

Chúng ta có phần dư của mô hình thu được với nút trong mạng nơron được biểu diễn với hàm f là một hàm phi tuyến như sau:

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t$$

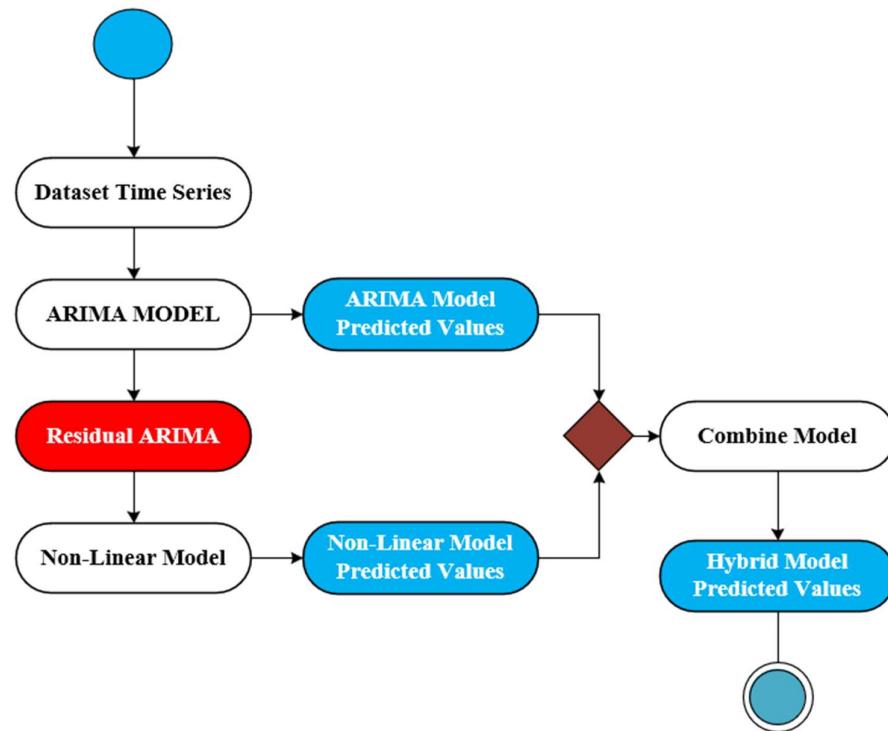
Cuối cùng, chúng tôi kết hợp mô hình tuyến tính ARIMA và mô hình phi tuyến LSTM hoặc SVR để đưa ra kết quả dự báo cuối cùng như sau:

$$\hat{Y}_t = \hat{L}_t + \hat{N}_t$$

Trong đó \hat{N}_t giá trị dự đoán của mô hình LSTM, SVR tại thời điểm t .

KẾT HỢP MÔ HÌNH ARIMA VỚI HỒI QUY, NEURAL NETWORK

3.2. Ứng dụng mô hình Hybrid ARIMA-SVR và ARIMA-LSTM



Hình 3.1 Mô hình lai giữa Linear Model và Non-Linear Model

Các bước của mô hình lai:

Bước 1: Chọn dữ liệu chuỗi thời gian về giá tiền điện tử và xử lý dữ liệu.

Bước 2: Thực hiện trên mô hình ARIMA. Đưa ra sai số dư.

Bước 2.1. Kiểm tra tính có định của dữ liệu chuỗi thời gian.

Bước 2.2 Chuyển chuỗi thời gian sang chuỗi tĩnh.

Bước 2.3 Chọn p, q, d sau đây cho mô hình ARIMA tối ưu nhất.

Bước 2.4 Đưa ra kết quả dự đoán

Bước 3: Lấy phần lỗi do sử dụng máy học hoặc mô hình hồi quy sau đó đưa ra dự báo.

Bước 4: Kết hợp các kết quả của mô hình dự báo phi tuyến tính và tuyến tính với nhau.

Bước 5: Đưa ra kết quả và đánh giá mô hình.

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Chương 4. PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

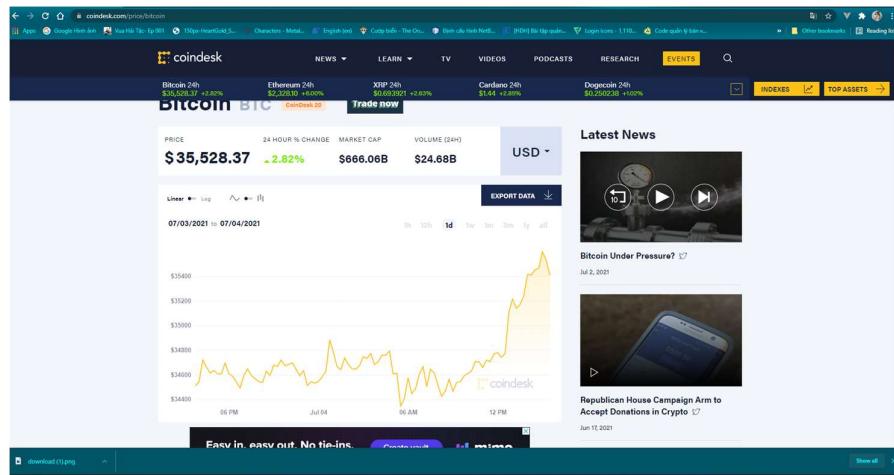
4.1. Pharse 1 – Học mô hình dự báo và khuyến nghị

4.1.1. Nguồn dữ liệu, chia train test

Trang web lấy nguồn dữ liệu để học: <https://www.coindesk.com/>

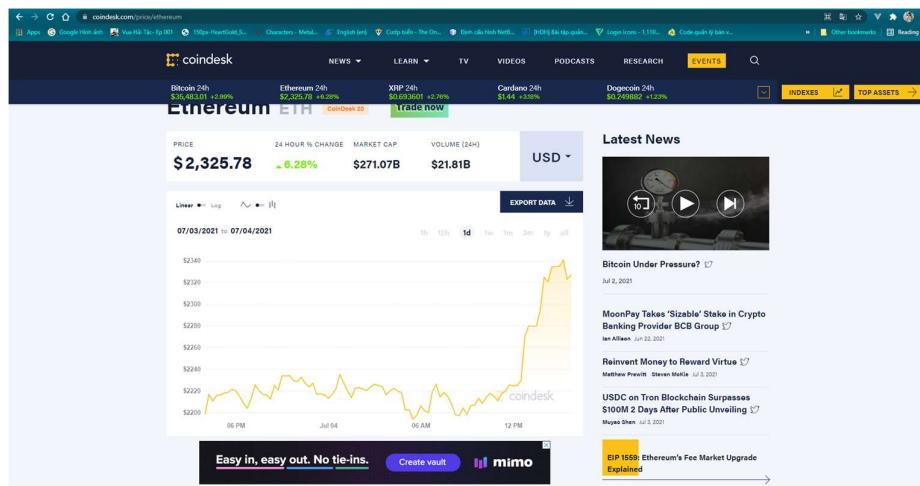
Trang web download dữ liệu Bitcoin

Chọn thời gian cần lấy dữ liệu → Export Data (CSV)



Hình 4.1 Trang Web dùng để download dữ liệu Bitcoin

Trang web download dữ liệu Ethereum



Hình 4.2 Trang Web dùng để download dữ liệu Ethereum

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Chia train/test

Ở tất cả các mô hình sẽ chia dữ liệu train:test theo tỉ lệ 8:2

Trực quan dữ liệu

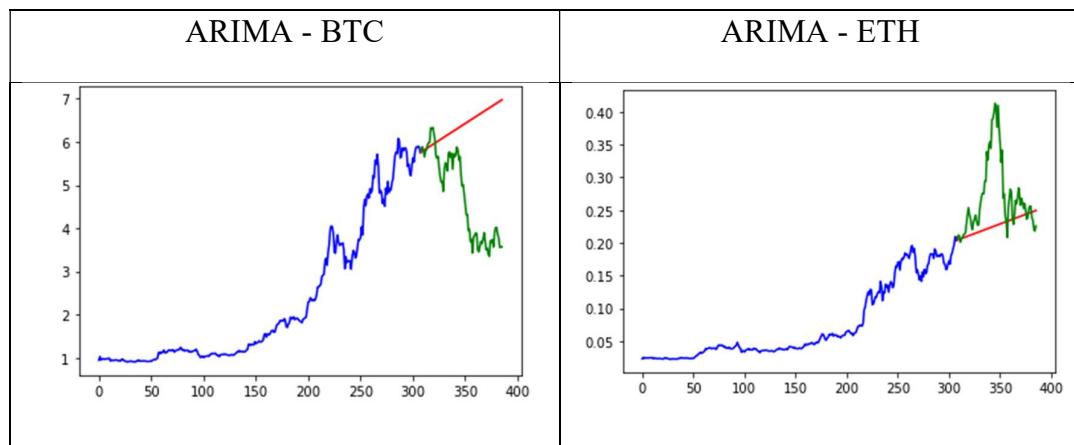


Hình 4.3 Đồ thị về Closing Price theo thời gian của đồng BTC và ETH

4.1.2. Kết quả dự báo từng mô hình dự báo

Định nghĩa: Màu xanh dương là giá trị train, màu xanh lục là giá trị test, màu đỏ là giá trị dự báo

