ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA: KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

VÕ TRẦN ĐẠI

NGUYỄN NHẬT THIÊN TÂN

ĐỒ ÁN HỌC MÁY THỐNG KÊ

XÂY DỰNG MÔ HÌNH DỰ ĐOÁN GIÁ TƯƠNG LAI CỦA THỊ TRƯỜNG CHỨNG KHOÁN DỰA TRÊN DỮ LIỆU CỦA NĂM TRƯỚC

STOCK PRICE PREDICTION

GIẢNG VIÊN HƯỚNG DẪN

HỒ THÁI NGỌC

ThS. VÕ DUY NGUYÊN

TS. NGUYỄN TẤN TRẦN MINH KHANG

TP. HỒ CHÍ MINH, 2021

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA: KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

VÕ TRẦN ĐẠI - 19521308

NGUYỄN NHẬT THIÊN TÂN - 19520922

ĐỒ ÁN HỌC MÁY THỐNG KÊ

XÂY DỰNG MÔ HÌNH DỰ ĐOÁN GIÁ TƯƠNG LAI CỦA THỊ TRƯỜNG CHỨNG KHOÁN DỰA TRÊN DỮ LIỆU CỦA NĂM TRƯỚC

STOCK PRICE PREDICTION

GIẢNG VIÊN HƯỚNG DẪN

HỒ THÁI NGỌC

ThS. VÕ DUY NGUYÊN

TS. NGUYỄN TẤN TRẦN MINH KHANG

TP. HỒ CHÍ MINH, 2021

LỜI CẢM ƠN

Nhóm chúng em xin gửi lời cảm ơn chân thành đến thầy Hồ Thái Ngọc, Ths. Võ Duy Nguyên và TS. Nguyễn Tấn Trần Minh Khang, cảm ơn các Thầy đã giảng dạy tận tình, giúp đỡ và chỉ bày chúng em từng ngày, không chỉ về mặt kiến thức mà còn là kĩ năng sống để có thể hoàn thành được đồ án này.

Đồ án này sẽ còn nhiều thiếu sót, chúng em mong nhận được những lời nhận xét đánh giá từ các Thầy để từ đó rút ra được những kinh nghiệm quý báu, có cơ hội ngày càng hoàn thiện bản thân.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	LSTM	Mô hình Long Short-Term Memory
2	ARIMA	Mô hình Autoregressive Integrated Moving Average
3	ANN	Mô hình Artificial Neural Networks
4	RMSE	Độ đo root-mean-square error
5	MAPE	Độ đo mean absolute percentage error
6	EDA	Phân tích khám phá dữ liệu
7	FB	Mã cổ phiếu của Facebook
8	MSFT	Mã cổ phiếu của Microsoft
9	TSLA	Mã cổ phiếu của Tesla
10	AAPL	Mã cổ phiếu của Apple
11	AR	Mô hình Autoregression
12	PACF	Partial correlation coefficients
13	SARIMA	Mô hình Seasonal Autoregressive Integrated Moving Average
14	SARIMAX	Mô hình Seasonal Autoregressive Integrated Moving Average with exogenos variables
15	MA	Mô hình Moving Average

TÓM TẮT

Dự đoán thị trường chứng khoán đã được xác định là một vấn đề thực tế rất quan trọng trong lĩnh vực kinh tế. Tuy nhiên, dự đoán kịp thời về thị trường thường được coi là một trong những vấn đề khó khăn nhất do đặc điểm của thị trường chứng khoán là nhiễu và dễ biến động. Trong khi những người ủng hộ giả thuyết thị trường hiệu quả tin rằng không thể dự đoán chính xác giá cổ phiếu, có những định đề chính thức chứng minh rằng việc lập mô hình chính xác và thiết kế các biến thích hợp có thể dẫn đến các mô hình sử dụng giá cổ phiếu và mô hình biến động giá cổ phiếu có thể được dự đoán rất chính xác. Các nhà nghiên cứu cũng đã làm việc về phân tích kỹ thuật cổ phiếu với mục tiêu xác định các mô hình trong biến động giá cổ phiếu bằng cách sử dụng các kỹ thuật khai thác dữ liệu tiên tiến. Trong nội dung đồ án này, chúng em đề xuất một cách tiếp cận mô hình kết hợp để dự đoán giá cổ phiếu xây dựng các mô hình dựa trên học máy và học sâu khác nhau, cụ thể là Mô hình Long Short Term Memory (LSTM) và các phương pháp truyền thống khác như mô hình ARIMA vào dự đoán giá cổ phiếu vào ngày hôm sau. Hơn nữa, sử dụng dự đoán của chúng em có thể tạo tiền đề cho một sự tham khảo để thực hiện các chiến lược đầu tư ngắn hạn, trung hạn và thậm chí là dài hạn. Dữ liệu đầu vào của chúng em chỉ chứa giá đóng cửa của cổ phiếu để áp dụng vào các mô hình. Thông qua các thử nghiệm của mình, chúng em tìm ra phương pháp xử lý và phù hợp nhất cho bài toán toán dự đoán time-series (chuỗi thời gian) với bộ dữ liệu nghiên cứu.

MỞ ĐẦU

Đặt vấn đề

Thị trường chứng khoán là nơi mà cổ phiếu có thể được chuyển nhượng, mua bán và lưu thông. Nó đã tồn tại khoảng 400 năm và đã trở thành một kênh quan trọng để các công ty lớn huy động vốn từ các nhà đầu tư. Một mặt, thông qua việc phát hành cổ phiếu, một lượng lớn vốn chảy vào thị trường chứng khoán, giúp tăng cường cấu thành hữu cơ của vốn doanh nghiệp bằng cách thúc đẩy tập trung vốn, thúc đẩy mạnh mẽ sự phát triển của nền kinh tế hàng hóa. Mặt khác, thông qua việc luân chuyển cổ phiếu, các quỹ được gộp lại và việc tích lũy vốn được thúc đẩy một cách hiệu quả. Vì vậy, thị trường chứng khoán được coi là phong vũ biểu của các hoạt động kinh tế, tài chính của một quốc gia hay khu vực. Đặc biệt, giá giao dịch của thị trường chứng khoán thường được dùng làm chỉ số cho giá cả và số lượng chứng khoán vì nó có thể phản ánh một cách khách quan quan hệ cung cầu của thị trường chứng khoán. Tuy nhiên, cơ chế hình thành giá cổ phiếu khá phức tạp. Việc sử dụng kết hợp các yếu tố khác nhau và hành vi đặc biệt của các yếu tố riêng lẻ, bao gồm các yếu tố chính trị, kinh tế và thị trường cũng như công nghệ và hành vi của nhà đầu tư, tất cả sẽ dẫn đến những thay đổi trong giá cổ phiếu. Kết quả là, giá cổ phiếu liên tục thay đổi, và sự thay đổi này tạo không gian sống cho các hoạt động đầu cơ và làm tăng rủi ro cho thị trường chứng khoán. Loại rủi ro này không những có thể gây thiệt hại về kinh tế cho nhà đầu tư mà còn có thể mang lại những tác dụng phụ nhất định đối với công cuộc xây dựng kinh tế của doanh nghiệp và quốc gia. Trong nhiều thập kỷ, đã có thảo luận trong khoa học về khả năng của một kỳ tích như vậy và đáng chú ý trong các tài liệu liên quan rằng hầu hết các dự đoán các mô hình không cung cấp dự đoán chính xác theo nghĩa chung. Tuy nhiên, có rất nhiều nghiên cứu từ các các ngành đang tìm cách đối mặt với thách thức đó, đưa ra một nhiều cách tiếp cận để đạt được mục tiêu đó. Một cách tiếp cận phổ biến là sử dụng thuật ngữ Học máy - nhịp điệu để tìm hiểu từ dữ liệu lịch sử giá để dự đoán

giá trong tương lai. Trên tập dữ liệu này, mô hình sẽ được huấn luyện, đánh giá và sẽ cố gắng dự đoán liệu giá của một cổ phiếu cụ thể sẽ tăng hay không trong một ngày.

