Dự đoán giá cổ phiếu dựa trên các mô hình thống kê và học máy

1st Nguyễn Hữu Thắng

IS403.N21

Trường đại học Công Nghệ Thông Tin

[20520759@gm.uit.edu.vn](mailto:20520759@gm.uit.edu.vn)

2nd Trần Thu Thảo

IS403.N21

Trường đại học Công Nghệ Thông Tin

[20520769@gm.uit.edu.vn](mailto:20520769@gm.uit.edu.vn)

3rd Võ Thị Hà Trang

IS403.N21

Trường đại học Công Nghệ Thông Tin

[20522043@gm.uit.edu.vn](mailto:20522043@gm.uit.edu.vn)

4th Võ Phạm Thùy Nhung

IS403.N21

Trường đại học Công Nghệ Thông Tin

[20520679@gm.uit.edu.vn](mailto:20520679@gm.uit.edu.vn)

5th Nguyễn Đình Trải

IS403.N21

Trường đại học Công Nghệ Thông Tin

[19522371@gm.uit.edu.vn](mailto:19522371@gm.uit.edu.vn)

*Tóm tắt* — Bài báo tập trung vào đánh giá hiệu quả của các phương pháp dự đoán giá cổ phiếu sử dụng các thuật toán học máy như ARIMA, Holt-Winters, TDNN, BVAR, NNAR, AAE, Linear Regression, RNN, LSTM và GRU. Kết quả đánh giá hiệu quả của các phương pháp này cung cấp thông tin quan trọng về độ chính xác và đáng tin cậy của dự đoán giá cổ phiếu. Hiểu rõ được hiệu quả của các thuật toán học máy trong dự đoán giá cổ phiếu có thể hỗ trợ nhà đầu tư và các chuyên gia tài chính trong việc đưa ra quyết định đầu tư thông minh và quản lý rủi ro.

Từ khóa — dự đoán giá cổ phiếu, thuật toán học máy, đầu tư, ARIMA, Holt-Winters, TDNN, BVAR, NNAR, AAE, Linear Regression, RNN, LSTM và GRU.

# Giới thiệu

Trong lĩnh vực đầu tư và quản lý tài chính, dự đoán giá cổ phiếu đóng vai trò quan trọng trong việc hỗ trợ quyết định đầu tư và quản lý rủi ro. Các thuật toán học máy đã phát triển mạnh mẽ và trở thành công cụ quan trọng trong việc dự đoán giá cổ phiếu. Điều này mang lại nhiều lợi ích và tiềm năng trong việc quyết định đầu tư, nhưng cần có sự so sánh và đánh giá giữa các phương pháp khác nhau để tìm ra thuật toán phù hợp nhất.

Trong nghiên cứu này, chúng tôi đánh giá hiệu quả của các thuật toán dự đoán giá cổ phiếu bao gồm Linear Regression, ARIMA, Holt-Winters, TDNN, BVAR, NNAR, AAE, RNN, LSTM và GRU. Mục tiêu là sử dụng các chỉ số sai số như MAE, MSE và RMSE để đánh giá hiệu quả của từng phương pháp. Kết quả đánh giá sẽ giúp chúng ta hiểu rõ hơn về khả năng và giới hạn của các thuật toán học máy trong dự đoán giá cổ phiếu, từ đó xác định thuật toán tốt nhất và phù hợp nhất với mục tiêu đầu tư và quản lý rủi ro.

Nghiên cứu này giúp cải thiện khả năng tối ưu hóa lợi nhuận và quản lý rủi ro trong lĩnh vực đầu tư chứng khoán, đồng thời hỗ trợ quyết định đầu tư thông minh và đạt được hiệu suất cao trong môi trường thị trường tài chính.

# Nghiên cứu liên quan

Trong phần này, chúng ta sẽ thảo luận về các nghiên cứu trước đó đã tìm hiểu về việc dự đoán giá cổ phiếu (hoặc các nghiên cứu khác) bằng cách sử dụng các thuật toán học máy hồi quy và học sâu. Girish K. Jha và Kanchan Sinha[1] đưa ra kết quả nghiên cứu về TDNN, khẳng định này có khả năng dự báo giá tốt hơn mô hình ARIMA trong trường hợp có mối quan hệ phi tuyến trong chuỗi giá, đồng thời cũng nhấn mạnh về việc lựa chọn mô hình mạng nơ-ron tối ưu và tính quan trọng của việc xem xét tính phi tuyến khi áp dụng mô hình dự báo giá. Cho, Kyunghyun [2] đã đưa ra kết luận rằng GRU là một kiến trúc mạng nơ-ron hồi quy hiệu quả và có thể đạt được kết quả tốt trong bài toán dịch máy, so với LSTM. Dự báo chuỗi thời gian tài chính đã thu hút sự quan tâm, và ARIMA, LSTM và BiLSTM là ba phương pháp phổ biến. Nghiên cứu so sánh như Siami-Namini và nhóm nghiên cứu [3] cho thấy LSTM và BiLSTM vượt trội hơn ARIMA trong dự báo chuỗi thời gian tài chính. Ngoài ra, một nghiên cứu khác tập trung vào dự báo TEC trong ionosphere bằng cách sử dụng dữ liệu IRNSS và so sánh giữa Holt-Winter và exponential smoothing. Tác giả Venkateswarlu cho thấy cả hai phương pháp đều có khả năng dự báo TEC, nhưng Holt-Winter cho kết quả tốt hơn trong việc dự báo TEC trong ionosphere [4]. Chúng ta cùng tham khảo mô hình Linear Regression, NNAR qua cuốn "Forecasting: Principles and Practice" của tác giả Rob J Hyndman và George Athanasopoulos [5] .NNAR sử dụng mạng neural để mô hình hóa mối quan hệ tuyến tính và phi tuyến tính, trong khi Linear Regression dựa trên giả định về mối quan hệ tuyến tính giữa biến dự báo và biến độc lập. Các thuật toán được minh họa bằng ngôn ngữ R.