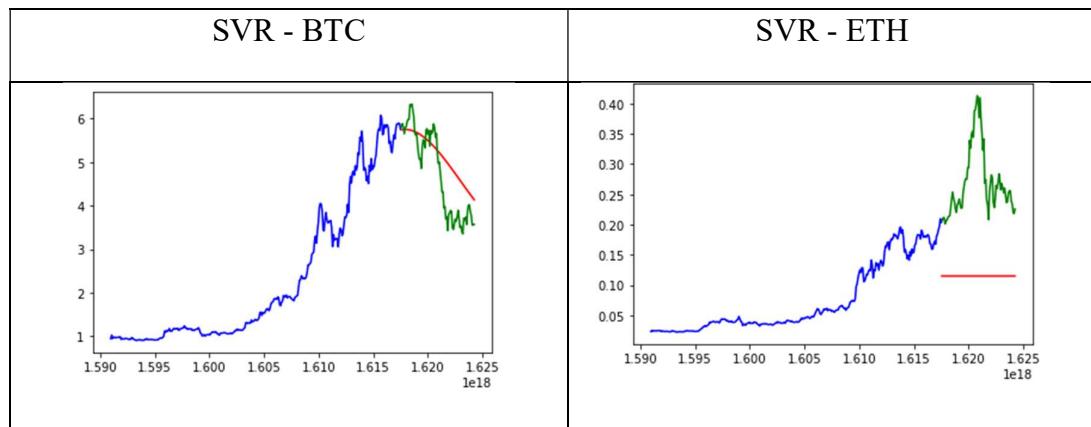
Mô hình ARIMA



Bảng 4.1 Kết quả dự báo mô hình ARIMA trên hai loại đồng tiền

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

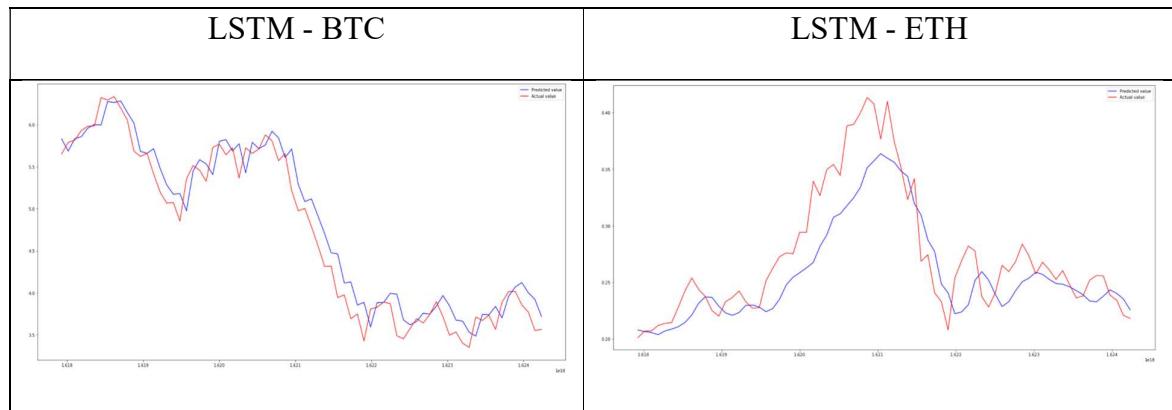
Mô hình SVR



Bảng 4.2 Kết quả dự báo mô hình SVR trên hai loại đồng tiền

Mô hình LSTM

Màu xanh là giá trị dự báo, màu đỏ là giá trị thực tế

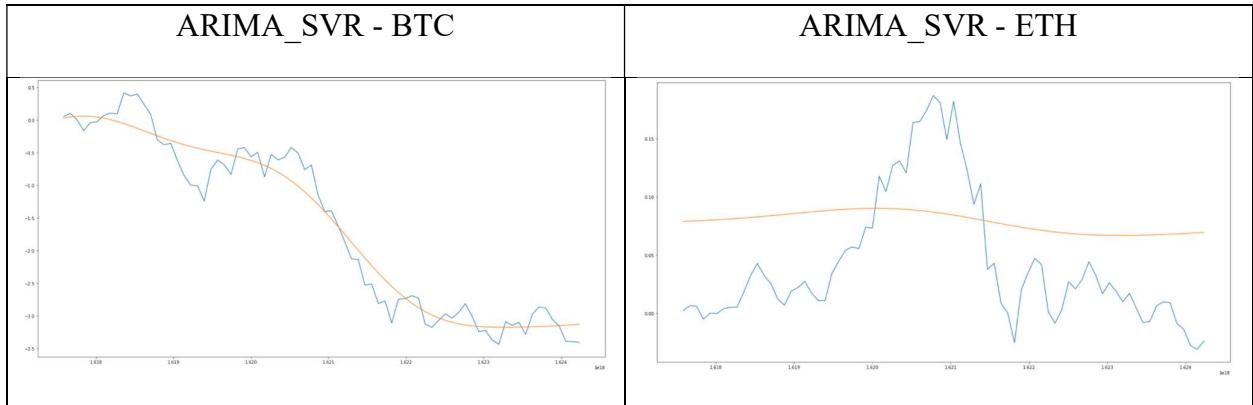


Bảng 4.3 Kết quả dự báo mô hình LSTM trên hai loại đồng tiền

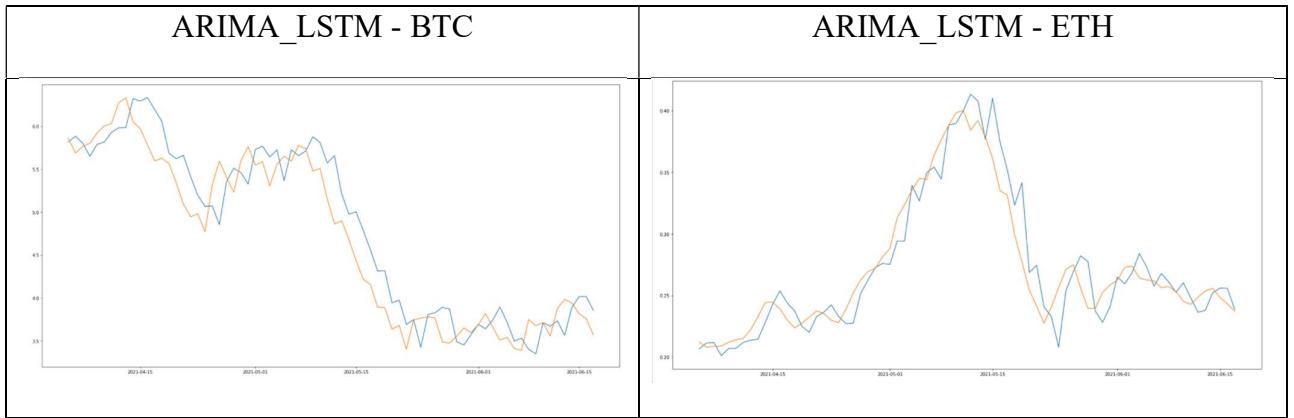
Mô hình ARIMA-SVR

Màu cam là giá trị dự báo, màu xanh là giá trị thực tế

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM



**Bảng 4.4 Kết quả dự báo mô hình ARIMA_SVR trên hai loại đồng tiền
Mô hình ARIMA-LSTM**



**Bảng 4.5 Kết quả dự báo mô hình ARIMA_LSTM trên hai loại đồng tiền
Đánh giá bằng các thông số**

- Từng mô hình đơn lẻ BTC

Model	RMSE	MAPE (%)
ARIMA	2.075	42.110
SVR	0.691	13.830
LSTM	0.231	4.178

Bảng 4.6 Error of individual models with BTC Closing Price

⇒ *Mô hình LSTM áp dụng để dự báo BTC tốt nhất. Mô hình ARIMA là mô hình không phù hợp với BTC nhất.*

- Từng mô hình đơn lẻ ETH

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Model	RMSE	MAPE (%)
ARIMA	0.071	14.736
SVR	0.164	55.437
LSTM	0.030	7.989

Bảng 4.7 Error of individual models with ETH Closing Price

⇒ Mô hình LSTM áp dụng để dự báo ETH tốt nhất. Mô hình SVR là mô hình không phù hợp với ETH nhất.

- Kết hợp mô hình BTC

Model	RMSE	MAPE (%)
ARIMA_SVR	0.275	4.644
ARIMA_LSTM	0.322	5.597

Bảng 4.8 Error of hybrid models with BTC Closing Price

⇒ Mô hình ARIMA_SVR áp dụng để dự báo BTC tốt nhất.

- Kết hợp mô hình ETH

Model	RMSE	MAPE (%)
ARIMA_SVR	0.063	23.224
ARIMA_LSTM	0.024	6.356

Bảng 4.9 Error of hybrid models with ETH Closing Price

⇒ Mô hình ARIMA_LSTM áp dụng để dự báo ETH tốt nhất.

4.1.3. Đánh giá mô hình mô hình dự báo phù hợp cho Pharse 2

Mô hình ARIMA_LSTM được đánh giá khá cao trong đồng tiền ảo ETH. Nên quyết định chọn mô hình ARIMA_LSTM cho Pharse 2 đối với đồng tiền này.

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Mô hình hybrid ARIMA_SVR tốt đối với đồng tiền BTC cho dự báo tuy nhiên về lâu dài thì mô hình LSTM có thể hoạt động tốt hơn SVR do mô hình này do tính chất học dữ liệu trước đó và ứng dụng neural network nên việc học của LSTM lâu dài cũng tốt hơn SVR. Việc sử dụng LSTM riêng lẻ chỉ làm cho mô hình hồi quy hoạt động tốt ở giai đoạn phi tuyến giả sử tại một khoảng thời điểm nào đó nó tăng theo hàm tuyến tính vậy thì việc sử dụng LSTM kết hợp ARIMA sẽ phòng ngừa trường hợp đó. Nên quyết định chọn mô hình ARIMA_LSTM cho Pharse 2 đối với đồng tiền BTC.