Mục tiêu

Qua đồ án lần này, chúng em cố gắng để nghiên cứu và có được hiểu biết rõ ràng hơn về vai trò và tác dụng của học máy trong các lĩnh vực khác nhau trong cuộc sống, một trong số đó là đề tài mà chúng em nghiên cứu trong lần này - dự đoán giá cổ phiếu. Đồng thời, bên cạnh việc nghiên cứu lý thuyết, chúng em cũng muốn tìm hiểu và cài đặt mô hình do chúng em huấn luyện, qua đó có thể kiểm chứng những kiến thức đã tìm hiểu và làm rõ hơn về ưu nhược điểm của từng loại mô hình.

Đối tượng và phạm vi nghiên cứu

- **Đối tượng:** Phương pháp học máy và học sâu cho bài toán dự đoán chuỗi thời gian, cụ thể là dự đoán giá chứng khoán.
- **Phạm vi:** Đồ án tập trung chủ yếu vào việc dự đoán giá cổ phiếu của 4 cổ phiếu thuộc nhóm Big Tech: Apple, Tesla, Microsoft, Facebook từ lúc được niêm yết trên sàn chứng khoán New York đến ngày 10-11-2017.
  Về giới hạn đồ án, chúng em chủ yếu tập trung nghiên cứu quy trình, các thuật toán và các phương pháp dự đoán chuỗi thời gian.

Kết quả nghiên cứu

Nghiên cứu của chúng em đạt được các kết quả sau:

- Thử nghiệm với mô hình học sâu (deep learning) LSTM và mô hình học máy (machine learning) ARIMA trên 4 mẫu dữ liệu được tách từ bộ dữ liệu gốc.
- Mô hình LSTM cho kết quả giá đóng cửa của cổ phiếu dự đoán có độ sai lệch thấp so với giá thật tương đương với mô hình ARIMA.

- Phân tích các kết quả đạt được, từ đó nhận thấy một số ưu và nhược điểm của mô hình học sâu và học máy cùng với kỹ thuật xử lý dữ liệu. Từ những kết quả trong đồ án này, chúng em mong có thể tạo thêm một kênh tham khảo cho các nhà đầu tư chứng khoán.

Cấu trúc đồ án

Đồ án gồm 5 chương với các nội dung chính như sau:

- **Chương 1:** Tổng quan

Giới thiệu về lĩnh vực dự đoán giá chứng khoán trong tương lai và tầm quan trọng của việc đưa ra dự đoán về giá của một cổ phiếu hiện nay

- **Chương 2:** Bộ dữ liệu nghiên cứu

Trong chương này, chúng em giới thiệu về các bộ dữ liệu giá chứng khoán đã sử dụng và đưa ra các nhận xét, phân tích và đánh giá về đặc điểm của các bộ dữ liệu để đề xuất được những phương pháp tiền xử lý của bộ dữ liệu cũng như xây dựng các mô hình phù hợp với dự đoán giá cổ phiếu

- **Chương 3:** Các phương pháp tiếp cận

Trình bày các phương pháp học máy và học sâu để giải quyết dự đoán chuỗi thời gian mà chúng em đã nghiên cứu và áp dụng trên bộ dữ liệu. Các phương pháp chúng em đã áp dụng là Long Short-Term Memory, Autoregressive Integrated Moving Average. Đây là các mô hình mang lại kết quả tốt đối với các bài toán dự đoán chuỗi thời gian và được đánh giá là có nhiều ưu điểm hơn trong lĩnh vực này.

- **Chương 4:** Cài đặt, thử nghiệm và đánh giá

Trong chương này, chúng em trình bày các cách đánh giá, các bước cài đặt mô hình dự đoán trên bộ dữ liệu, đồng thời trình bày các kết quả đã đạt được từ các mô hình. Bên cạnh đó, tiến hành so sánh kết quả với nhau trên các bộ dữ liệu.

- **Chương 5:** Kết luận và hướng phát triển

Tổng kết các kết quả đã đạt được và đề xuất các hướng phát triển trong tương lai để cải thiện được hiệu suất của các mô hình cũng như giúp cho việc dự đoán trên bộ dữ liệu đạt được tỉ lệ chính xác cao hơn.

Chương 1 TỔNG QUAN

1.1. Giới thiệu đề tài

Cổ phiếu là sản phẩm tài chính gồm các đặc trưng như rủi ro cao, lợi nhuận cao, giao dịch linh hoạt và được nhiều nhà đầu tư ưa chuộng. Các nhà đầu tư có thể thu được lợi nhuận lớn bằng cách ước tính xu hướng giá cổ phiếu. Tuy nhiên, giá cổ phiếu chịu ảnh hưởng của rất nhiều yếu tố khác nhau. Do đó, việc dự đoán giá cổ phiếu là rất khó khăn và là đề tài được nghiên cứu rất nhiều.

Cơ chế dự đoán giá cổ phiếu là nền tảng cho việc hình thành các chiến lược đầu tư và phát triển các mô hình quản lý rủi ro. Tuy nhiên, giả thuyết thị trường hiệu quả (Efficient Market Hypothesis) tuyên bố rằng việc dự đoán giá cổ phiếu là bất khả thi. Tuy nhiên, sự phát triển nhanh chóng của công nghệ máy tính đã dẫn tới kết quả là có nhiều thuật toán học máy đã và đang được ứng dụng dụng để dự đoán chuyển động của thị trường chứng khoán một cách nhất quán, từ đó ước tính giá trị của nhiều loại tài sản trong tương lai, điển hình trong số đó là giá cổ phiếu của các công ty.

Thị trường chứng khoán và tài chính vốn dĩ có xu hướng không thể đoán trước và thậm chí là phi logic, giống như kết quả của cuộc bỏ phiếu Brexit hoặc cuộc bầu cử Mỹ vừa qua. Do những đặc điểm này, dữ liệu tài chính được cho là sở hữu một cấu trúc khá hỗn loạn và thường khiến cho việc tìm kiếm các mẫu đáng tin cậy trở nên khó khăn. Mô hình hóa các cấu trúc hỗn loạn yêu cầu các thuật toán học máy có khả năng tìm ra các cấu trúc ẩn bên trong dữ liệu và dự đoán chúng sẽ ảnh hưởng như thế nào đến chính nó trong tương lai. Phương pháp hiệu quả nhất để đạt được điều này là Học máy và Học sâu. Học sâu có thể giải quyết các cấu trúc phức tạp một cách dễ dàng và trích xuất các mối quan hệ giúp tăng thêm độ chính xác của các kết quả được tạo ra.

Học máy có khả năng làm cho toàn bộ quá trình dự đoán dễ dàng hơn bằng cách phân tích lượng lớn dữ liệu, phát hiện các mẫu quan trọng và tạo ra một đầu ra duy nhất nhằm định hướng các nhà giao dịch đến một quyết định cụ thể dựa trên giá tài sản dự đoán.

Dữ liệu chính là nguồn tài nguyên vô cùng giá trị trong việc xây dựng mô hình dự đoán. Vì tính đặc thù của chuỗi thời gian liên quan đến số liệu tài chính, chúng ta phải đối mặt khá nhiều thách thức trong việc dự báo giá trị trong tương lai.

Các phương pháp truyền thống được dùng để dự đoán thường dựa trên mô hình thống kê và kinh tế lượng. Tuy nhiên, những mô hình này rất khó để áp dụng với dữ liệu time-series (chuỗi thời gian) bất ổn định.

Những mô hình cũ đã thường được sử dụng là ARIMA (Auto Regressive Integrated Moving Average) và ANN (Artificial Neural Networks) được cho là có khá nhiều nhược điểm bên cạnh những ưu điểm vốn có của nó. Ngoài ra, trong những năm gần đây, các nhà nghiên cứu cho rằng mô hình LSTM (Long Short-Term Memory) có độ chính xác khi dự đoán cao hơn.

Dự báo có thể được định nghĩa là dự đoán về một số sự kiện hoặc sự kiện trong tương lai bằng cách phân tích dữ liệu lịch sử. Nó trải dài trên nhiều lĩnh vực bao gồm kinh doanh và công nghiệp, kinh tế, khoa học môi trường và tài chính. Các bài toán liên quan đến dự báo có thể được phân loại như sau:

• Dự báo ngắn hạn (dự báo cho một số ít giây, phút, ngày, tuần hoặc tháng).