Trong bài báo này chúng tôi sẽ sử dụng ngôn ngữ python, với mô hình NNAR sẽ sử dụng thuật toán AR LSTM (Autoregressive LSTM) dùng gói Keras trong TensorFlow và so sánh kết quả với LSTM. [6]

Trong nghiên cứu của Robin M. Schmidt, tác giả trình bày một tổng quan về việc sử dụng mạng RNN và các biến thể của nó, bao gồm LSTM, trong dự đoán chuỗi thời gian. Nghiên cứu này khám phá các ứng dụng và phương pháp liên quan đến dự đoán chuỗi thời gian và cung cấp cái nhìn rõ ràng về hiệu quả của RNN trong lĩnh vực này.

Trong nghiên cứu của Lisandro Abrego1 and ParO€ ¨sterholm , nghiên cứu này xem xét tầm quan trọng của các yếu tố nội địa và bên ngoài đối với tăng trưởng GDP của Colombia. Kết quả cho thấy yếu tố nội địa, như tình hình đầu tư và chính sách tài khóa, chiếm phần lớn tăng trưởng, trong khi yếu tố bên ngoài như tăng trưởng kinh tế toàn cầu chỉ đóng vai trò nhỏ hơn. Các tác động từ chính sách tiền tệ và chính sách tài khóa đối với tăng trưởng là trung bình, trong khi tác động từ tăng trưởng toàn cầu mạnh hơn đáng kể. Dự báo tăng trưởng từ mô hình tương đối phù hợp với dự báo khác và cho thấy tăng trưởng kinh tế sẽ giảm trong thời gian tới. Tuy nhiên, nền kinh tế Colombia có khả năng chống chịu suy thoái từ các tác động bên ngoài và có tiềm năng phục hồi nhanh chóng.

# Kiến trúc mô hình

## Linear Regression

Trong trường hợp đơn giản nhất, mô hình hồi quy cho phép tồn tại mối quan hệ tuyến tính giữa biến phụ thuộc y và một biến độc lập duy nhất x.

Yt=β0+β1xt+εt

Trong đó:

Yt: là biến phụ thuộc

β0: hệ số chặn

β1: hệ số hồi quy cho biến độc lập

εt: sai số

Theo hình trên ta thấy các hệ số β0 và β1 lần lượt biểu thị giao điểm và hệ số góc của đường thẳng. Hệ số chặn β­0 đại diện cho giá trị dự đoán của y khi x=0, β1 đại diện cho sự thay đổi của y khi x thay đổi.

## Holt-Winters

Thuật toán Holt-winters là một phương pháp dự báo chuỗi thời gian, kết hợp Exponential Smoothing với các yếu tố mùa vụ (seasonality) và xu hướng (trend). Ba thành phần làm mịn (smoothing) cần thiết để dự báo trong mô hình holt-winters là level, trend, seasonal, những thành phần này nằm trong khoảng từ 0 đến 1. Dựa trên yếu tố mùa vụ, các mô hình Holt-winter được phân thành hai loại là mô hình cộng (additive models) và mô hình nhân (multiplicative models).

ADDITIVE MODEL

Mô hình này được sử dụng khi kích thước dữ liệu không phụ thuộc vào yếu tố mùa vụ. Dự báo cho dữ liệu trước 1 ngày (Ft) và dự báo trước *m* ngày (Ft+m) có thể được tính toán bằng các phương trình sau:

Để dự báo, trước tiên khởi tạo và ước tính các thành phần seasonal , có thể được thực hiện bằng các phương trình sau:

MULTIPLICATIVE MODEL

Mô hình được sử dụng khi kích thước dữ liệu phụ thuộc vào yếu tố mùa vụ. Dự báo cho dữ liệu *m* ngày tới ( Ft+m ) có thể được biểu diễn bằng các phương trình sau:

Trong mô hình nhân, các phương trình cho giá trị khởi tạo của level và trend giống với phương trình level và trend trong mô hình cộng. Các thành phần theo mùa khởi tạo được đưa ra là:

## ARIMA

Mô hình ARIMA là viết tắt của quá trình tự hồi quy (AutoRegression -AR), quá trình trung bình trượt (Moving Average – MA) và tích hợp sai phân Integrated – I.

* *Các loại mô hình ARIMA*

Mô hình ARIMA không có tính mùa vụ

Mô hình ARIMA có tính mùa vụ (Seasonal ARIMA – SARIMA)

* *Chuỗi dừng*

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 các yếu tố xu thế. Với hầu hết các phương pháp thống kê dự báo, đều phải đảm bảo tính dừng của chuỗi dữ liệu vì thế việc kiểm tra tính dừng là rất quan trọng.

Để kiểm định tính dừng của dữ liệu ta có hai phương pháp kiểm định phổ biến: Kiểm định Dickey Fuller3(DF) và Dickey Fuller cải tiến(ADF4).

* *Mô hình ARIMA không có tính mùa vụ*

Với p,d,q lần lượt là các số không âm

I(d): Integrated – So sánh sự khác nhau giữa d quan sát (Hiệu giữa giá trị hiện tại và d giá trị trước đó).

AR(p): Autoregression – là quá trình tìm mối quan hệ giữa dữ liệu hiện tài p dữ liệu quá khứ trước đó. (Gọi là lag)

yt = a0 + a1 yt-1 + a­2 yt-2 + … + ap yt-p + ԑt

Điều kiện dừng của việc chọn p:

MA(q): Moving Average – là quá trình tìm mối qua hệ giữa dữ liệu hiện tại và q phần lỗi quá khứ trước đó

yt =β0 + β1 ԑt-1 + β2 ԑt-2 + … + βq ԑt-q + µt

Điều kiện dừng của việc chọn q:

## TDNN

TDNN (Time Delay Neural Network) là một mô hình mạng nơ-ron thời gian được sử dụng để mô hình hóa và dự đoán chuỗi thời gian. Nó khác biệt với mạng nơ-ron thông thường bằng cách kết nối các nút ẩn với các giá trị vào của các bước thời gian trước đó. Cấu trúc này cho phép TDNN nhìn vào quá khứ và sử dụng thông tin lịch sử để dự đoán giá trị tiếp theo trong chuỗi thời gian. Một kiến trúc TDNN thông thường bao gồm lớp đầu vào, một hoặc nhiều lớp ẩn và lớp đầu ra.