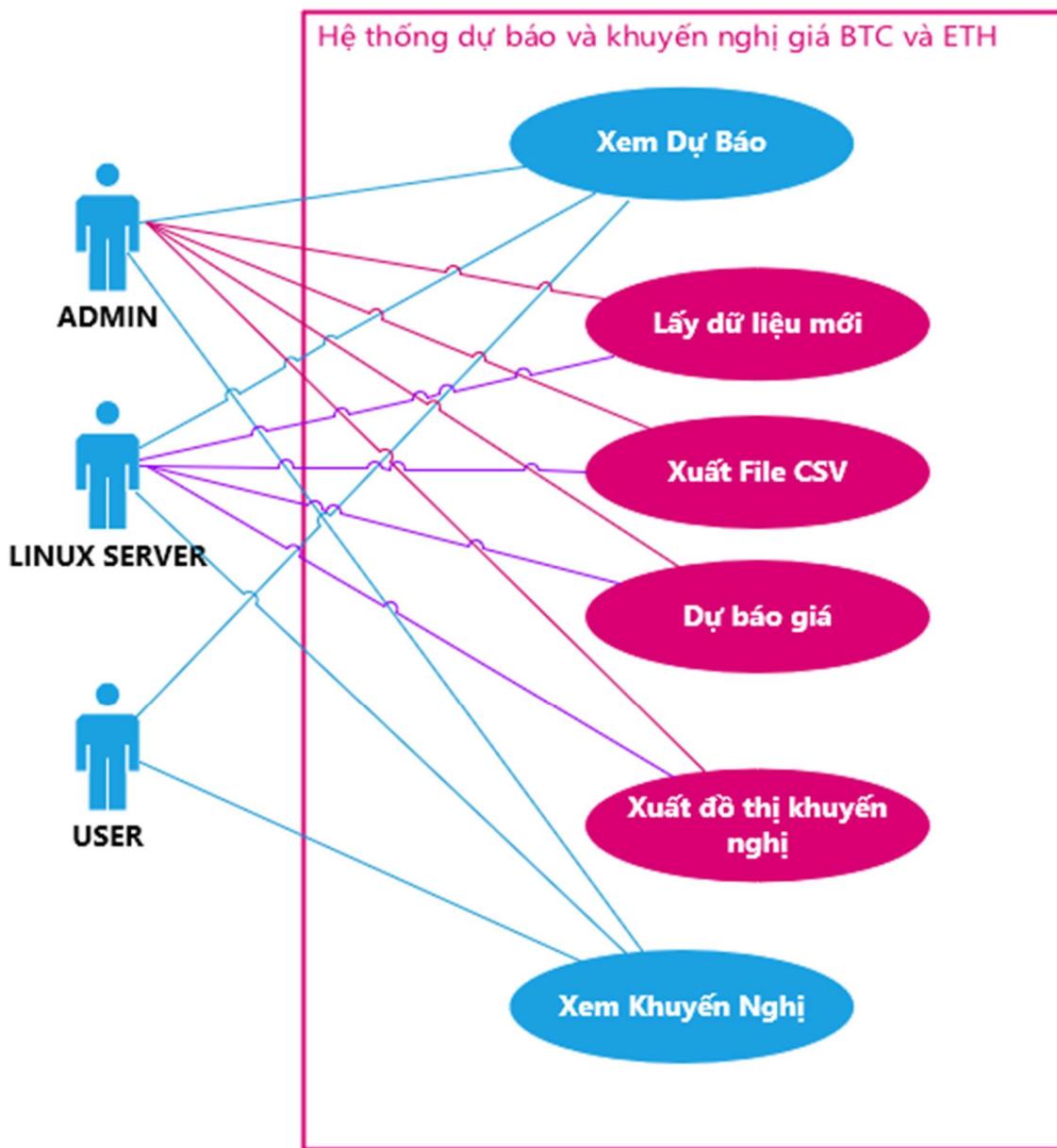
4.1.4. Mô hình khuyến nghị Q-Learning cho Pharse 2

Vì Q-Learning là ứng dụng của máy học và trí tuệ nhân tạo, vì vậy việc đánh giá của mô hình này sẽ không chính xác bằng cách học tính toán lại giá trị Q-Value và cập nhật vào Q-Table cho thấy mô hình này đã làm cho mô hình này đạt độ tin cậy nhất định vì vậy việc áp dụng Q-Learning vào mô hình khuyến nghị Pharse 2 là một điều tất yếu.

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

4.2. Pharse 2 – Quy trình xây dựng hệ thống API

4.2.1. Thiết kế thành phần chức năng



Hình 4.4 Usecase tổng quan hệ thống dự báo và khuyến nghị giá BTC và ETH
Đặc tả Usecase tổng quan

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Tên Usecase: Xem dự báo	Mã Usecase: UC01
Tiền điều kiện: Vào đúng URL Client của hệ thống ở chế độ http	
Hậu điều kiện: Nếu xem thành công thì bảng dự báo và đồ thị dự báo sẽ xuất hiện trên hệ thống	
Tác nhân chính: Admin, User, Linux Server	Mức độ quan trọng: Cao
Mô tả Usecase: Usecase cho phép người dùng xem giá trị dự báo và bảng dự báo những ngày tiếp theo và 49 ngày trước.	
Luồng sự kiện chính <ul style="list-style-type: none">1. Trỏ đúng URL đến màn hình xem dự báo2. Load dữ liệu dự báo và dữ liệu lịch sử từ API server3. Hiển thị ra màn hình kết quả4. Kết thúc	

Bảng 4.10 Usecase xem dự báo

Tên Usecase: Xem khuyến nghị	Mã Usecase: UC02
Tiền điều kiện: Vào đúng URL Client của hệ thống ở chế độ http	
Hậu điều kiện: Nếu xem thành công thì đồ thị khuyến nghị sẽ xuất hiện trên hệ thống	
Tác nhân chính: Admin, User, Linux Server	Mức độ quan trọng: Cao
Mô tả Usecase: Usecase cho phép người dùng xem hành động khuyến nghị ngày tiếp theo và 49 ngày trước.	
Luồng sự kiện chính <ul style="list-style-type: none">1. Trỏ đúng URL đến màn hình xem dự báo2. Load đồ thị khuyến nghị từ API server	

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

- | |
|---------------------------------|
| 3. Hiển thị ra màn hình kết quả |
| 4. Kết thúc |

Bảng 4.11 Usecase xem khuyến nghị

Tên Usecase: Lấy dữ liệu mới	Mã Usecase: UC03
Tiền điều kiện: Vào đúng URL Server của hệ thống. Đối với Server Linux thì server tự route đúng vào API.	
Hậu điều kiện: Dữ liệu được cập nhật vào CSDL MySQL (Vào CSDL xem kết quả)	
Tác nhân chính: Admin, Linux Server	Mức độ quan trọng: Cao
Mô tả Usecase: Lấy dữ liệu mỗi ngày từ trang web coindesk, sau đó lưu vào cơ sở dữ liệu	
Luồng sự kiện chính <ul style="list-style-type: none">1. Trỏ vào đường dẫn URL Server2. Thêm một ngày mới vào Bảng (Chức năng chỉ áp dụng với Linux Server)3. Lấy dữ liệu coindesk. Lưu dữ liệu vào cơ sở dữ liệu4. Kết thúc	

Bảng 4.12 Usecase lấy dữ liệu mới

Tên Usecase: Xuất file CSV	Mã Usecase: UC04
Tiền điều kiện: Vào đúng URL Server của hệ thống. Đối với Server Linux thì server tự route đúng vào API.	
Hậu điều kiện: Lấy dữ liệu từ database 100 dòng đầu sau đó xuất ra file CSV và lưu thu mục public của server	
Tác nhân chính: Admin, Linux Server	Mức độ quan trọng: Cao
Mô tả Usecase: Lấy từ CSDL lưu vào file CSV	

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Luồng sự kiện chính

1. Trỏ vào đường dẫn URL Server
2. Lấy dữ liệu từ CSDL
3. Lưu vào file CSV
4. Kết thúc

Bảng 4.13 Usecase xuất file CSV

Tên Usecase: Dự báo giá	Mã Usecase: UC05
Tiền điều kiện: Vào đúng URL Server của hệ thống. Đối với Server Linux thì server tự route đúng vào API.	
Hậu điều kiện: Giá trị Predict được cập nhật vào CSDL	
Tác nhân chính: Admin, Linux Server	Mức độ quan trọng: Cao
Mô tả Usecase: Lấy file CSV từ thư mục public, dùng mô hình ARIMA_LSTM học sao đó lưu lại vào CSDL.	
Luồng sự kiện chính	
<ol style="list-style-type: none">1. Trỏ vào đường dẫn URL Server2. Lấy dữ liệu từ CSV thư mục public3. Học và dự báo bằng mô hình ARIMA_LSTM học lại4. Lưu giá trị predict vào CSDL5. Kết thúc	

Bảng 4.14 Usecase Dự báo giá

Tên Usecase: Xuất đồ thị khuyến nghị	Mã Usecase: UC06
Tiền điều kiện: Vào đúng URL Server của hệ thống. Đối với Server Linux thì server tự route đúng vào API. Và phải thực hiện predict trước khi khuyến nghị để lấy giá trị dự báo ngày tiếp theo	
Hậu điều kiện: Đồ thị khuyến nghị được và lưu vào thư mục public	
Tác nhân chính: Admin, Linux Server	Mức độ quan trọng: Cao

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

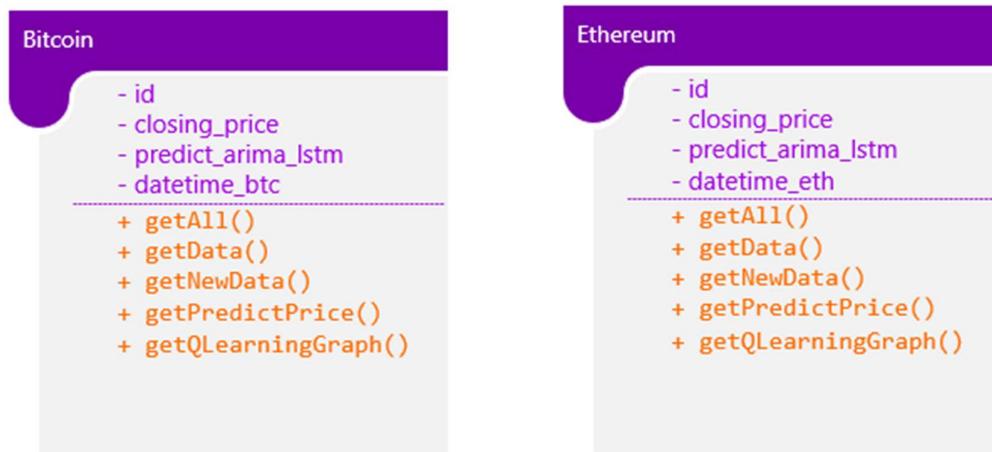
Mô tả Usecase: Lấy file CSV mới nhất từ CSDL (sau việc dự báo), học mô hình Q-Learning đưa ra kết quả từng ngày. Xuất file png vào thư mục public