• Dự báo trung hạn (dự báo từ 1 đến 2 năm).

• Dự báo dài hạn (dự đoán sau 2 năm).

1.2. Tính ứng dụng của đề tài

- Phân tích được đặc điểm của bộ dữ liệu.
- Xây dựng mô hình dự đoán giá đóng cửa của từng loại cổ phiếu trong bộ dữ liệu.
- Giúp đưa ra thông tin tham khảo cho nhà đầu tư ra quyết định nên đầu từ vào một trong 4 cổ phiếu của nhóm Big Tech hay không.
- Từ kết quả phân tích khám phá dữ liệu (EDA), ta có thể trích xuất được đặc trưng của các cổ phiếu để nhận biết tỉ lệ rủi ro và lợi nhuận có thể đạt được.

1.3. Kết luận

Như chúng em đã đề cập ở phần giới thiệu đề tài, việc dự đoán được giá tương lai của một cổ phiếu chỉ dựa vào giá của những năm trước là một thử thách và khó khăn rất lớn. Do tính chất dễ biến động và quy luật cung-cầu của thị trường chịu tác động và chi phối của nhiều nhiều yếu tố nên đồ án chỉ nhằm mục đích nghiên cứu và đưa ra thông tin tham khảo là chính.

Chương 2 BỘ DỮ LIỆU NGHIÊN CỨU

2.1. Bộ dữ liệu Stocks Price Data

Bộ dữ liệu Stock Price Data được cung cấp bởi tác giả Pier Paolo Ippolito. Bộ dữ liệu là lịch sử các bản ghi về giá cổ phiếu của nhiều cổ phiếu khác nhau như Apple, Microsoft, Facebook, Tesla và được lưu trữ trong file định dạng .csv. Bộ dữ liệu gồm 19.586 dòng dữ liệu, 8 thuộc tính:

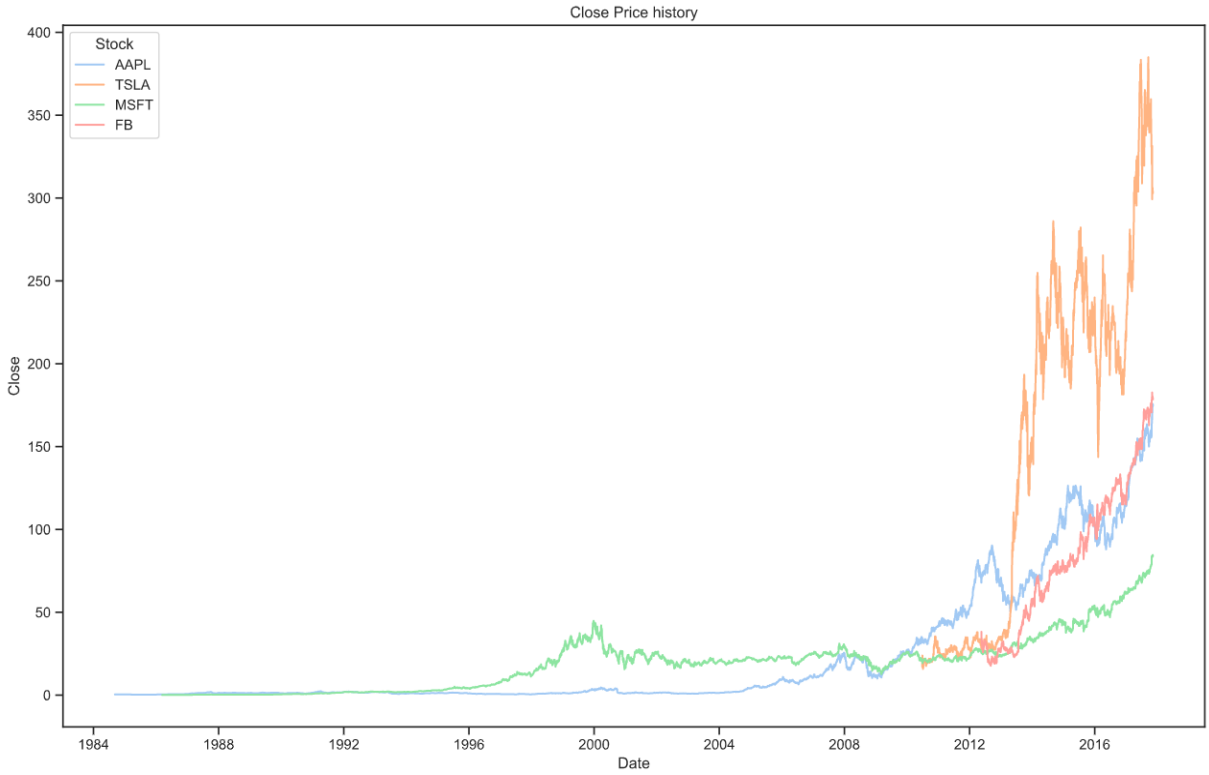
- Ngày giao dịch (Date): là ngày giao dịch mua bán cổ phiếu bắt đầu từ thứ Hai đến thứ 6 hàng tuần, không giao dịch vào ngày Lễ.
- Giá mở cửa (Open): là giá khi bắt đầu phiên giao dịch trong ngày. Giá này có thể biến động hoặc giữ nguyên khi cuối ngày kết thúc
- Giá cao nhất trong ngày (High): là lúc giá đạt đỉnh trong ngày khi các nhà đầu tư tích cực mua vào mã cổ phiếu đó
- Giá thấp nhất trong ngày (Low): là lúc giá đạt đáy trong ngày khi các nhà đầu tư tích cực bán tháo mã cổ phiếu đó
- Giá đóng cửa (Close): là giá cuối cùng trong ngày khi kết thúc phiên giao dịch mua bán cổ phiếu trong ngày.
- Khối lương giao dịch trong ngày (Volume): thể hiện tính thanh khoản trong phiên, cũng như sự quan tâm từ hai phe Long/Short (đầu cơ/bán khống) có trên thị trường.
- Tên loại cổ phiếu (Stock): là mã định danh của một công ty niêm yết trên sàn giao dịch chứng khoán
- Hợp đồng mở (OpenInt): Khối lượng hợp đồng đang giao dịch trên thị trường và duy trì vị thế sang các phiên giao dịch sau đó. Nó cung cấp bức tranh chính xác hơn về hoạt động giao dịch quyền chọn và liệu dòng tiền chảy vào thị trường hợp đồng tương lai và quyền chọn đang tăng hay giảm.

Date	Open	High	Low	Close	Volume	OpenInt	Stock
1984-09-07	0.42388	0.42902	0.41874	0.42388	23220030	0	AAPL
1984-09-10	0.42388	0.42516	0.41366	0.42134	18022532	0	AAPL
1984-09-11	0.42516	0.43668	0.42516	0.42902	42498199	0	AAPL
1984-09-12	0.42902	0.43157	0.41618	0.41618	37125801	0	AAPL
…	…	…	…	…	…	…	…
2012-05-18	42.05	45	38	38.23	580438450	0	FB
2012-05-21	36.53	36.66	33	34.03	169418988	0	FB
2012-05-22	32.61	33.59	30.94	31	101876406	0	FB
2012-05-23	31.37	32.5	31.36	32	73678512	0	FB
…	…	…	…	…	…	…	…
1986-03-13	0.0672	0.07533	0.0672	0.07533	1371330506	0	MSFT
1986-03-14	0.07533	0.07533	0.07533	0.07533	409569463	0	MSFT
1986-03-17	0.07533	0.07533	0.07533	0.07533	176995245	0	MSFT
1986-03-18	0.07533	0.07533	0.07533	0.07533	90067008	0	MSFT
…	…	…	…	…	…	…	…
2010-06-28	17	17	17	17	0	0	TSLA
2010-06-29	19	25	17.54	23.89	18783276	0	TSLA
2010-06-30	25.79	30.42	23.3	23.83	17194394	0	TSLA
2010-07-01	25	25.92	20.27	21.96	8229863	0	TSLA
…	…	…	…	…	…	…	…