* Lớp đầu vào: Lớp này có số nút bằng số quan sát trễ (lags) của dữ liệu thời gian mà ta muốn sử dụng. Mỗi nút đầu vào đại diện cho một giá trị quan sát trễ.
* Lớp ẩn: Lớp ẩn của TDNN sẽ có một số nút tùy chọn. Mỗi nút ẩn sẽ có một hàm kích hoạt phi tuyến như sigmoid, tanh hoặc ReLU.
* Lớp đầu ra: Lớp này có một nút đầu ra duy nhất để dự đoán giá trị tiếp theo của chuỗi thời gian.
* Biểu thức chung cho mạng TDNN nhiều lớp là:

Trong đó:

*f* và *g*là hàm kích hoạt ở lớp ẩn và lớp đầu ra.

*p* là số lượng nút đầu vào.

*q* là số lượng các nút ẩn.

*Bij* là trọng số được gắn vào kết nối giữa nút đầu vào thứ i và nút thứ j của lớp ẩn.

*αj* là trọng số được gắn vào kết nối giữa nút ẩn vào nút đầu ra.

*Yt-i* là đầu vào thứ i (lag) của mô hình.

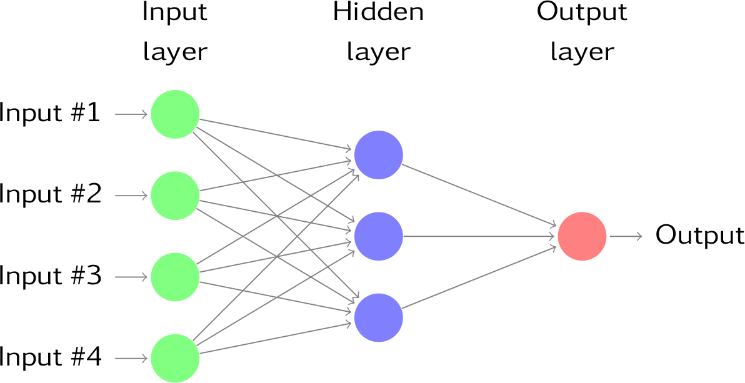
Trong quá trình huấn luyện, chúng ta sẽ điều chỉnh các trọng số Wi và W’j thông qua các thuật toán huẩn luyện như lan truyền ngược (backpropagation) để tối thiểu hóa sai số dự đoán giữa đầu ra thực tế và đầu ra dự đoán của mạng. Thông qua việc lặp lại quá trình này trên nhiều epoch và sử dụng các thuật toán tối ưu như gradient descent, mạng TDNN sẽ học cách mô hình hóa và dự đoán dữ liệu chuỗi thời gian.

## NNAR

* *Kiến trúc mạng nơ-ron (Neural Network)*

Mạng nơ-ron bao gồm nhiều nơ-ron được tổ chức thành các lớp. Các lớp bao gồm lớp đầu vào, lớp đầu ra và cũng có thể có lớp trung gian chứa các nơ-ron ẩn. Với một mạng nơ-ron đơn giản nhất sẽ không chứa các lớp ẩn và lúc này nó tương đương với hồi quy tuyến tính.

Khi chúng ta thêm một lớp trung gian với các nơ-ron ẩn, mạng nơ-ron sẽ trở nên phi tuyến tính được minh họa như hình bên dưới:



Hình Kiến trúc mạng của NNAR

* *Neural Network Autoregression* (*NNAR)*

Tương tự như khi sử dụng các giá trị lag trong mô hình hồi quy tuyến tính, khi dùng các quan sát trong quá khứ (các giá trị lag) làm đầu vào cho mạng nơ-ron để dự đoán các giá trị đầu ra ta gọi đây là mô hình mạng nơ-ron tự hồi quy - Neural Network Autoregression (NNAR). Xét các mạng chuyển tiếp với một lớp ẩn được ký hiệu: .

NNAR(P, k), trong đó P là số lag đầu vào, k là các nút trong lớp ẩn.

* *Hàm tương quan một phần (PACF)*

PACF hay còn gọi là hàm tự tương quan một phần là hàm dùng để xác định bậc tự tương quan trên laq đến p. PACF này được viết bằng phương trình.

## RNN

RNN (Recurrent Neural Network) là một loại neural network được sử dụng chủ yếu cho dữ liệu có cấu trúc chuỗi như chuỗi thời gian, ngôn ngữ tự nhiên và dữ liệu văn bản. Thuật toán RNN sử dụng khái niệm "trạng thái ẩn" (hidden state) để duy trì thông tin về quá khứ và sử dụng thông tin đó khi xử lý dữ liệu trong tương lai.

Chúng ta ký hiệu trạng thái ẩn và đầu vào tại thời điểm t lần lượt là Ht ∈ Rn x h và Xt∈Rn x d, trong đó n là số lượng mẫu, d là số lượng đầu vào của mỗi mẫu và h là số lượng đơn vị ẩn. Hơn nữa, chúng ta sử dụng ma trận trọng số Wxh ∈ Rd×h, ma trận Whh ∈ Rh×h để truyền thông tin từ trạng thái ẩn đến trạng thái ẩn, và tham số bias bh ∈ R1×h.

Cuối cùng, tất cả thông tin này được truyền qua một hàm kích hoạt φ, thường là hàm sigmoid logistic hoặc hàm tanh, để chuẩn bị các gradient để sử dụng trong quá trình lan truyền ngược. Kết hợp tất cả các ký hiệu này lại với nhau, ta thu được Phương trình 1 là biến ẩn ẩn và Phương trình 2 là biến đầu ra.

Ht = φh (XtWxh + Ht−1Whh + bh) (1)

Ot = φo (HtWho + bo) (2)

Vì Ht được tính toán đệ quy dựa trên Ht−1 và quá trình này diễn ra cho mỗi bước thời gian, RNN bao gồm các dấu vết của tất cả các trạng thái ẩn trước đó của Ht−1 cũng như chính Ht−1.