Luồng sự kiện chính

1. Trỏ vào đường dẫn URL Server
2. Lấy dữ liệu từ CSV sau khi dự báo thư mục public
3. Học Q-Learning sau đó xuất ra file CSV
4. Kết thúc

Bảng 4.15 Usecase Xuất đồ thị khuyến nghị

4.2.2. Thiết kế thành phần dữ liệu



Hình 4.5 Class diagram hệ thống

Mô tả class

Bitcoin Class

Thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Ý nghĩa
id	int		ID Bitcoin một ngày
closing_price	float	null	Giá cuối ngày

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

predict_arima_lstm	float	null	Giá dự báo bằng hybrid ARIMA_LSTM
datetime_btc	Datetime	Default (Unique)	Ngày lưu trong CSDL BTC

Bảng 4.16 Bảng thuộc tính Bitcoin

Hàm	UseCase	Ý nghĩa
getAll()	UC05	Lấy dữ liệu mới
getData()	UC04	Xuất file CSV
getPredictPrice()	UC05	Dự báo giá
getQLearningGraph()	UC06	Xuất đồ thị khuyến nghị

Bảng 4.17 Bảng hàm Bitcoin Class

Ethereum Class

Thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Ý nghĩa
id	int		ID Bitcoin một ngày
closing_price	float	null	Giá cuối ngày
predict_arima_lstm	float	null	Giá dự báo bằng hybrid ARIMA_LSTM
datetime_eth	Datetime	Default (Unique)	Ngày lưu trong CSDL ETH

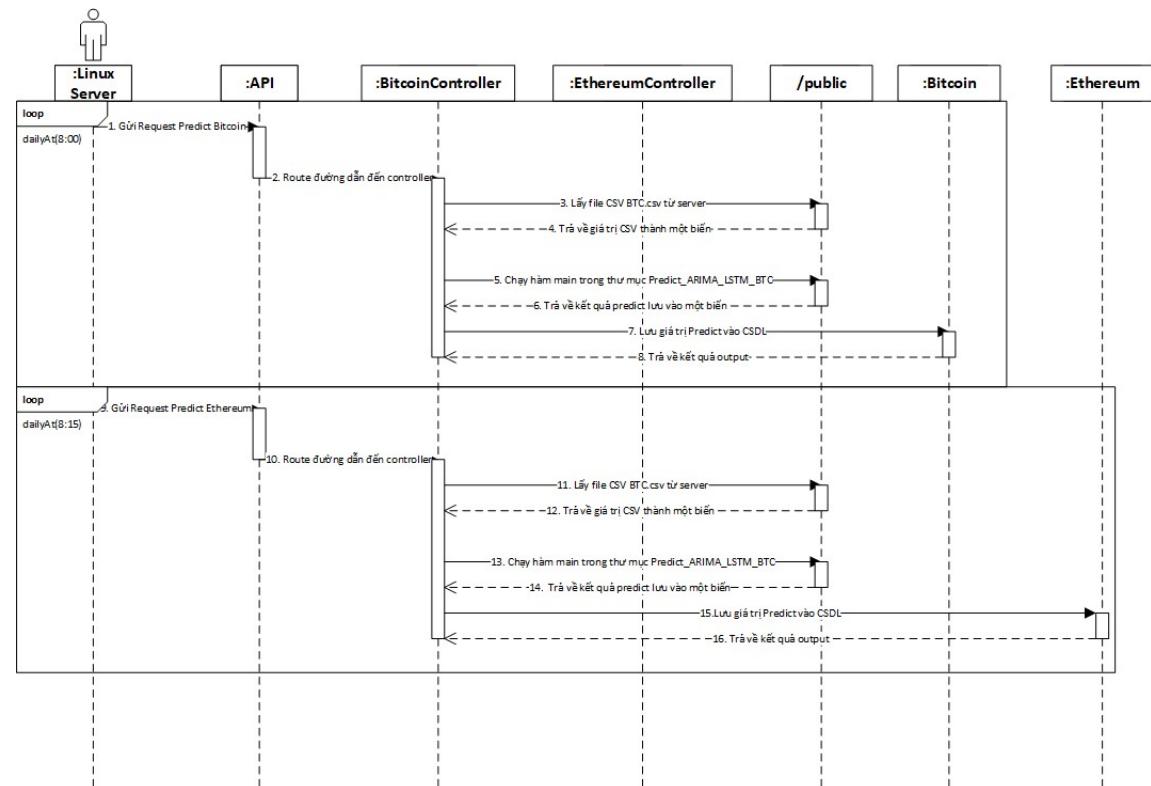
Bảng 4.18 Bảng thuộc tính Ethereum

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Hàm	Usecase	Ý nghĩa
getAll()	UC05	Lấy dữ liệu mới
getData()	UC04	Xuất file CSV
getPredictPrice()	UC05	Dự báo giá
getQLearningGraph()	UC06	Xuất đồ thị khuyến nghị

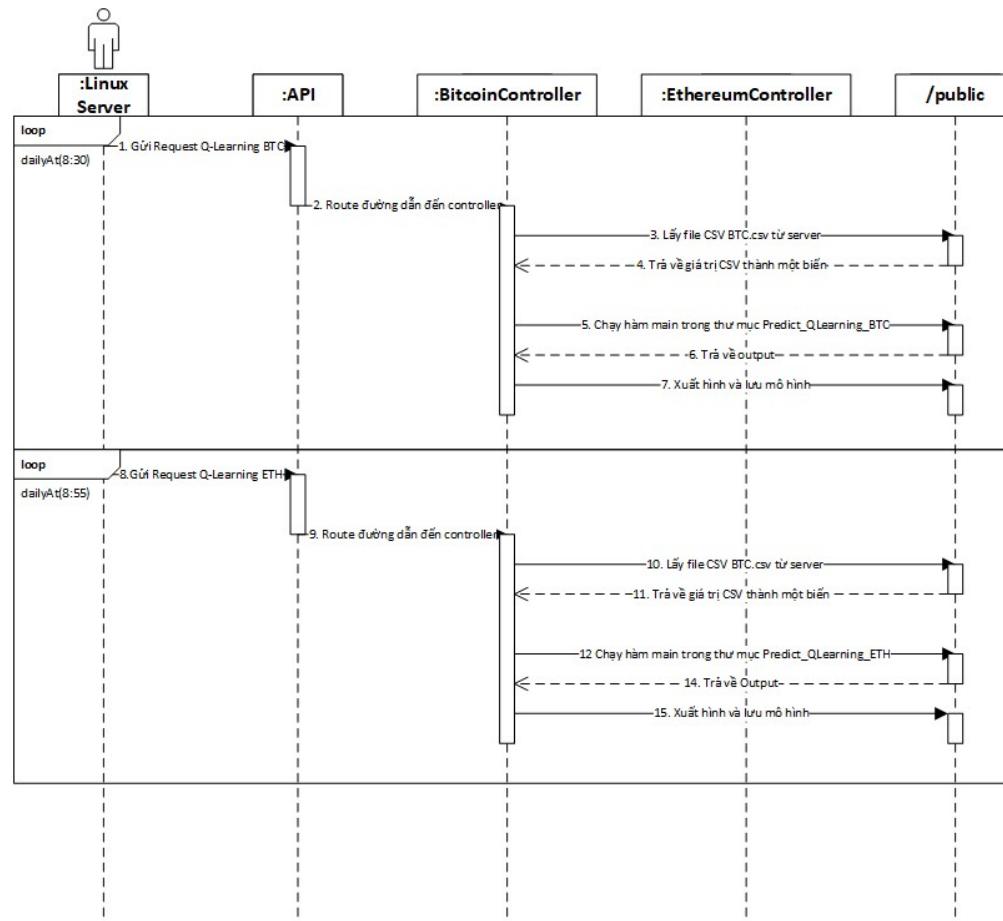
Bảng 4.19 Bảng hàm Ethereum Class

4.2.3. Thiết kế thành phần xử lý



Hình 4.6 Sequence Diagram Dự báo giá BTC và ETH

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM



Hình 4.7 Sequence Diagram khuyến nghị BTC và ETH

4.3. Pharse 3 – Triển khai hệ thống API lên Server Clould Linux

4.3.1. Cơ chế định thời trong Laravel [22]

Định nghĩa Task Scheduling trong thư mục `app\Console\Kernel.php` trong hàm `schedule`. Ví dụ minh họa xóa một tài khoản `recent_users` hàng ngày.

```
/*
protected function schedule(Schedule $schedule)
{
    $schedule->call(function () {
        DB::table('recent_users')->delete();
    })->daily();
}
```

Hình 4.8 Ví dụ minh họa về Task Scheduling

Một số hàm định thời Task Scheduling được laravel dung sẵn

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Method	Description
<code>->cron('* * * * *');</code>	Run the task on a custom cron schedule
<code>->everyMinute();</code>	Run the task every minute
<code>->everyTwoMinutes();</code>	Run the task every two minutes
<code>->everyThreeMinutes();</code>	Run the task every three minutes
<code>->everyFourMinutes();</code>	Run the task every four minutes
<code>->everyFiveMinutes();</code>	Run the task every five minutes
<code>->everyTenMinutes();</code>	Run the task every ten minutes
<code>->everyFifteenMinutes();</code>	Run the task every fifteen minutes
<code>->everyThirtyMinutes();</code>	Run the task every thirty minutes
<code>->hourly();</code>	Run the task every hour