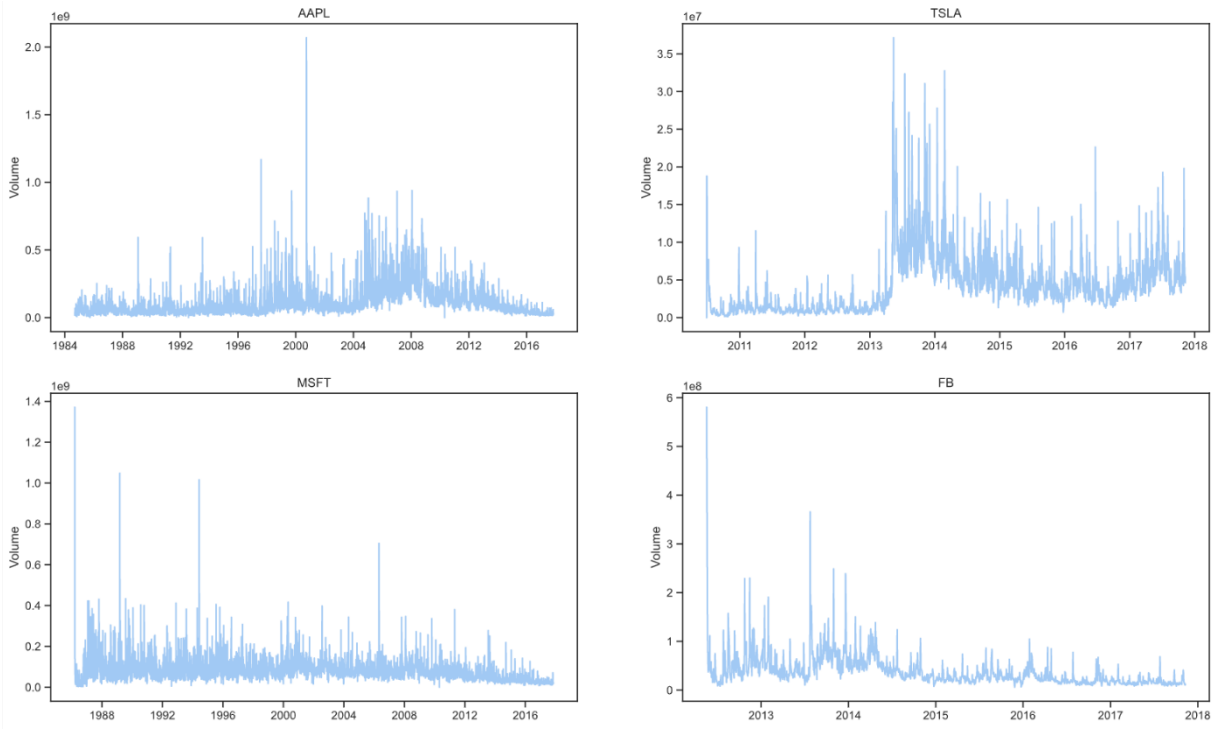
Bảng 2.1: Bộ dữ liệu Stock data

Lịch sử giao dịch của 4 nhóm cổ phiếu như sau:

- Apple: từ ngày 07-09-1984 đến ngày 10-11-2017 tương đương với 8.364 điểm dữ liệu chiếm 42.71% của bộ dữ liệu.
- Tesla: từ ngày 28-06-2010 đến ngày 10-11-2017 tương đương với 1.858 điểm dữ liệu chiếm 9.49% của bộ dữ liệu.
- Microsoft: từ ngày 13-03-1986 đến ngày 10-11-2017 tương đương với 7.983 điểm dữ liệu chiếm 41.76% của bộ dữ liệu.
- Facebook: từ ngày 18-05-2012 đến ngày 10-11-2017 tương đương với 1.381 điểm dữ liệu chiếm 6.04% của bộ dữ liệu.

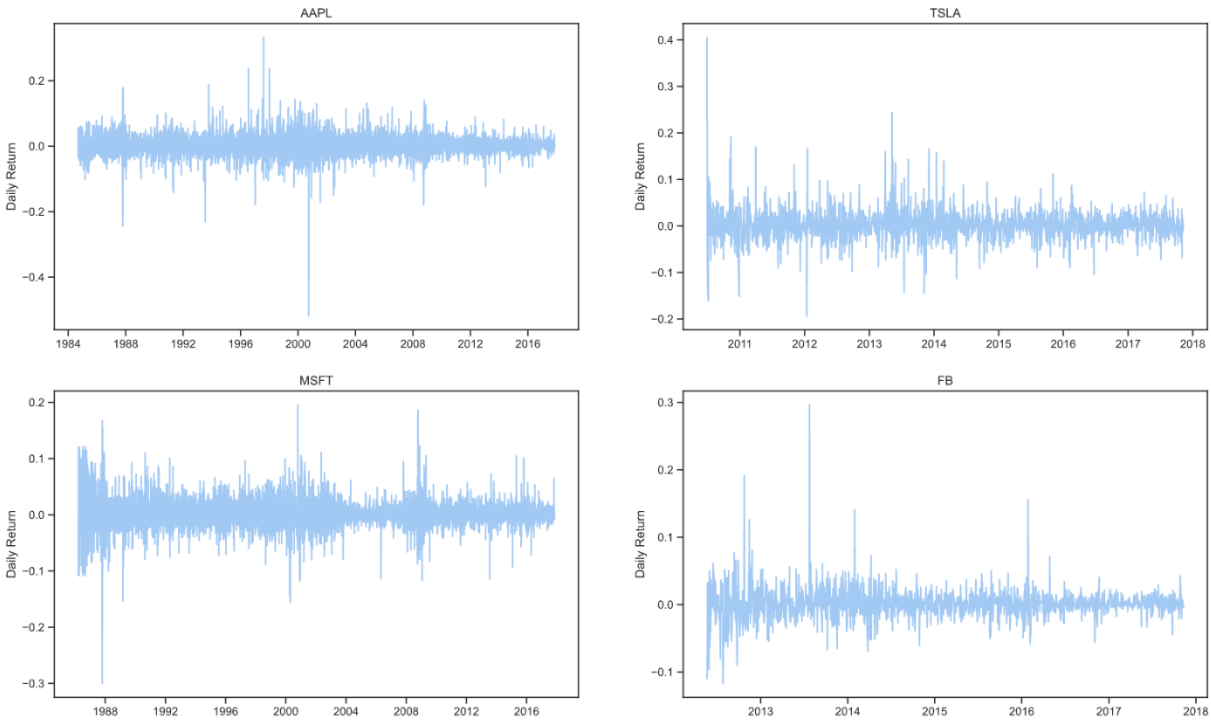


Hình 2.1: Lịch sử giá đóng cửa của 4 mã cổ phiếu Apple (AAPL), Tesla (TSLA), Microsoft (MSFT) và Facebook (FB).



Hình 2.2: Khối lượng giao dịch trong ngày của 4 mã cổ phiếu Apple (AAPL), Tesla (TSLA), Microsoft (MSFT) và Facebook (FB).

Các cổ phiếu của 4 công ty công nghệ có giá đóng cửa có sự biến động giữa giá đóng cửa hiện tại so với ngày trước đó tương tự nhau.