Nếu chúng ta so sánh notation cho RNN với notation tương tự cho Feedforward Neural Networks thẳng, chúng ta có thể thấy rõ sự khác biệt đã được mô tả trước đó. Trong Phương trình 3, chúng ta có thể thấy quá trình tính toán cho biến ẩn ẩn, trong khi Phương trình 4 thể hiện biến đầu ra.

H = φh (XWxh + bh) (3)

O = φo (HWho + bo) (4)

RNN được sử dụng rộng rãi trong nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, dịch máy, nhận dạng giọng nói, dự đoán chuỗi thời gian và nhiều ứng dụng khác có liên quan đến dữ liệu chuỗi.

## LSTM

LSTM là một loại RNN đặc biệt với các tính năng bổ sung để ghi nhớ chuỗi dữ liệu. Việc ghi nhớ xu hướng trước đó của dữ liệu có thể thực hiện được thông qua một số cổng cùng với một dòng bộ nhớ được tích hợp trong một LSTM điển hình.

Mỗi LSTM là một tập hợp các ô hoặc mô-đun hệ thống, nơi các luồng dữ liệu được thu thập và lưu trữ. Cấu trúc bên trong của một ô LSTM được thể hiện trong hình:

A picture containing diagram, line, plan, technical drawing

Description automatically generated

Hình Kiến trúc ô LSTM [7]

Một mô hình LSTM bao gồm ba cổng: *cổng quên* (forget gate), *cổng đầu vào* (input gate) và *cổng đầu ra* (output gate).

* *Forget gate:* Cổng này thường được sử dụng để đưa ra quyết định về thông tin nào cần được xóa khỏi bộ nhớ LSTM. Đầu ra của cổng này được tính như sau:
* *Input gate*: Cổng này đưa ra quyết định liệu thông tin mới có được thêm vào bộ nhớ LSTM hay không. Cổng này bao gồm hai lớp: 1) lớp *sigmoid* và 2) lớp “*tanh*”. Đầu ra của hai lớp này được tính toán như sau:

Sự kết hợp của hai lớp này cung cấp một bản cập nhật cho bộ nhớ LSTM, trong đó giá trị hiện tại bị quên bằng cách sử dụng lớp cổng quên. Phương trình sau biểu diễn phương trình toán học của nó:

* *Output gate:* Cổng này sử dụng một lớp *sigmoid* để đưa ra quyết định phần nào của bộ nhớ LSTM đóng góp vào đầu ra. Sau đó, nó thực hiện một hàm *tanh* phi tuyến tính để ánh xạ các giá trị giữa −1 và 1, kết quả được nhân với đầu ra của một lớp sigmoid. Phương trình sau đây biểu thị các công thức để tính toán đầu ra: [8]

## GRU

Gated Recurrent Unit (GRU) là một loại mô hình mạng nơ-ron hồi quy (RNN) được giới thiệu nhằm giải quyết vấn đề mất mát thông tin xa trong các mạng RNN truyền thống. GRU sử dụng cơ chế cổng để điều chỉnh quá trình truyền thông tin trong mạng và giúp mô hình nhớ được thông tin quan trọng trong quá khứ.

GRU có cấu trúc đơn giản hơn so với mô hình LSTM nhưng vẫn đạt được hiệu suất tương tự trong nhiều nhiệm vụ. Nó bao gồm hai cổng chính:

* Update gate: Cổng cập nhật xác định tỷ lệ thông tin nào sẽ được cập nhật từ trạng thái trước đó và thông tin mới nào sẽ được truyền vào trạng thái hiện tại. Nó quyết định liệu mô hình có nên quên hoặc ghi nhớ thông tin từ quá khứ.
* Reset gate: Cổng khôi phục xác định tỷ lệ thông tin nào từ trạng thái trước đó sẽ bị đặt lại (reset) để tạo không gian cho thông tin mới. Nó giúp loại bỏ những thông tin không cần thiết hoặc không quan trọng.

Các cổng cập nhật và khôi phục được kết hợp để tính toán trạng thái ẩn mới (ht) trong GRU. Các giá trị của hai cổng này sẽ quyết định phần nào của thông tin từ trạng thái trước đó sẽ được cập nhật và phần nào sẽ bị đặt lại, từ đó giúp mô hình hiểu và ghi nhớ thông tin quan trọng trong chuỗi dữ liệu. GRU sử dụng một lớp ẩn duy nhất để tính toán trạng thái ẩn mới. Điều này giúp mô hình tiết kiệm tài nguyên tính toán và thời gian huấn luyện.

## BVAR

Bayesian Vector Autoregression (BVAR) sử dụng các phương pháp Bayes để ước lượng mô hình VAR. BVAR khác với các mô hình VAR tiêu chuẩn ở chỗ các tham số mô hình được coi là biến ngẫu nhiên, có xác suất biết trước, thay vì các giá trị cố định.

Thuật toán BVAR (Bayesian Vector Autoregression) là một phương pháp dự báo trong lĩnh vực thống kê bayesian được sử dụng để mô hình hóa và dự đoán hệ thống các biến ngẫu nhiên liên kết với nhau trong thời gian. BVAR xem xét mối quan hệ đồng thời giữa các biến phụ thuộc và các biến độc lập trong một hệ thống dữ liệu chuỗi thời gian.