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Method	Description
<code>->hourlyAt(17);</code>	Run the task every hour at 17 minutes past the hour
<code>->everyTwoHours();</code>	Run the task every two hours
<code>->everyThreeHours();</code>	Run the task every three hours
<code>->everyFourHours();</code>	Run the task every four hours
<code>->everySixHours();</code>	Run the task every six hours
<code>->daily();</code>	Run the task every day at midnight
<code>->dailyAt('13:00');</code>	Run the task every day at 13:00
<code>->twiceDaily(1, 13);</code>	Run the task daily at 1:00 & 13:00
<code>->weekly();</code>	Run the task every Sunday at 00:00
<code>->weeklyOn(1, '8:00');</code>	Run the task every week on Monday at 8:00

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Method	Description
<code>->monthly();</code>	Run the task on the first day of every month at 00:00
<code>->monthlyOn(4, '15:00');</code>	Run the task every month on the 4th at 15:00
<code>->twiceMonthly(1, 16, '13:00');</code>	Run the task monthly on the 1st and 16th at 13:00
<code>->lastDayOfMonth('15:00');</code>	Run the task on the last day of the month at 15:00
<code>->quarterly();</code>	Run the task on the first day of every quarter at 00:00
<code>->yearly();</code>	Run the task on the first day of every year at 00:00
<code>->yearlyOn(6, 1, '17:00');</code>	Run the task every year on June 1st at 17:00
<code>->timezone('America/New_York');</code>	Set the timezone for the task

Bảng 4.20 Danh sách các hàm schedule được dựng sẵn Laravel

Các hàm định thời schedule dựng trong hệ thống

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Vào 07:30 mỗi ngày Linux Server sẽ lấy dữ liệu mới từ website Coindesk lần các trang web liên quan

```
$schedule->call(function () {
    $ch = curl_init();
    curl_setopt($ch, option: CURLOPT_URL, value: "http://112.78.4.49/api/getNewDataBTC");
    curl_setopt($ch, option: CURLOPT_HEADER, value: 0);
    curl_exec($ch);
    curl_close($ch);

    $ch = curl_init();
    curl_setopt($ch, option: CURLOPT_URL, value: "http://112.78.4.49/api/getNewDataETH");
    curl_setopt($ch, option: CURLOPT_HEADER, value: 0);
    curl_exec($ch);
    curl_close($ch);
})->dailyAt('07:30');
```

Hình 4.9 Schedule lấy dữ liệu mới

Vào 07:40 mỗi ngày Linux Server sẽ xuất liệu ra file CSV để chuẩn bị cho quá trình học máy python

```
//function export data to csv
$schedule->call(function () {
    $ch = curl_init();
    curl_setopt($ch, option: CURLOPT_URL, value: "http://112.78.4.49/api/getAllBTC");
    curl_setopt($ch, option: CURLOPT_HEADER, value: 0);
    curl_exec($ch);
    curl_close($ch);

    $ch = curl_init();
    curl_setopt($ch, option: CURLOPT_URL, value: "http://112.78.4.49/api/getAllETH");
    curl_setopt($ch, option: CURLOPT_HEADER, value: 0);
    curl_exec($ch);
    curl_close($ch);
})->dailyAt('07:40');
```

Hình 4.10 Schedule lấy dữ liệu mới nhất xuất thành CSV

4.3.2. Cơ chế CRON trong Server Linux

Thiết lập cơ CRON trong linux như sau để mỗi phút Linux Server đều hoạt động như sau:

Trong server Linux gõ lệnh: **crontab -e**

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Thêm file cấu hình cron để server luôn chạy: * * * * * cd

```
/var/www/KLTN_Minhhut/ProjectLaravel && php artisan schedule:run  
>> /dev/null 2>&1
```

```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
* * * * * cd /var/www/KLTN_Minhhut/ProjectLaravel && php artisan schedule:run >> /dev/null 2>&1
```

Hình 4.11 File cấu hình crontab

Lưu file crontab

Kiểm tra Crontab đã hoạt động hay không gõ lệnh **crontab -l**

```
root@CloudServer_4d49:/var/www/KLTN_Minhhut# crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
* * * * * cd /var/www/KLTN_Minhhut/ProjectLaravel && php artisan schedule:run >> /dev/null 2>&1
```

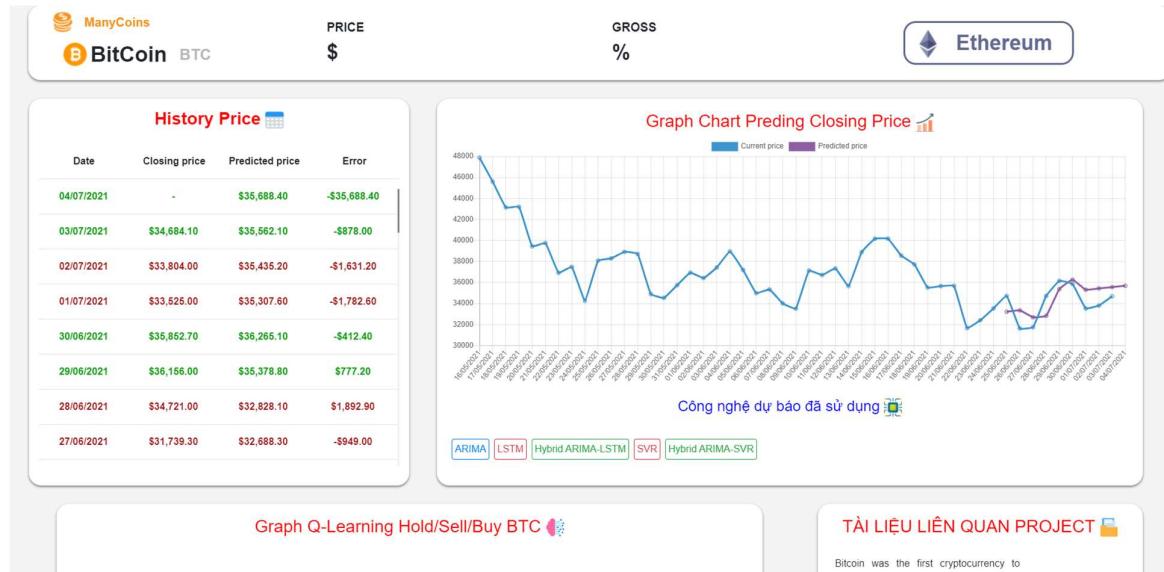
Hình 4.12 Danh sách các crontab đang chạy trong server Linux

Kiểm tra log Crontab: grep CRON /var/log/syslog

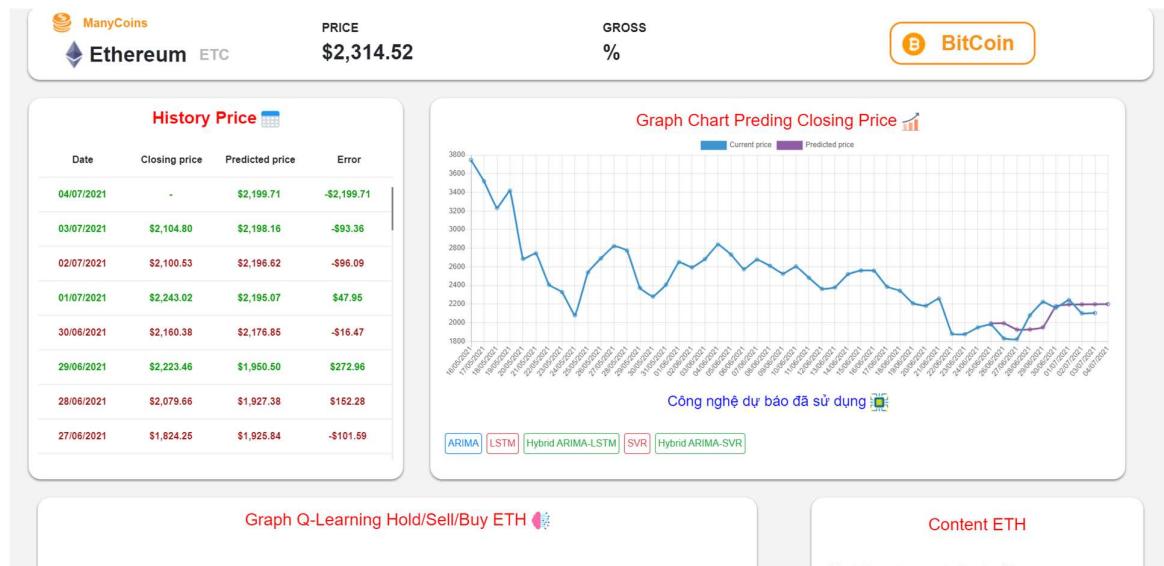
PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

4.4. Pharse 4 – Xây dựng giao diện và giao tiếp API

Màn hình tổng quan trang web



Hình 4.13 Màn hình tổng quan Bitcoin Page



Hình 4.14 Màn hình tổng quan Ethereum

Màn hình xem lịch sử giá dạng bảng và xem kết quả giá dự báo

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

History Price 📅			
Date	Closing price	Predicted price	Error
04/07/2021	-	\$2,199.71	-\$2,199.71
03/07/2021	\$2,104.80	\$2,198.16	-\$93.36
02/07/2021	\$2,100.53	\$2,196.62	-\$96.09
01/07/2021	\$2,243.02	\$2,195.07	\$47.95
30/06/2021	\$2,160.38	\$2,176.85	-\$16.47
29/06/2021	\$2,223.46	\$1,950.50	\$272.96
28/06/2021	\$2,079.66	\$1,927.38	\$152.28
27/06/2021	\$1,824.25	\$1,925.84	-\$101.59