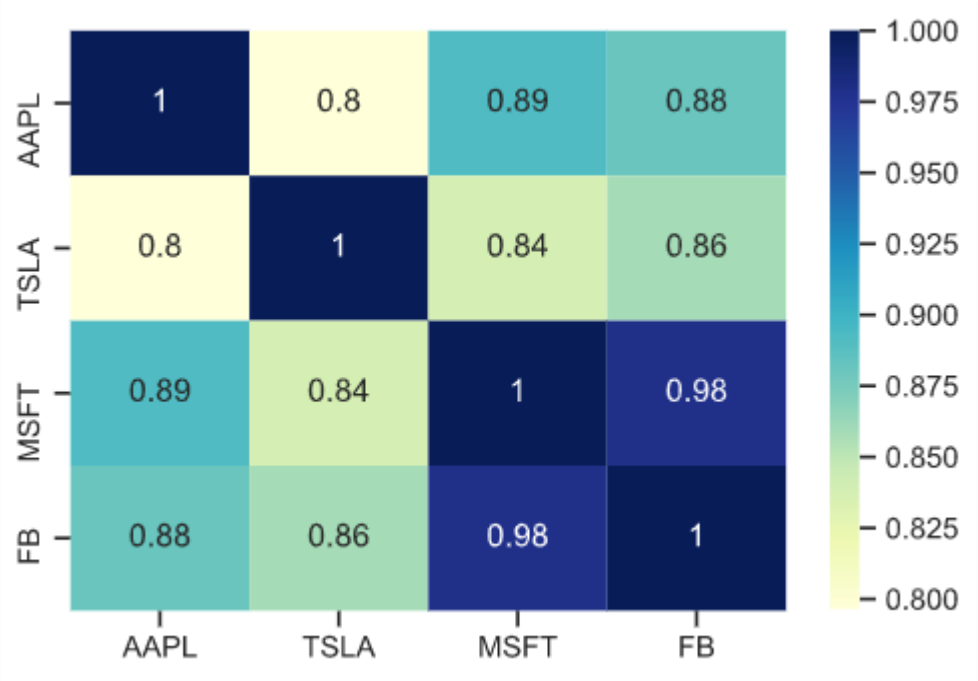


Hình 2.3: Sự biến động giá đóng cửa hiện tại so với ngày trước đó của 4 mã cổ phiếu Apple (AAPL), Tesla (TSLA), Microsoft (MSFT) và Facebook (FB).

2.2. Kết luận

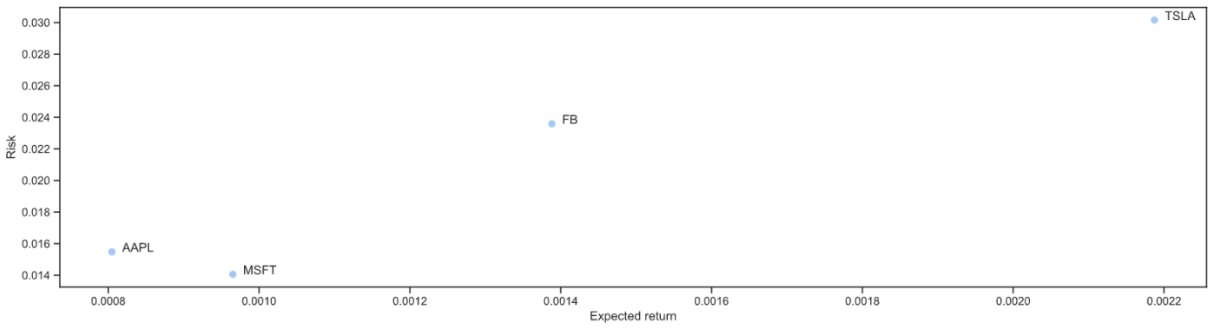
Sự khác nhau giữa số lượng các điểm dữ liệu giữa các mã cổ phiếu cho thấy rằng các công ty công nghệ trẻ niêm yết trên sàn chứng khoán như Facebook, Tesla có giá trị cổ phiếu tăng trưởng mạnh mẽ trong thời gian ngắn. Các cổ phiếu có tuổi đời lâu như Microsoft, Apple tăng trưởng ổn định.

Các mã cổ phiếu trong bộ dữ liệu thuộc nhóm ngành công nghệ nên có sự liên quan với nhau, khi các nhà đầu tư xuống tiền để đầu tư vào một nhóm ngành cụ thể tăng trưởng nhanh nào đó, mà ở đây là nhóm ngành công nghệ.



Hình 2.4: Mối quan hệ giữa 4 mã cổ phiếu Apple (AAPL), Tesla (TSLA), Microsoft (MSFT) và Facebook (FB).

Tuy nhiên, tỉ lệ rủi ro và khả năng sinh lời của các cổ phiếu trên là hoàn toàn không giống nhau. Trong đó, cổ phiếu của Tesla (TSLA) có khả năng sinh lời cao và đi kèm với rủi ro cao hơn hẳn các cổ phiếu khác nằm trong nhóm 4 cổ phiếu đang xét. Cổ phiếu của Microsoft là cổ phiếu có khả năng sinh lời tạm ổn và có mức độ rủi ro thấp nhất trong nhóm 4 cổ phiếu trên.



Hình 2.5: Mối liên hệ tỉ lệ lợi nhuận mong đợi và nguy cơ của Apple (AAPL), Tesla (TSLA), Microsoft (MSFT) và Facebook (FB).

Chương 3 CÁC PHƯƠNG PHÁP TIẾP CẬN

Thông qua việc tìm hiểu và nghiên cứu các công trình về dự đoán chuỗi thời gian (time series), chúng em quyết định cài đặt các phương pháp học máy và học sâu trong bài toán chuỗi thời gian (time series) làm tiền đề, tiếp đó là đề xuất phương pháp mô hình tối ưu nhất của các mô hình đơn để có được mô hình mang lại độ chính xác cao nhất, sở dĩ chúng em ưu tiên sử dụng các phương pháp học sâu vì nó mang lại hiệu quả tốt hơn các phương pháp dự đoán truyền thống. Trong đồ án này, chúng em cài đặt các mô hình như ARIMA và LSTM. Cùng với việc tiền xử lý dữ liệu như phân tách và loại bỏ những thuộc tính không cần thiết.

3.1. Tiền xử lý dữ liệu (Pre-processing)

Chúng em sử dụng một số phương pháp để tiền xử lý cho dữ liệu trước khi làm đầu vào cho các mô hình dự đoán như sau:

- Định dạng (format) dữ liệu "Date" theo đúng định dạng yêu cầu như 1984-09-07. Loại bỏ các dòng dữ liệu trống.
- Chia dữ liệu thành 4 nhóm nhỏ tương ứng với các đối tượng ở "Stock" thành 4 bộ dữ liệu độc lập của các cổ phiếu AAPL, TSLA, FB và MSFT.
- Loại bỏ các thuộc tính không cần thiết như "Open", "Low", "High", "Volume", "OpenInt" và "Stock", chỉ giữ lại thuộc tính quan trọng là "Close" và "Date".
- Sau khi hoàn thành các bước tiền xử lý cơ bản, đối với phương pháp học máy và học sâu, chúng em có những cách xử lý dữ liệu khác nhau. Cụ thể như sau:
    - Đối với phương pháp học máy, chúng em chia dữ liệu theo tỉ lệ 80:20, 80% dữ liệu dùng để huấn luyện (training) và 20% còn lại dùng để kiểm thử (testing).
    - Đối với phương pháp học sâu, chúng em cũng chia tương tự 80:20. Vì đầu vào (input) của mô hình LSTM chỉ nhận các vectơ từ 3 chiều trở

lên (tensor) nên chọn 5 ngày liên tiếp tương ứng với giá cổ phiếu của 1 tuần giao dịch thành 1 điểm dữ liệu X và ngày thứ 6 là Y. Ví dụ:

[43.436000000 00001]	[43.436000000 00001]	[43.436000000 00001]	[43.436000000 00001]	[43.436000000 00001]
[44.179]	[44.179]	[44.179]	[44.179]	[44.179]
[45.019]	[45.019]	[45.019]	[45.019]	[45.019]
[44.877]	[44.877]	[44.877]	[44.877]	[44.877]
…	…	…	…	…