Công thức chính của thuật toán BVAR được biểu diễn dưới dạng mô hình Vector Autoregression (VAR) với một tiếp cận Bayesian. Mô hình VAR có dạng:

Yt = c + A1 \* Yt-1 + A2 \* Yt-2 + ... + Ap \* Yt-p + et

Trong đó:

Yt là vector các biến phụ thuộc tại thời điểm t

c là vector hệ số chặn

A1, A2, ..., Ap là các ma trận hệ số hồi quy

Yt-1, Yt-2, ..., Yt-3 là vector các biến phụ thuộc tại các thời điểm trước đó

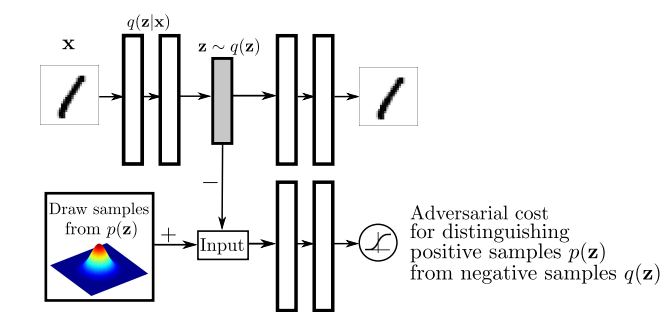
et là vector các thành phần lỗi ngẫu nhiên

Thuật toán BVAR áp dụng phương pháp Bayesian để ước lượng các tham số trong mô hình VAR. Nó sử dụng các tiền kiến thức thông qua việc đặt các prior xác định trước cho các tham số và ước lượng posterior của chúng thông qua Markov Chain Monte Carlo (MCMC) hoặc các phương pháp khác.

Việc sử dụng tiếp cận Bayesian trong BVAR cho phép tính toán phân phối posterior của các tham số, giúp ta đánh giá được sự không chắc chắn trong dự đoán và cung cấp phạm vi tin cậy cho kết quả dự báo.

## AAE

AAE (Adversarial Autoencoder) là một kiến trúc mô hình học sâu kết hợp giữa **Autoencoder** và mạng **Adversarial** nhằm chủ yếu để tạo ra dữ liệu mới có cùng phân phối với dữ liệu huấn luyện ban đầu và thực hiện các nhiệm vụ như phân loại, phân đoạn và tái tạo dữ liệu.



Hình Mô hình AAE

* **Autoencoder** là một kiến trúc neural network được sử dụng để học biểu diễn tự động (latent representation) của dữ liệu đầu vào. Nó bao gồm hai phần chính: encoder (bộ mã hóa) và decoder (bộ giải mã). Encoder chuyển đổi dữ liệu đầu vào thành biểu diễn ẩn (latent representation) có chiều thấp hơn, trong khi decoder chuyển đổi biểu diễn ẩn trở lại dữ liệu ban đầu.
* Mạng **Adversarial** là một mạng được sử dụng để tạo ra dữ liệu giả mô phỏng dữ liệu thực tế. Nó bao gồm hai thành phần chính: generator (bộ sinh) và discriminator (bộ phân biệt). Generator nhận đầu vào là một biểu diễn ẩn ngẫu nhiên và cố gắng tạo ra dữ liệu giả mô phỏng dữ liệu thực tế. Trong khi đó, discriminator nhận đầu vào là dữ liệu thực tế và dữ liệu giả và cố gắng phân biệt chúng.

# Thực nghiệm

## Dữ liệu

Investing.com là một trang web tài chính hàng đầu cung cấp thông tin và dịch vụ về thị trường tài chính và đầu tư. Trang web này cung cấp thông tin về các chỉ số chứng khoán, tỷ giá ngoại tệ, giá và biểu đồ của các công cụ tài chính như cổ phiếu, hàng hóa, tiền điện tử và nhiều loại tài sản khác. Các tập dữ liệu được dùng trong nghiên cứu này dựa trên dữ liệu lịch sử do trang Investing cung cấp bao gồm giá cổ phiếu các công ty Alexandria Real Estate Equities Inc (ARE), Apollo Commercial Real Estate Finance Inc (ARI), Mid-America Apartment Communities Inc (MAA) được thu thập từ ngày 02/01/2015 đến ngày 16/06/2023 Tập dữ liệu bao gồm các thông tin liên quan đến giá cổ phiếu như sau:

* Date: ngày giao dịch của cổ phiếu.
* Open: giá mở cửa của cổ phiếu.
* Price: giá đóng cửa của cổ phiếu.
* High: giá cao nhất của cổ phiếu.
* Low: giá thấp nhất của cổ phiếu.
* Vol: khối lượng giao dịch của cổ phiếu trong ngày giao dịch tương ứng.
* Change %: phần trăm thay đổi giá của cổ phiếu so với ngày giao dịch trước đó.

Trong nghiên cứu này, chúng tôi chia tập dữ liệu thành ba phần: train, test và validate. Các phần dữ liệu được chia theo tỉ lệ tương ứng như sau: 70% - 20% - 10% hoặc 60% - 20% - 20%. Chúng tôi tập trung vào giá trị Price của cổ phiếu để dự đoán. Bảng 1 cung cấp các thống kê mô tả cơ bản về dữ liệu, bao gồm giá trị trung bình, độ lệch chuẩn, giá trị lớn nhất, giá trị nhỏ nhất và số lượng giá trị có sẵn, ...

|  |  |  |  |
| --- | --- | --- | --- |
| Giá trị | ARE | ARI | MAA |
| Count | 2129 | 2129 | 2129 |
| Mean | 136.61879 | 15.45914 | 122.69520 |
| Min | 71.65 | 4.24 | 71.15 |
| Max | 223.57 | 19.68 | 229.44 |
| Mode | 92.64 | 18.66 | 100.0 |
| Median | 128.24 | 16.62 | 107.98 |
| Variance | 1118.77045 | 10.98388 | 1385.89197 |
| ST dev | 33.44 | 3.31419 | 37.22757 |
| CV | 0.26082 | 0.19941 | 0.34476 |
| Skew | 0.36263 | -0.85507 | 0.93720 |
| Kurtosis | -0.62161 | -0.48045 | -0.06742 |
| 25% | 112.29 | 13.0 | 96.33 |
| 50% | 128.24 | 16.62 | 107.98 |
| 75% | 160.71 | 18.26 | 147.31 |
| 100% | 223.57 | 19.68 | 229.44 |
| Box Plot | Ảnh có chứa biểu đồ, ảnh chụp màn hình, hàng, Hình chữ nhật  Mô tả được tạo tự động | Ảnh có chứa ảnh chụp màn hình, biểu đồ, Hình chữ nhật, hàng  Mô tả được tạo tự động | Ảnh có chứa biểu đồ, ảnh chụp màn hình, văn bản, hàng  Mô tả được tạo tự động |
| Histogram | Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Sơ đồ  Mô tả được tạo tự động | Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Sơ đồ  Mô tả được tạo tự động | Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Sơ đồ  Mô tả được tạo tự động |