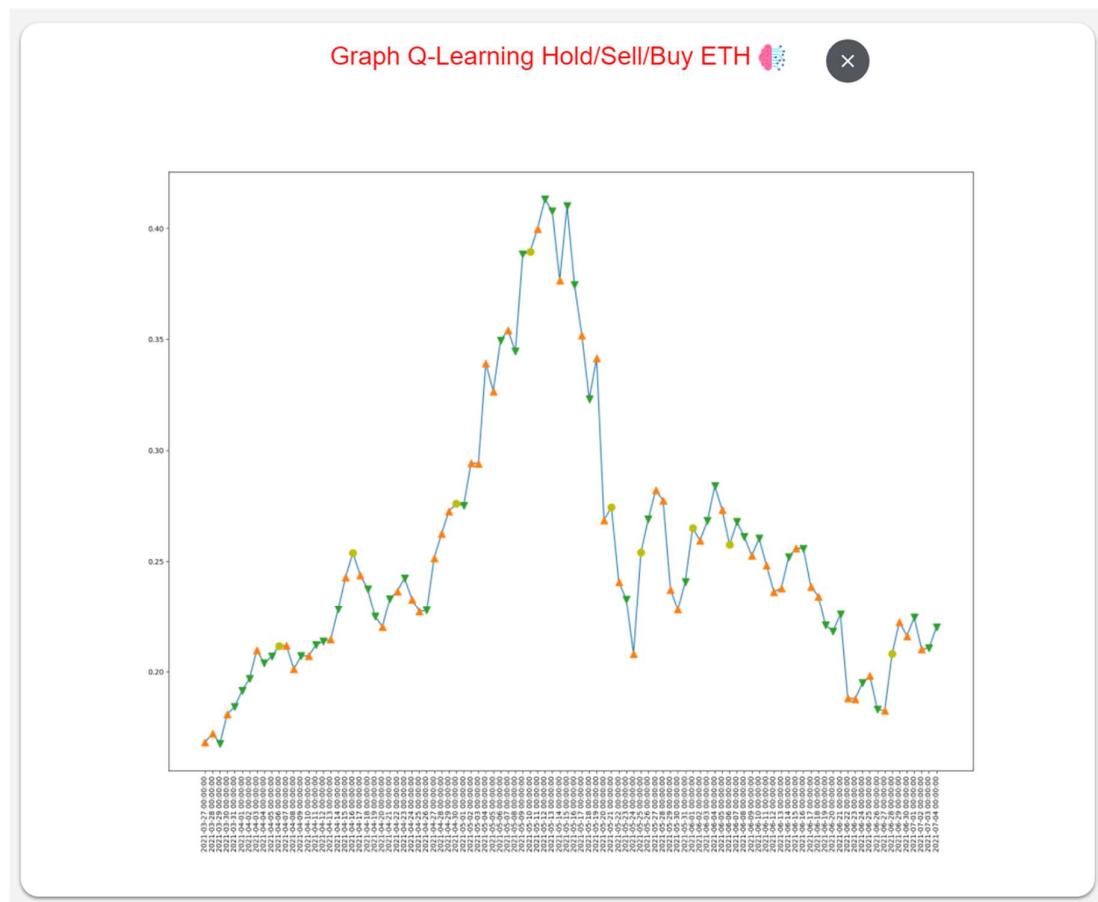
Hình 4.15 Màn hình xem lịch sử giá dạng bảng
Biểu đồ dự báo giá dạng chart



Hình 4.16 Màn hình xem lịch sử giá dạng biểu đồ

PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

Biểu đồ khuyến nghị hành động dạng chart



Hình 4.17 Màn hình khuyến nghị hành động

Ghi chú

1. Tam giác màu cam là hành động mua
2. Tam giác màu xanh lá là hành động bán
3. Hình tròn xanh mạ là hành động giữ.

KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

Chương 5. KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

Kiến thức đạt được trong khi quá trình thực hiện khóa luận tốt nghiệp:

- Tìm hiểu chuỗi thời gian và bản chất của chuỗi thời gian
- Tìm hiểu được mô hình ARIMA
- Hiểu được cơ chế của thuật giải SVR và mô hình LSTM.
- Biết được việc kết hợp ARIMA_SVR, ARIMA_LSTM
- Đánh giá mô hình dự báo
- Biết được kiến thức Q-Learning
- Xây dựng được các chức năng phục vụ cho quá trình xây dựng hệ thống dự báo và khuyến nghị.
- Hiểu được cơ chế quy trình xây dựng một website từ lúc phân tích đến lúc đưa lên môi trường production
- Khả năng nghiên cứu tài liệu
- Khả năng quản lý source code

Hạn chế:

- Có nhiều công nghệ dự báo mới tốt hơn chưa được tìm hiểu
- Mô hình dự báo sẽ không tốt với những trường hợp tăng giảm đột ngột
- Thuật toán chưa tối ưu nhất do chưa biết điều chỉnh phù hợp nhất

5.2. Hướng phát triển

Tìm hiểu thêm nhiều mô hình mới kết hợp nhiều thuật toán mới để mô hình cho kết quả tốt nhất.

Xây dựng thêm nhiều chức năng khuyến nghị hơn như là khuyến nghị dựa trên mô hình nén Nhật⁸.

⁸ <https://www.lotforex.vn/cac-mo-hinh-nen-nhat/>

KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

Xây dựng ứng dụng trên nhiều đơn vị đồng tiền, nghiên cứu thêm nhiều đối tượng dự báo liên quan chuỗi thời gian liên quan đến các vấn đề: Kinh tế, văn hóa, xã hội, du lịch,....

Ngoài việc dự báo, khuyến nghị dựa trên giá việc nghiên cứu đồng tiền BTC, ETH ảnh hưởng đến các chỉ báo khác như giá xăng, giá dầu,... Và cải tiến mô hình dự báo dựa trên những yếu tố này.

TÀI LIỆU THAM KHẢO

Chương 6. TÀI LIỆU THAM KHẢO

- [1] TS. Nguyễn Thị Minh An, “Khái niệm dự báo”. [Trực tuyến]. Địa chỉ: <http://quantri.vn/dict/details/9174-khai-niem-du-bao>. [Truy cập 03/7/2021]
- [2] Minh Anh, “Dự báo là gì? Các phương pháp dự báo”, 2018. [Trực tuyến]. Địa chỉ: <https://vietnamfinance.vn/du-bao-la-gi-cac-phuong-phap-du-bao-20180504224208609.htm>. [Truy cập: 03/7/2021]
- [3] S'Mann, “Ethereum là gì? Thông tin về Ethereum cho người mới”, 2018. [Trực tuyến]. Địa chỉ: <https://syndicator.vn/ethereum-la-gi.html>. [Truy cập: 03/7/2021]
- [4] Xiaobing Yang, Qionghong Duan, Jianjie Wang, Zhengbin Zhang, “Seasonal Variation of Newly Notified Pulmonary Tuberculosis Cases from 2004 to 2013 in Wuhan, China”, 2014
- [5] Rob J Hyndman, “Cyclic and seasonal time series”, 2011. [Trực tuyến]. Địa chỉ: <https://robjhyndman.com/hyndtsight/cyclists/>. [Truy cập 03/7/2021]
- [6] Alex Mitrani, “Achieving Stationarity With Time Series Data”, 2020
- [7] ritvikmath, “Time Series Talk : Augmented Dickey Fuller Test + Code”, 2020. [Trực tuyến]. Địa chỉ: https://www.youtube.com/watch?v=1opjnegr_hA&t=393s. [Truy cập 04/7/2021]
- [8] “Bài 19: Support Vector Machine”, 2017. [Trực tuyến]. Địa chỉ: <https://machinelearningcoban.com/2017/04/09/sm/>. [Truy cập 04/7/2021]
- [9] “Bài 21: Kernel Support Vector Machine”, 2017. [Trực tuyến]. Địa chỉ: <https://machinelearningcoban.com/2017/04/22/kernelsmv/>. [Truy cập 04/7/2021]
- [10] hanhbd, “Neural Network”, 2020. [Trực tuyến]. Địa chỉ: <https://kipalog.com/posts/Neural-Network>. [Truy cập 04/7/2021]
- [11] Dương Thu Trang, “ỨNG DỤNG MẠNG NƠRON NHÂN TẠO DỰ BÁO SỐ HỌC SINH TUYỀN VÀO TRUNG TÂM GDNN – GDTX QUẬN ĐỐNG ĐA”, 2011.

TÀI LIỆU THAM KHẢO

- [12] Nttuan8, “Bài 3: Neural network”, 2019. [Trực tuyến]. Địa chỉ: <https://nttuan8.com/bai-3-neural-network/>. [Truy cập 04/7/2021]
- [13] “Neural Network Toolbox hardlim”. [Trực tuyến]. Địa chỉ: <https://www.cs.montana.edu/courses/spring2005/530/help/nnhelp/hardlim.html>. [Truy cập 04/7/2021]
- [14] “Artificial Neural Network - Building Blocks”, [Trực tuyến]. Địa chỉ: https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_building_blocks.htm. [Truy cập 04/7/2021]
- [15] Huy Truong Minh, “SỬ DỤNG MÔ HÌNH MẠNG THẦN KINH NHÂN TẠO ĐỂ ĐƯỜNG BÁO GIÁ CHUNG KHOÁN”, [Trực tuyến]. Địa chỉ: <https://www.chiemtinhtaichinh.com/2019/12/20/su-dung-mo-hinh-mang-kinh-nhan-tao-de-du-bao-gia-chung-khoan.html>. [Truy cập 04/7/2021]
- [16] “Understanding LSTM Networks”, 2015, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Truy cập 20/6/2021]
- [17] Professor Dr. W. Brauer, Dr. Jurgen Schmidhuber, “Diplomarbeit Im Fach Informatik”, 1991.
- [18] Shi Yan, “Understanding LSTM and its diagrams” , 2016, <https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714>
- [19] Ketan Doshi, “Reinforcement Learning Explained Visually (Part 4): Q Learning, step-by-step”, 2020.
- [20] Yu-Sheng Kao, Kazumitsu Nawata, Chi-Yo Huang, “Predicting Primary Energy Consumption Using Hybrid ARIMA and GA-SVR Based on EEMD Decomposition”, 2020
- [21] Peter G. Zhang, “Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. Neurocomputing 50, 159-175”, 2003.