Bảng 3.1: Dữ liệu đầu vào X dạng tensor của LSTM

44.646
44.631
44.126
43.695
43.401
…

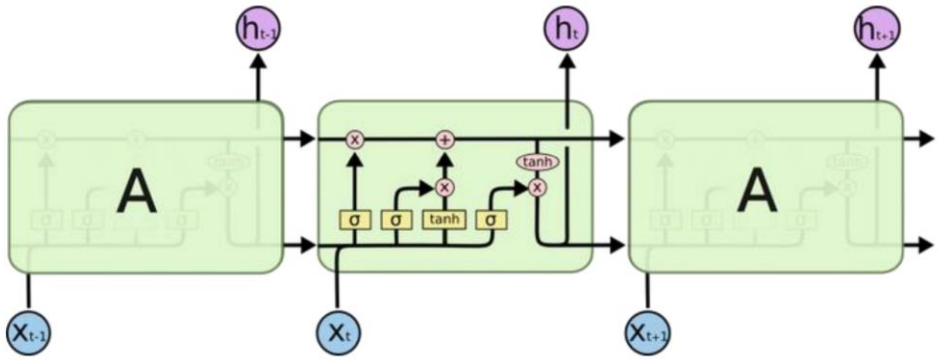
Bảng 3.2: Dữ liệu đầu ra Y của LSTM

3.2. Các phương pháp

3.2.1. Long Short-Term Memory

LSTM là một phương pháp phân loại hiện đại, nó được sử dụng rất nhiều trong các nhiệm vụ phân loại và hầu như luôn đạt được những kết quả tốt. Nó được giới thiệu lần đầu bởi Hochreiter & Schmidhuber (1997). LSTM là một loại mô hình mở rộng của RNN – một mô hình mạng nơ-ron nhân tạo được thiết kế để xử lý các loại dữ liệu có dạng tuần tự, trong mạng RNN trạng thái ẩn tại mỗi bước thời gian sẽ được tính toán

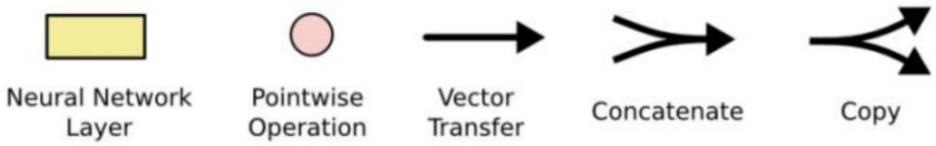
dựa vào dữ liệu đầu vào tại bước thời gian tương ứng và các thông tin có từ bước thời gian trước đó, tạo khả năng ghi nhớ các thông tin đã được tính toán ở những bước thời gian trước của mạng hiện tại. Nó được xem như là mạng nơ-ron nhân tạo phù hợp nhất để nhận dạng các mẫu câu trong chuỗi dữ liệu như văn bản, hình ảnh và các dữ liệu có dạng tuần tự, tuy nhiên việc chỉ có thể nhớ thông tin tại các bước gần nhất là một nhược điểm của cấu trúc mạng RNN, do đó trong cấu trúc mạng RNN thì các những phần tử đầu tiên trong chuỗi đầu vào không mang nhiều ảnh hưởng đến kết quả tính toán dự đoán cho chuỗi đầu ra ở các bước sau của mô hình. Mạng LSTM khắc phục được nhược điểm không có khả năng ghi nhớ thông tin từ các bước có khoảng cách xa.



Hình 3.1: Các mô-đun lặp của mạng LSTM chứa 4 layer

Trong đó, các ký hiệu sử dụng trong mạng LSTM được giải nghĩa như hình 3.1 sau đây:

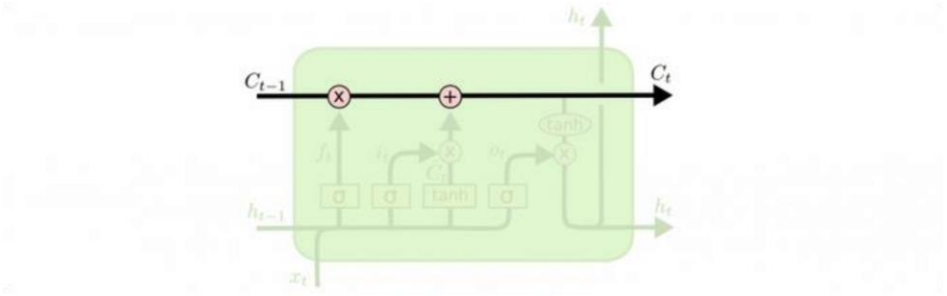
- Hình chữ nhật nền vàng là các lớp ẩn của mạng nơ-ron.
- Hình tròn nền hồng biểu diễn toán tử Pointwise.
- Đường kẻ gộp lại với nhau biểu thị phép nối các toán hạng.
- Và đường rẽ nhánh biểu thị cho sự sao chép từ vị trí này sang vị trí khác.



Hình 3.2: Các kí hiệu sử dụng trong mạng LSTM

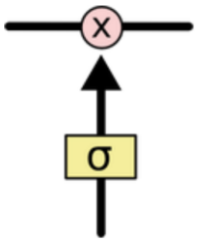
Có lẽ sau khi quan sát mô hình thiết kế của LSTM, chúng ta nhận ra ngay, đây là một bảng mạch số, gồm các mạch logic và các phép toán logic trên đó. Thông tin, hay nói khác hơn là tần số của dòng điện di chuyển trong mạch sẽ được lưu trữ, lan truyền theo cách mà chúng ta thiết kế bảng mạch.

Mấu chốt của LSTM là cell state (tế bào trạng thái), đường kẻ ngang chạy dọc ở trên top diagram. Cell state giống như băng chuyền. Nó chạy xuyên thẳng toàn bộ mắc xích, chỉ một vài tương tác nhỏ tuyến tính (minor linear interaction) được thực hiện. Điều này giúp cho thông tin ít bị thay đổi xuyên suốt quá trình lan truyền.



Hình 3.3: Tế bào trạng thái LSTM giống như một băng truyền

LSTM có khả năng thêm hoặc bớt thông tin vào cell state, được quy định một cách cẩn thận bởi các cấu trúc gọi là cổng (gate). Các cổng này là một cách (tuỳ chọn) để định nghĩa thông tin băng qua. Chúng được tạo bởi hàm sigmoid và một toán tử nhân pointwise.

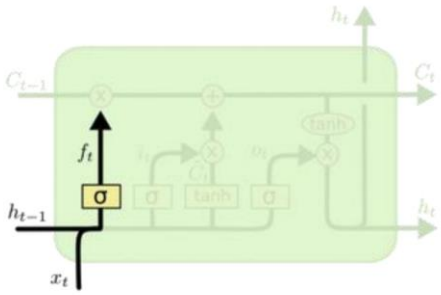


Hình 3.4: Cổng trạng thái LSTM

Hàm kích hoạt Sigmoid có giá trị từ 0 – 1, mô tả độ lớn thông tin được phép truyền qua tại mỗi lớp mạng. Nếu ta thu được zero điều này có nghĩa là "không cho bất kỳ cái gì đi qua", ngược lại nếu thu được giá trị là một thì có nghĩa là "cho phép mọi thứ đi qua". Một LSTM có ba cổng như vậy để bảo vệ và điều khiển cell state.

Quá trình hoạt động của LSTM được thông qua các bước cơ bản sau:

Bước đầu tiên của mô hình LSTM là quyết định xem thông tin nào chúng ta cần loại bỏ khỏi cell state. Tiến trình này được thực hiện thông qua một sigmoid layer gọi là "forget gate layer" – cổng chặn. Đầu vào là h_{t-1} và x_t , đầu ra là một giá trị nằm trong khoảng [0, 1] cho cell state C_{t-1}. 1 tương đương với "giữ lại thông tin", 0 tương đương với "loại bỏ thông tin".