Bảng Thống kê mô tả dữ liệu

Theo bảng 1, trong 3 loại cổ phiếu thì ARE và MAA có giá trị trung bình gần tương đương nhau và chênh lệch khá lớn so với ARI. Phương sai và độ lệch chuẩn của ARE và MAA khá lớn, cho thấy sự biến động lớn của dữ liệu. Skew của 3 bộ dữ liệu có sự khác nhau dẫn đến độ phân phối của chúng cũng khác nhau, MAA (skewness gần bằng 0) xu hướng gần đối xứng, ARE (skewness âm) phân phối lệch về phía trái, ARI (skewness dương) xu hướng lệch về phía bên phải. Kurtosis, Cả 3 đều âm, phân phối của cả 3 đều nhọn hơn so với phân phối chuẩn, trong đó MAA có phân phối gần với phân phối chuẩn (kurtosis gần bằng 0).

## Công cụ

Trong quá trình nghiên cứu, chúng tôi sử dụng Google Colab làm môi trường phát triển và chạy mã nguồn. Ngôn ngữ lập trình Python được sử dụng với các thư viện quan trọng như NumPy, Pandas, Matplotlib.pyplot, scikit-learn và TensorFlow. Các công cụ này giúp chúng tôi thực hiện các thực nghiệm, phân tích dữ liệu và xây dựng các mô hình dự đoán giá cổ phiếu. Kết hợp của các công cụ này đã giúp chúng tôi đánh giá hiệu quả của các phương pháp dự đoán và tìm ra thuật toán phù hợp cho mục tiêu đầu tư và quản lý rủi ro.

## Các độ đo đánh giá

Trong nghiên cứu này, chúng tôi sử dụng ba độ đo để đánh giá hiệu suất của các mô hình: RMSE (Root mean squared error), MAE (Mean Absolute Error) và MAPE (Mean absolute percentage error).

RMSE đo lường sự chênh lệch giữa các giá trị dự đoán và giá trị thực tế bằng cách tính căn bậc hai của trung bình bình phương sai.

MAE tính trung bình giá trị tuyệt đối của các sai số dự đoán so với giá trị thực tế.

MAPE đo lường mức độ sai số trung bình giữa các dự đoán và giá trị thực tế trong các dự báo.

# Kết quả

## Độ đo đánh giá

### *Tập dữ liệu ARE*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Train-test-validate | RMSE | MAPE | MAE |
| Linear Regression | 7-2-1 | 20.50 | 19.34 | 18.38 |
| 6-2-2 | 18.37 | 15.35 | 17.78 |
| Holt-Winters | 7-2-1 | 21.37 | 9.35 | 16.77 |
| 6-2-2 | 13.90 | 7.03 | 11.36 |
| ARIMA | 7-2-1 | 25.70 | 11.06 | 21.17 |
| 6-2-2 | 13.51 | 6.75 | 10.93 |
| TDNN | 7-2-1 | 5.45 | 2.25 | 4.32 |
| 6-2-2 | 5.35 | 2.32 | 4.05 |
| NNAR | 7-2-1 | 0.03 | 3.68 | 0.02 |
| 6-2-2 | 0.03 | 4.23 | 0.03 |
| RNN | 7-2-1 | 2.7 | 1.47 | 2.06 |
| 6-2-2 | 3.23 | 1.54 | 2.52 |
| LSTM | 7-2-1 | 3.16 | 1.35 | 2.50 |
| 6-2-2 | 3.37 | 1.40 | 2.29 |
| GRU | 7-2-1 | 2.72 | 1.1 | 1.97 |
| 6-2-2 | 3.33 | 1.34 | 2.16 |
| AAE | 7-2-1 | 11.52 | 7.69 | 10.02 |
| 6-2-2 | 28.81 | 13.64 | 24.34 |

Bảng 2 Bảng thống kê độ đo của các mô hình trên tập dữ liệu ARE

### *Tập dữ liệu ARI*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Train-test-validate | RMSE | MAPE | MAE |
| Linear Regression | 7-2-1 | 4.11 | 22.68 | 3.76 |
| 6-2-2 | 2.48 | 11.71 | 2.05 |
| Holt-Winters | 7-2-1 | 3.67 | 23.30 | 3.33 |
| 6-2-2 | 7.22 | 65.42 | 6.50 |
| ARIMA | 7-2-1 | 3.08 | 18.62 | 2.69 |
| 6-2-2 | 7.01 | 63.25 | 6.22 |
| TDNN | 7-2-1 | 0.26 | 1.51 | 0.20 |
| 6-2-2 | 0.53 | 3.63 | 0.35 |
| NNAR | 7-2-1 | 0.02 | 3.02 | 0.02 |
| 6-2-2 | 0.06 | 16.25 | 0.05 |
| RNN | 7-2-1 | 2.71 | 1.47 | 2.06 |
| 6-2-2 | 3.22 | 1.54 | 2.52 |
| LSTM | 7-2-1 | 0.28 | 1.69 | 0.22 |
| 6-2-2 | 1.29 | 10.72 | 0.97 |
| GRU | 7-2-1 | 0.25 | 1.47 | 0.2 |
| 6-2-2 | 1.22 | 10.50 | 1.01 |
| AAE | 7-2-1 | 0.49 | 3.7 | 0.4 |
| 6-2-2 | 0.38 | 2.66 | 0.31 |