TÀI LIỆU THAM KHẢO

[22] “Task Scheduling”, 2021, [Trực tuyến]. Địa chỉ:
<https://laravel.com/docs/8.x/scheduling>, [Truy cập 04/7/2021]

Predicting the closing price of Cryptocurrency using hybrid ARIMA, regression and machine learning

¹Dinh Thuan Nguyen, ²Minh Nhut Nguyen

^{1,2}University of Information Technology, VNU-HCM, Ho Chi Minh, Vietnam

thuannd@uit.edu.vn, 17520867@gm.uit.edu.vn

ABSTRACT

The Cryptocurrency is growing strongly and widely such as Bitcoin (BTC) and Ethereum (ETH) used in the world, which has attracted wide attention of researchers in recent times. In this work, Autoregressive Integrate Moving Average (ARIMA) model, machine learning algorithms, Support Vector Regression (SVR) will be implemented to predict the closing price of The Cryptocurrency the next day. After that, we present hybrid methods between ARIMA model, machine learning and regression to improve the forecasting of Cryptocurrency price. After experimenting, researching and giving the results of the improvement of the hybrid-model through evaluation by RMSE and MAPE.

Keywords: BTC prediction, ETH prediction, ARIMA, machine learning, SVR, Hybrid model.

INTRODUCTION

In the report about investigation “Cryptocurrency” [1] by Jake Frankenfield. A cryptocurrency is a digital or virtual currency that is secured by cryptography, which makes it nearly impossible to counterfeit or double-spend. Many cryptocurrencies are decentralized networks based on blockchain technology.

In October 2008, Bitcoin was firstly introduced by Satoshi Nakamoto in the report "Bitcoin: A Peer-to-Peer Electronic Cash System" [2]. In 2009, Nakamoto has released software that created

Bitcoin and had a large community using Bitcoin around the world so far. The biggest advantage of Bitcoin is controlled by blockchain technology that help to transparent transactions. As a cryptocurrency, Ethereum is the second largest cryptocurrency after bitcoin. The application of the ETH coin is the remaining decentralized application "dApps". The difference between ETH and BTC is that it works through a smart contract.

In recent years, Cryptocurrencies develop strongly, in June 2021 the price of 1 BTC coin is equivalent to \$34,040.92, and 1 ETH coin is equivalent to \$2,097.96. Therefore, researching and forecasting cryptocurrency prices is a hot topic.

In the investigation and study of the predictive model, we will perform the first phase of each model individually: ARIMA, LSTM, SVR and then give the results. In phase 2, we combine ARIMA with LSTM to produce ARIMA-LSTM hybrid model and ARIMA combine SVR model to produce ARIMA-SVR model. Then evaluate the effectiveness of the hybrid model.

RELATED WORKS

Poongodi Manoharan [3] applied an ARIMA, LSTM, SVM model to predict the next day's Bitcoins based on the closing price of Bitcoin in the previous. ARIMA models, LSTM Model, and SVR model are installed and experimented. The prediction accuracy is about 49%.

Saad Alahmari [4] used Support Vector Machine with SVR model with Linear, Polynomial and “Radial Basis Function”(RBF) kernels to predict the price of the next day's three cryptocurrencies Bitcoin (BTC), Ethereum (ETH), XRP. The results of the study were quite high. e.g with the largest data dataset of Bitcoin with $R^2 = 78\%$ and with the smallest data dataset with $R^2 = 56\%$.

Ioannis E. Livieris [5] applied an CNN-LSTM model for Cryptocurrency Forecasting. The experiment shows that using LSTM model will achieve higher results than traditional statistical models ARIMA. In addition the paper also gives the training test results of different selection data giving different results and comparing them with each other. Most of the models above use an independent method such as statistics, regression, and machine learning to make future cryptocurrency price forecasts. Our work will replicate the above cases using the hybrid model

ARIMA-LSTM and ARIMA-SVR to forecast the next days and compare the results with the isolated models.

THE PROPOSED METHOD OR PERHAPS METHODOLOGY

ARIMA model (Autoregressive Integrated Moving Average)

ARIMA Model [6] is a combination of autoregressive model (AR) and moving average model (MA) with Integrated (I) in stationary time series. An ARIMA model (p, d, q) includes:

- AR (Autoregression): It is a regression process based on the dependence of the observed sample's value with number lagged observations sample and denoted by “p”.
- I (Integrated): It is the process of taking the difference. Most trending time series do not count. Therefore, taking the difference is an important thing to be concern with. To form a stationary series, the simplest method is to take the difference and “d” represents the process of taking the difference of order d.
- MA (Moving Average): This is a regression process based on the dependence of the sample value on the observed residual errors with the observed sample with numerical lag and denoted by “q”.

They are defined as follows:

$$AR(p) = y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \varepsilon_t$$

Where y_t is the stationary value, c is the constant value or in other words the degrees of freedom, ϕ_i the regression coefficient of the time corresponding to lag 1, 2 , 3, ... p and ε_t is the residual error at time t. And MA(q) can written by as follows:

$$MA(q) = y_t = \mu + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

Where y_t is the stationary value, μ is the constant value of residual error or in other words the degrees of freedom, θ_i the regression coefficient of residual error of the time corresponding to lag 1, 2 , 3, ... q and ε_t is the residual error at time t.

So, the combined model of AR(p) and MA(q) models is represented as follows:

$$y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \mu + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

with $\phi_i \neq 0$, $\theta_i \neq 0$ and $\sigma_\varepsilon^2 > 0$.

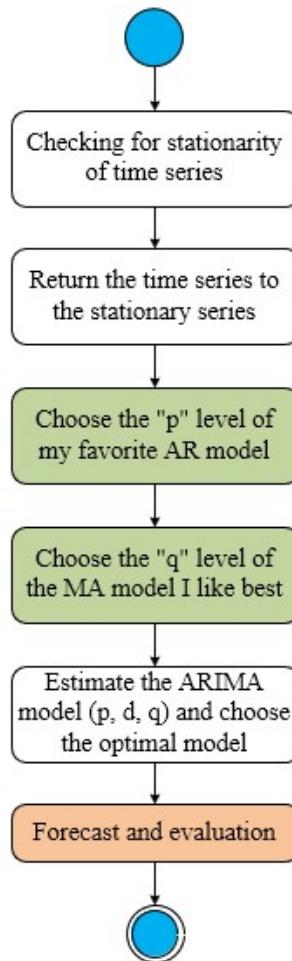
As the concept of the "integrated" process presented, it is the process of taking the difference with a difference of order d, e.g $I(1) = x_t - x_{t-1}$. And $I(d)$ written by as follows:

$$I(d) = D(D^{d-1}(y_t))$$

The ARIMA process is carried out according to the Box–Jenkins method as follows:

Figure 1

Box-Jenkins Methods applied ARIMA process.

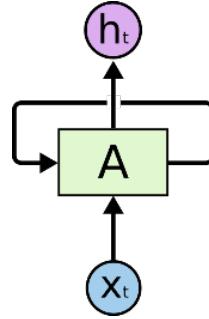


LSTM model (Long Short Term Memory)

That was the traditional neural network's inability to remember previous periods which causes its drawback, with time series models the retention of memory was vital to the values at the previous point in time to find the pattern of the time series. Since then, a recurrent neural network (RNN) was born to solve this problem. The architecture of a simple RNN is shown as follows:

Figure 2

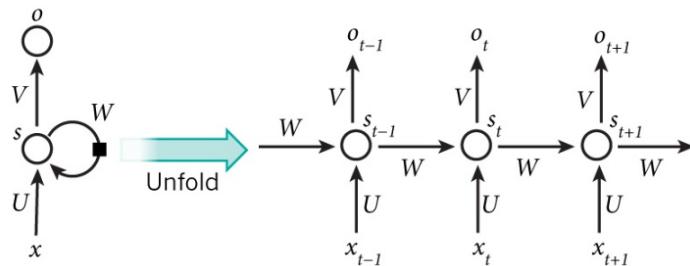
Recurrent Neural Networks have loops. [7]



With The detail architecture of a simple RNN is shown as follows:

Figure 3

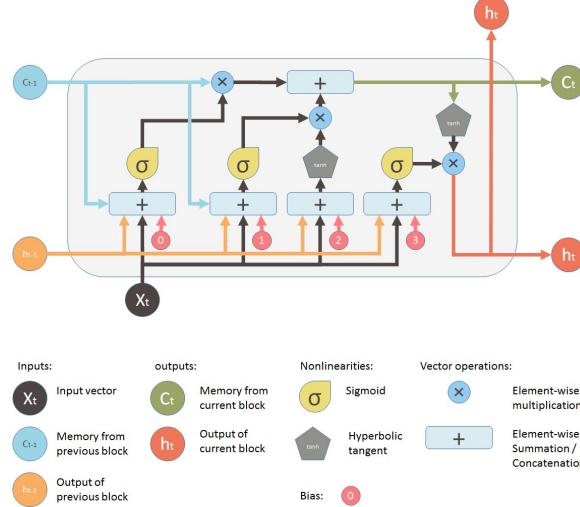
A recurrent neural network and the unfolding in time of the computation involved in its forward computation. [8]



Natural RNNs cannot be learned over long periods of time. Theoretically, RNN can solve the problem of long-term dependencies by adding parameters, but RNN cannot learn these parameters through research [9]. Fortunately, the LSTM network is a special kind of RNN and solves the issue of long-term dependencies. A LSTM block [10] includes indicators:

Figure 4.

LSTM block [9]



- Forget Gate: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
- At this gate, if the Output value is 1, it means keep all information, and 0 indicate that all information will be discarded.
- Input Gate : $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ The input value gate of the LSTM block consists of the current and the previous value.
- Tanh layer with input gate: $\tilde{C} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
- Cell State for next input: $C_t = f_t * C_{t-1} + i_t * \tilde{C}$
- Output Gate: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ The output value gate of the LSTM block consists of the current and the previous value.
- New State: $h_t = o_t * \tanh(C_t)$

The LSTM model is suitable for the time series model because it explores the relationship between law of the time series. Because the LSTM model is suitable for sequential models such as time series models.