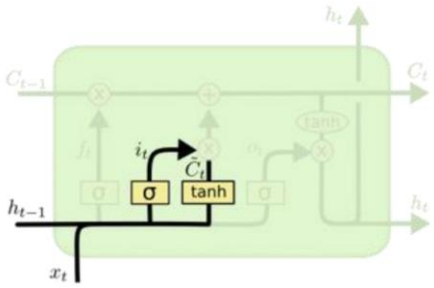


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Hình 3.5: LSTM focus f

Bước tiếp theo, ta cần quyết định thông tin nào cần được lưu lại tại cell state. Ta có hai phần. Một, single sigmoid layer được gọi là "input gate layer" quyết định các giá trị

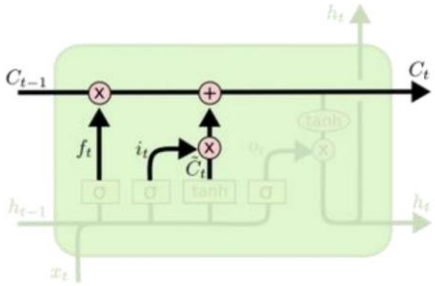
chúng ta sẽ cập nhật. Tiếp theo, một $tanh$ layer tạo ra một vector ứng viên mới, $\tilde{C}_t$ được thêm vào trong ô trạng thái.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hình 3.6: LSTM focus i

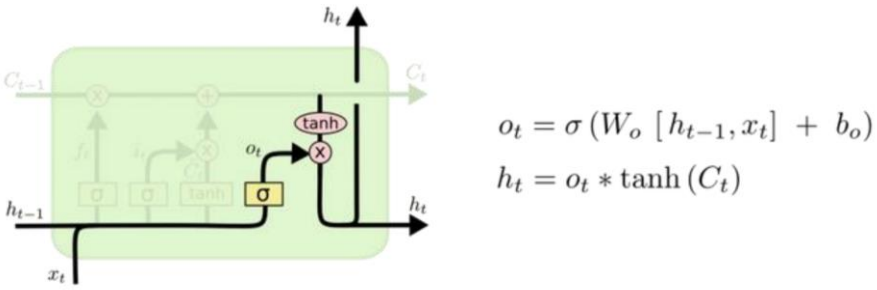
Ở bước tiếp theo, ta sẽ kết hợp hai thành phần này lại để cập nhật vào cell state. Lúc cập nhật vào cell state cũ C_{t-1} vào cell state mới C_t. Ta sẽ đưa state cũ hàm f_t, để quên đi những gì trước đó. Sau đó, ta sẽ thêm $i_t \times C_t$. Đây là giá trị ứng viên mới, co giãn (scale) số lượng giá trị mà ta muốn cập nhật cho mỗi state.



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

Hình 3.7: LSTM focus c

Cuối cùng, ta cần quyết định xem thông tin output là gì. Output này cần dựa trên cell state của chúng ta, nhưng sẽ được lọc bớt thông tin. Đầu tiên, ta sẽ áp dụng single sigmoid layer để quyết định xem phần nào của cell state chúng ta dự định sẽ output. Sau đó, ta sẽ đẩy cell state qua $tanh$ (đẩy giá trị vào khoảng -1 và 1) và nhân với một output sigmoid gate, để giữ lại những phần ta muốn output ra ngoài.



Hình 3.8: LSTM focus o.

Mô hình LSTM là một bước đột phá mà chúng ta đạt được từ mô hình RNN.

3.2.2. Autoregressive Integrated Moving Average Model

- Mô hình ARIMA được giới thiệu bởi George Box và Gwilym Jenkins vào năm 1970. Theo một số nhà nghiên cứu, mô hình ARIMA là một trong những phương pháp phổ biến và được sử dụng rộng rãi nhất cho việc dự báo chuỗi thời gian (time-series).
- Mô hình ARIMA đã chứng tỏ là hiệu quả trong việc đưa ra dự đoán ngắn hạn. Ngoài ra, mặc dù việc sử dụng tương đối đơn giản, mô hình ARIMA vượt trội hơn các mô hình cấu trúc phức tạp khác trong việc dự đoán trong khoản thời gian ngắn hạn.
- Box và Jenkins đã tạo ra một phương pháp ba giai đoạn cho việc lựa chọn mô hình. Các giai đoạn này bao gồm: Identification, Estimation, Diagnostic Checking.
- Đây là một mô hình thuộc lớp mô hình diễn giải một time series nhất định dựa trên kết quả của chính nó trong quá khứ, bao gồm độ trễ (lags), và lỗi dự báo trễ (lagged forecast errors).
- ARIMA là thuật toán dự đoán dựa trên ý tưởng: Thông tin về các giá trị trong quá khứ của time-series có thể được sử dụng để dự đoán các giá trị trong tương

lai. Nó dựa trên khái niệm thống kê về tương quan nối tiếp, trong đó các điểm dữ liệu trong quá khứ có ảnh hưởng đến các điểm dữ liệu trong tương lai.

- ARIMA Model được xác định dựa trên 3 tham số: p, d, q. Trong đó:

  p: Chỉ số của AR.

  d: Số lần khử sai biệt (differencing) cần thiết để làm cho chuỗi thời gian có trạng thái đứng yên (stationarity).

  q: Chỉ số của MA, là số lỗi dự báo bị trễ trong phương trình dự đoán.

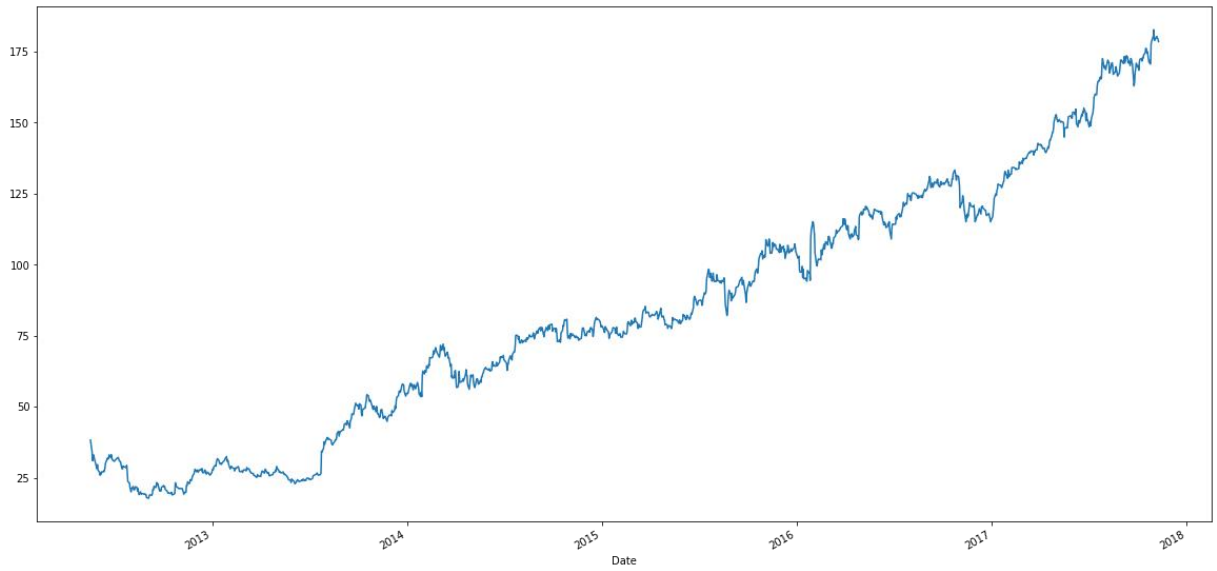
- **AR (Autoregression):** Mô hình hồi quy sử dụng quan hệ phụ thuộc giữa giá trị hiện tại và giá trị quá khứ, đề cập để việc sử dụng các giá trị trong quá khứ trong phương trình hồi quy cho time-series.
- **I (Integration):** Thực hiện thao tác khử sai biệt (differencing) (thiết lập giá trị của 1 điểm dữ liệu = giá trị của chính nó – giá trị điểm dữ liệu trước nó) để làm cho time-series đứng yên.
- **MA (Moving Average):** Mô hình trung bình động tính toán phần dư hoặc sai số của chuỗi thời gian trong quá khứ và tính toán các giá trị hiện tại hoặc tương lai trong time-series.
- ARIMA Model gồm các loại:
  - Non-seasonal ARIMA.
  - Seasonal ARIMA (SARIMA).
  - Seasonal ARIMA with exogenos variables (SARIMAX).