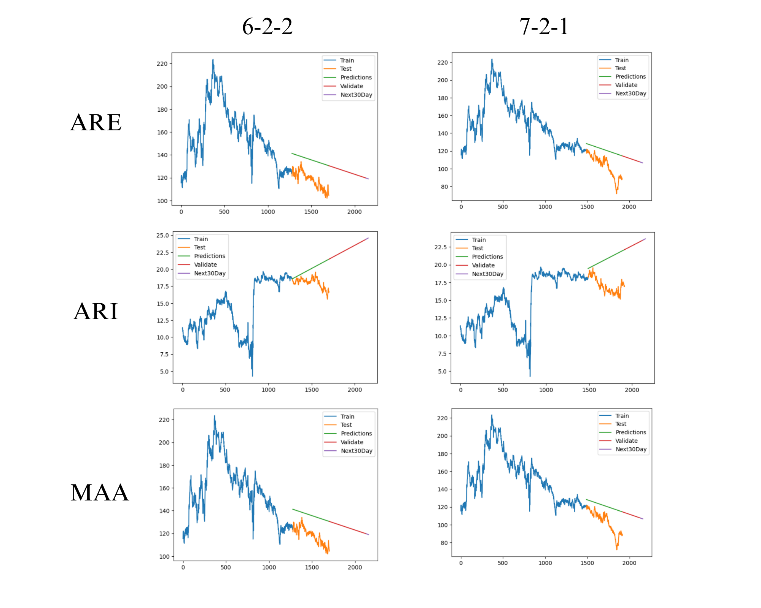
Bảng 3 Bảng thống kê độ đo của các mô hình trên tập dữ liệu ARI

### *Tập dữ liệu MAA*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Train-test-validate | RMSE | MAPE | MAE |
| Linear Regression | 7-2-1 | 20.5 | 19.34 | 18.38 |
| 6-2-2 | 18.37 | 15.35 | 17.78 |
| Holt-Winters | 7-2-1 | 53.01 | 24.20 | 46.30 |
| 6-2-2 | 23.98 | 16.36 | 21.40 |
| ARIMA | 7-2-1 | 60.26 | 27.83 | 53.10 |
| 6-2-2 | 24.05 | 16.44 | 21.46 |
| TDNN | 7-2-1 | 13.17 | 5.71 | 11.04 |
| 6-2-2 | 5.94 | 2.77 | 4.19 |
| NNAR | 7-2-1 | 0.02 | 3.89 | 0.02 |
| 6-2-2 | 0.07 | 15.62 | 0.04 |
| RNN | 7-2-1 | 2.7 | 1.47 | 2.06 |
| 6-2-2 | 3.22 | 1.5 | 2.52 |
| LSTM | 7-2-1 | 3.60 | 1.44 | 2.69 |
| 6-2-2 | 3.00 | 1.57 | 2.05 |
| GRU | 7-2-1 | 2.63 | 1.11 | 2.003 |
| 6-2-2 | 2.9 | 1.5 | 1.92 |
| AAE | 7-2-1 | 11.83 | 6.8 | 10.5 |
| 6-2-2 | 10.5 | 4.1 | 8.0 |

Bảng 4 Bảng thống kê độ đo của các mô hình trên tập dữ liệu MAA

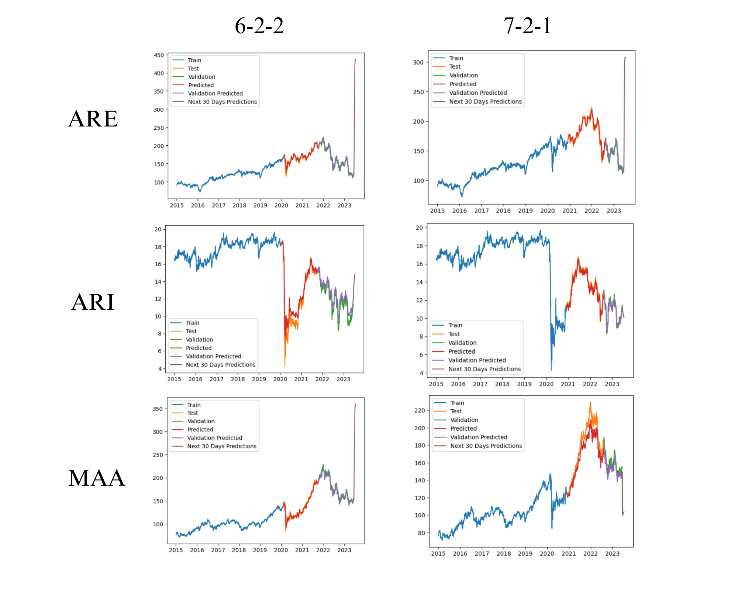
## Kết quả dự đoán



Hình Kết quả dự đoán của Linear regression

A picture containing text, diagram, plot, line

Description automatically generated

Hình Kết quả dự đoán của Holt-Winters 

Hình Kết quả dự đoán của NNAR

A picture containing text, diagram, plot, line

Description automatically generated

Hình Kết quả dự đoán của LSTM

Ảnh có chứa văn bản, biểu đồ, hàng, Sơ đồ

Mô tả được tạo tự động

Hình Kết quả dự đoán của TDNN

Ảnh có chứa văn bản, biểu đồ, chữ viết tay, hàng

Mô tả được tạo tự động

Hình Kết quả dự đoán của GRU

A picture containing text, diagram, plot, line

Description automatically generated

Hình Kết quả dự đoán của RNN

A picture containing text, diagram, plot, line

Description automatically generated

Hình Kết quả dự đoán của AAE

A picture containing text, diagram, line, plot

Description automatically generated

Hình Kết quả dự đoán của ARIMA

## Các thuật toán tốt nhất

Qua quá trình thực hiện 3 tập dữ liệu trên 10 thuật toán với 2 tỉ lệ 7-2-1 và 6-2-2, chúng tôi thu được kết quả:

Đối với tập dữ liệu ARE, mô hình NNAR đạt hiệu suất tốt nhất trên hầu hết các độ đánh giá, ở cả hai tỉ lệ (7-2-1, 6-2-2), mô hình cho kết quả RMSE tối ưu là 0.03 cho thấy giá trị cổ phiếu dự báo tương đương với giá trị thật. Dựa trên độ đo RMSE chúng tôi kết luận mô hình NNAR là mô hình tốt nhất cho việc dự báo giá cổ phiểu của công ty ARE, theo sau đó là 2 mô hình RNN và GRU với RMSE lần lượt là 2.7 (7-2-1), 3.23 (6-2-2) và 2.72 (7-2-1), 3.33 (6-2-2).