SVR model (Support Vector Regression)

SVR [11] is a regression model that is a module of SVM. It explores the relationship of variables

X and Y to find a boundary line dividing the two classes where that boundary line is the most optimal.

Assume that the training set data pairs are represented as follows:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

Where $x_i \in \mathbb{R}^d$ represents d instance of input data, y_i represents label or output data. And model find Y value can be written with function as follows:

$$Y_t = f(X_t) = f(x_1 w_1 + x_2 w_2 + \dots + x_t w_t + b)$$

Our goal is to find the optimal w and b so that the margin of the forecast function reaches a minimum value.

$$(w, b) = \arg \max_{(w, b)} \left\{ \min_n \frac{y_n (w^T x_n + b)}{\sqrt{\sum_{i=1}^T w_i^2}} \right\} = \arg \max_{(w, b)} \left\{ \frac{1}{\sqrt{\sum_{i=1}^T w_i^2}} \min_n y_n (w^T x_n + b) \right\}$$

Results of other authors and Method theory of hybrid model

Monghwan Seo [12] applied hybrid model to forecast Primary Energy Consumption. The results of the combined model compared with the independent use of the figure gave positive results with the RMSE of the hybrid model being 2.13 while the independent use of ARIMA 2.318 showed that the hybrid model had less error.

Emmanuel Dave [13] conducted experiments on the use of the ARIMA single model, the use of the single model, and the use of the hybrid model. Using the ARIMA_LSTM hybrid model gives better RMSE and MAPE results than either individual model.

Seeing the effectiveness of combining two models to achieve better results, we researched the combination of ARIMA_SVR and ARIMA_LSTM models with cryptocurrencies to forecast prices for the next day.

Peter G. Zhang [13] proposed the combination of ARIMA and ANN models for time series forecasting. The main idea of this proposal is based on linear components (ARIMA) and nonlinear components (ANN). The forecast model is shown below:

$$Y_t = L_t + N_t$$

Where Y_t is the value in the time series. L_t is the linear component and N_t is the nonlinear component.

First, we use the ARIMA model to predict the linear components and the non-linear components. We call them the error residual, denoted e_t . It is calculated as follows:

$$e_t = Y_t - \hat{L}_t$$

Where \hat{L}_t which is the predictive value of the ARIMA model at time t.

We have the residuals of the model obtained with n nodes in the neural network represented with the function f being a nonlinear function as follows:

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t$$

Finally, we combine ARIMA linear model and ANN nonlinear model to give the final forecast results as follows:

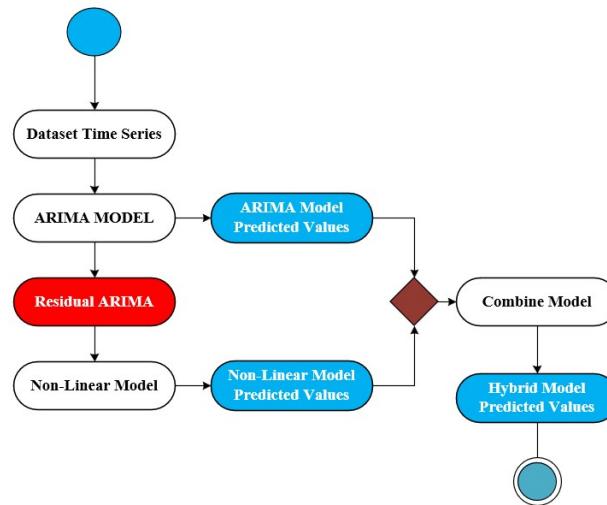
$$\hat{Y}_t = \hat{L}_t + \hat{N}_t$$

Where \hat{N}_t which is the predictive value of the ANN model at time t.

With nonlinear component not only ANN but we can also apply with LSTM or SVR.

Figure 5.

Hybrid model architecture between linear model and nonlinear model



Step 1: Select the time series data on the prices of cryptocurrencies and data processing.

Step 2: Implemented on the ARIMA model. Give residual error.

Step 2.1. Check stationary of time series data.

Step 2.2 Transform time series to stationary series.

Step 2.3 Choose the following p, q, d for the most optimal ARIMA model.

Step 2.4 Give result of predicted

Step 3: Get the error part for using machine learning or regression model then make forecast.

Step 4: Combine the results of the non-linear and linear forecasting models together.

Step 5: Give the results and evaluate the model.

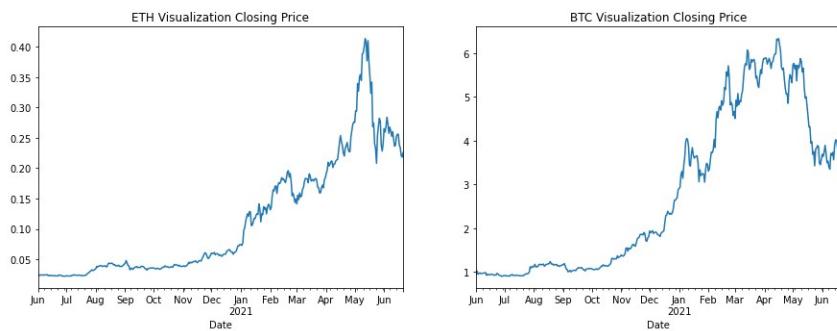
ANALYSIS

Closing price of Cryptocurrencies in Coindesk

We collect from data source CoinDesk¹, we take data from June 2020 to June 21, 2021 on two main coins and has 385 rows per coin, Bitcoin (BTC), and Ethereum (ETH). The object that we forecast is the Closing Price with the unit of 10000 USD. We will visualize the charts of these two coins in line chart form.

Figure 6.

The Closing Price ETH and BTC chart



¹ <https://www.coindesk.com/>

Split train test and handle stationary

We split the initial data into two parts with the train part we take the first 80% of the data, and the last 20% counterpart, which used for testing. We then test the stationarity of the time series using the extended Dickey and Fuller test.

If the p-value of the test is less than 0.05, we conclude that the series is stationary and proceed to start forecasting for the ARIMA model, and vice versa, we must return to the stationary series by calculating the difference between the two time points. Adjacent to each other to return to a stationary series:

$$diff_t = y_t - y_{t-1}$$

Evaluation time series forecasting models

To evaluate the predictive model of our forecasting model, we use 2 criterias as follows:

- Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{\sum_i^n \left(\frac{|y_i - f_i|}{y_i} \right)}{n}$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\sum_i^n (f_i - y_i)^2}{n}}$$

Where **n** is the population of the sample, y_i is the actual value at time i, f_i is the predicted value at time i.

RESULT

Experiment with individual models: For each individual model we use automatic ARIMA by pyramid² library which will automatically derive the matching ARIMA model. For SVR we use 'rbf' kernel, and lastly LSTM we use 'relu' activation function.

² https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html

Table 1.*Error of individual models with BTC Closing Price*

Model	RMSE	MAPE (%)
ARIMA	2.075	42.110
SVR	0.691	13.830
LSTM	0.231	4.178

Table 2.*Error of individual models with ETH Closing Price*

Model	RMSE	MAPE (%)
ARIMA	0.071	14.736
SVR	0.164	55.437
LSTM	0.030	7.989

Next, we will give the results of combining ARIMA model with SVR and LSTM. With SVR we use ‘rbf’ kernel and LSTM we use ‘relu’ activation function.

Table 3.*Error of hybrid models with BTC Closing Price*

Model	RMSE	MAPE (%)
ARIMA_SVR	0.275	4.644
ARIMA_LSTM	0.322	5.597

Table 4.*Error of hybrid models with ETH Closing Price*

Model	RMSE	MAPE (%)
ARIMA_SVR	0.063	23.224
ARIMA_LSTM	0.024	6.356

The end result with BTC coin, it is better to use ARIMA_LSTM individually for this coin,

because the ARIMA model itself gives a bad regression result, so ARIMA_LSTM is not as good as the individual LSTM . As for the ETH coin, the use of the hybrid model gives better results than the individual models.

CONCLUSION

Predicting the price of cryptocurrencies is a difficult job, requiring extensive and in-depth research on the cryptocurrency market. In addition, it also needs the support of machine learning and statistical modeling. in forecasting the next day's price of this currency. The combination of the model or the hybrid model leads to the predictive model with more effective quality as higher results, low error rate in detail.

REFERENCES

1. Jake Frankenfield, “Cryptocurrency”, 2021.
2. Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2008.
3. Poongodi Manoharan, Vignesh Vijayakumar, Naveen Chilamkurti, “Bitcoin price prediction using ARIMA model”, 2020.
4. Saad Alahmari, “Predicting the Price of Cryptocurrency using Support Vector Regression Methods”, 2020
5. Ioannis E. Livieris, Niki Kiriakidou, Stavros Stavroyiannis, Panagiotis Pintelas, “An Advanced CNN-LSTM Model for Cryptocurrency Forecasting”, 2021
6. Robert Nau, “Introduction to ARIMA models”, 2014
7. “Understanding LSTM Networks”, 2015, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> [Accessed 6/20/2021]
8. Denny Britz, “Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs”, 2015, <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/> [Accessed 6/20/2021]
9. Professor Dr. W. Brauer, Dr. Jurgen Schmidhuber, “Diplomarbeit Im Fach Informatik”, 1991

10. Shi Yan, “Understanding LSTM and its diagrams” , 2016,
<https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714>
11. [Accessed 6/20/2021]
12. Debasish Basak, Srimanta Pal, Dipak Chandra Patranabis, “Support Vector Regression”,
2007
13. Yu-Sheng Kao, Kazumitsu Nawata, Chi-Yo Huang, “Predicting Primary Energy Consumption Using Hybrid ARIMA and GA-SVR Based on EEMD Decomposition”,
2020
14. Peter G. Zhang, “Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. Neurocomputing 50, 159-175”, 2003.