Đối với tập dữ liệu ARI, mô hình NNAR đạt hiệu suất tốt nhất trên hầu hết các độ đánh giá, mô hình cho kết quả RMSE tối ưu là 0.02 (7-2-1), 0.06 (6-2-2). Dựa trên độ đo RMSE chúng tôi kết luận mô hình NNAR là mô hình tốt nhất cho việc dự báo giá cổ phiểu của công ty ARE, theo sau đó là 2 mô hình AAE và GRU với RMSE lần lượt là 0.49 (7-2-1), 0.38 (6-2-2) và 0.25 (7-2-1), 1.22 (6-2-2).

Đối với tập dữ liệu MAA, mô hình NNAR đạt hiệu suất tốt nhất trên hầu hết các độ đánh giá, mô hình cho kết quả RMSE tối ưu là 0.02 (7:2:1), 0.07 (6:2:2). Dựa trên độ đo RMSE chúng tôi kết luận mô hình NNAR là mô hình tốt nhất cho việc dự báo giá cổ phiểu của công ty ARE, theo sau đó là 2 mô hình GRU và RNN với RMSE lần lượt là 2.63 (7-2-1), 2.9(62-2) và 2.7(7-2-1), 3.22 (6-2-2).

Sau khi đã xác định hai mô hình tốt nhất cho mỗi tỉ lệ thực nghiệm ở mỗi loại cổ phiếu, chính tôi tiến hành áp dụng chúng để dự báo giá cổ phiếu trong 30 ngày tiếp theo tình từ ngày 19/06/2023 và trực quan kết quả dự báo dưới dạng biểu đồ đường.

# Kết luận

Trong nghiên cứu này, chúng tôi đã tiến hành phân tích và so sánh hiệu suất của một loạt thuật toán dự đoán giá cổ phiếu trong môi trường dữ liệu thời gian. Các thuật toán được xem xét bao gồm Linear Regression, ARIMA, Holt-Winters, TDNN, NNAR, AAE, BVAR, RNN, LSTM và GRU. Kết quả cho thấy NNAR, GRU và LSTM là ba thuật toán cho ra kết quả tốt nhất trong việc dự đoán giá cổ phiếu. Cả ba thuật toán này đã chứng minh khả năng dự đoán giá cổ phiếu một cách hiệu quả.

Tuy nhiên, chúng tôi nhận thấy rằng việc áp dụng thuật toán BVAR gặp hạn chế và chưa đạt được kết quả như mong đợi. Cần lưu ý rằng các thuật toán dự đoán giá cổ phiếu chỉ dựa trên dữ liệu lịch sử và có rất nhiều yếu tố khác có thể ảnh hưởng đến giá cổ phiếu. Do đó, việc dự đoán giá cổ phiếu vẫn còn mức độ không chắc chắn. Khi áp dụng dự đoán giá cổ phiếu trong thực tế, cần phân tích kỹ lưỡng và cân nhắc các yếu tố khác để có kết quả chính xác.

Tóm lại, nghiên cứu này đã đánh giá hiệu suất của một loạt thuật toán dự đoán giá cổ phiếu. NNAR, GRU và LSTM đã cho ra kết quả tốt nhất trong số các thuật toán được nghiên cứu. Tuy nhiên, chúng tôi cũng ghi nhận hạn chế trong việc áp dụng thuật toán BVAR. Những kết quả này có thể đóng góp quan trọng cho phát triển và ứng dụng các mô hình dự đoán giá cổ phiếu trong thực tế.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | Jha, Girish K., and Kanchan Sinha, "Time-delay neural networks for time series prediction: an application to the monthly wholesale price of oilseeds in India," *Neural Computing and Applications 24,* 2014. |
| [2] | Cho, Kyunghyun, et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation.". |
| [3] | Siami-Namini, Sima, Neda Tavakoli, and Akbar Siami Namin., "A comparative analysis of forecasting financial time series using arima, lstm, and bilstm.," 2019. |
| [4] | Venkateswarlu, G., and A. D. Sarma., "Performance of Holt-Winter and exponential smoothing methods for forecasting ionospheric TEC using IRNSS data," *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT),* 2017. |
| [5] | Hyndman, Rob J., and George Athanasopoulos, Forecasting: principles and practice, OTexts, 2018. |
| [6] | TensorFlow Time series forecasting, "TensorFlow," [Online]. Available: https://www.tensorflow.org/tutorials/structured\_data/time\_series?hl=en. |
| [7] | As' ad, Mohamad., "Finding the Best ARIMA Model to Forecast Daily Peak Electricity Demand," *2012.* |
| [8] | Schmidt, Robin M., "Recurrent neural networks (rnns): A gentle introduction and overview.," 2019. |
| [9] | Siami-Namini, Sima, and Akbar Siami Namin., "Forecasting economics and financial time series: ARIMA vs. LSTM.," 2018. |
| [10] | Makhzani, Alireza, et al, "Adversarial autoencoders.," 2015. |
| [11] | Benmahdjoub, Khalida, Zohra Ameur, and Mina Boulifa., "Forecasting of rainfall using time delay neural network in Tizi-Ouzou (Algeria)," *Energy Procedia 36 ,* 2013. |
| [12] | As’ad, Mohamad, and Sigit Setyowibowo Sujito, "Prediction of Daily Gold Prices Using an Autoregressive Neural Network.," *Inform 5.2,* 2020. |
| [13] | Mou, Hanlin, and Junsheng Yu., "CNN-LSTM prediction method for blood pressure based on pulse wave.," *Electronics 10.14,* 2021. |
| [14] | Abrego, Lisandro, and Pär Österholm., "External linkages and economic growth in Colombia: insights from a bayesian Var model.," 2008. |
| [15] | Mondal, Prapanna, Labani Shit, and Saptarsi Goswami., "Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices.," *International Journal of Computer Science, Engineering and Applications 4.2,* 2014. |