3.2.2.1. Ý nghĩa của các tham số p, d, q (ARIMA(p,d,q))

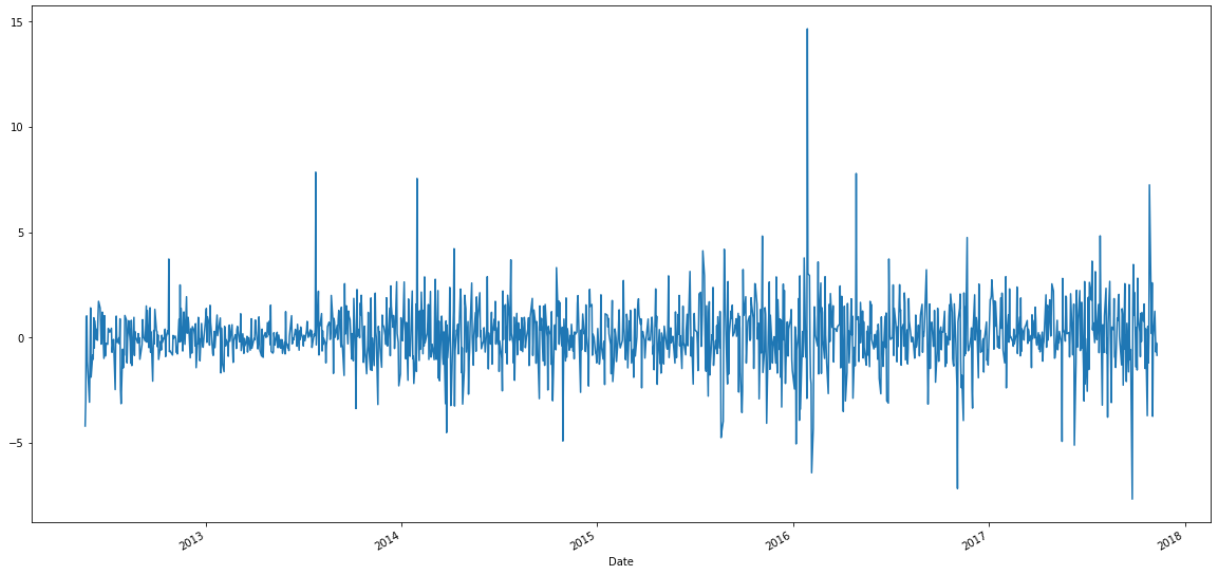
- Tham số p:
  - Là bậc của AR, chỉ số độ trễ dùng để dự đoán.
- Tham số d:
  - ARIMA là mô hình hồi quy tuyến tính sử dụng độ trễ của chính nó để dự đoán. Vì vậy, nó hoạt động tốt nhất khi các yếu tố dự đoán không tương

quan và độc lập với nhau, hay nói cách khác là chúng ta cần time-series đứng yên.

- Một trong những cách để time-series đứng yên chính là khử sai biệt (differencing), tức cho giá trị đang xét bằng giá trị chính nó trừ cho giá trị của điểm dữ liệu trước nó.
- "d" chính là số lần khử sai biệt (differencing) tối thiểu để làm cho time-series đứng yên.



Hình 3.9: Chuỗi thời gian trước khi khử sai biệt (differencing)

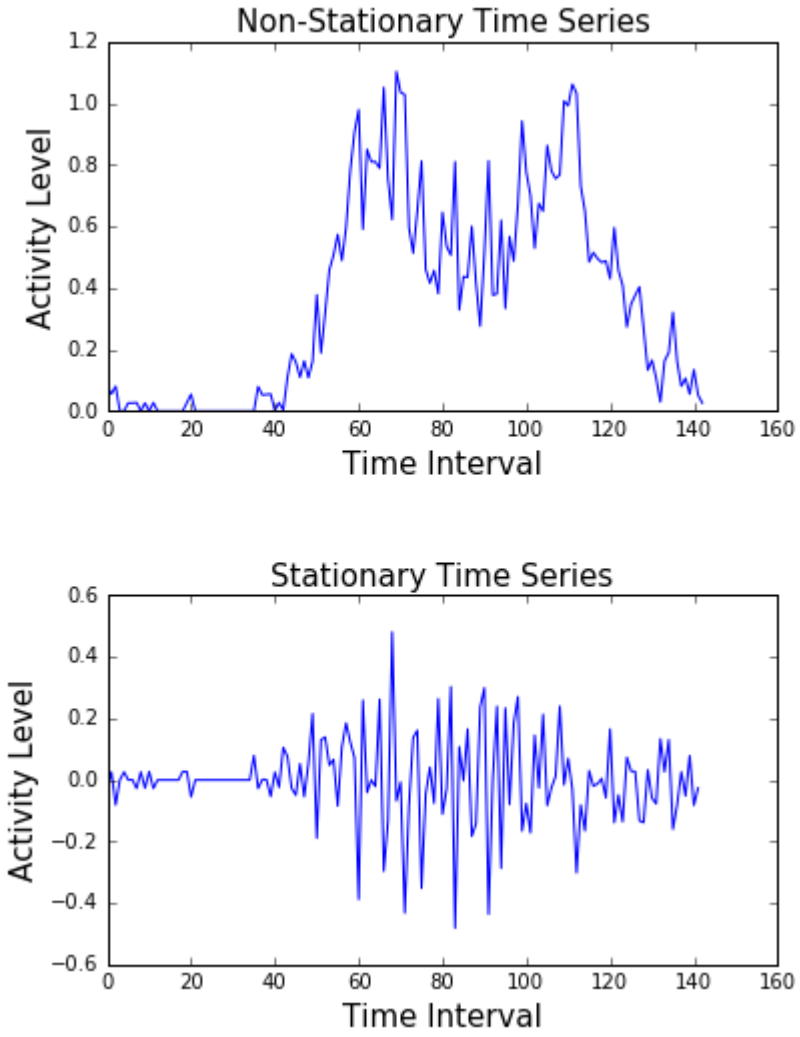


Hình 3.10: Chuỗi thời gian sau khi khử sai biệt (differencing) 1 lần

- Tham số q:
    - "q" chỉ kích thước cửa sổ trung bình động, hay còn gọi là bậc của đường trung bình động. Bậc này chỉ số lượng lỗi trước đó được dùng để đưa ra dự đoán về giá trị của 1 điểm dữ liệu nào đó trong time-series.

3.2.2.2. Khái niệm chuỗi thời gian tĩnh (stationary time-series)

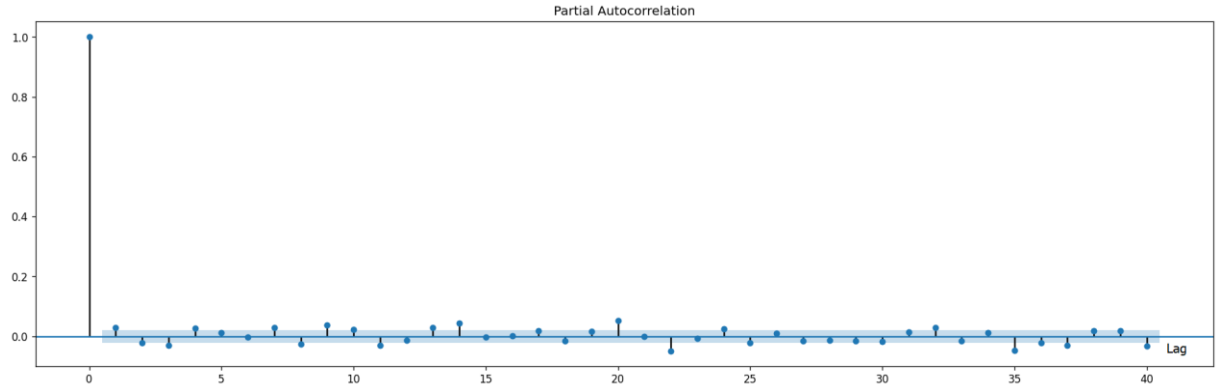
Chuỗi thời gian tĩnh là chuỗi có các đặc tính về thống kê như giá trị trung bình, phương sai, tự tương quan, v.v. đều không đổi theo thời gian. Hầu hết các phương pháp dự báo thống kê đều dựa trên giả định rằng chuỗi thời gian có thể được hiển thị gần như đứng yên (tức là "cố định") thông qua việc sử dụng các phép biến đổi toán học (logarit, diff,…)



Hình 3.11: Minh hoạ chuỗi thời gian tĩnh và chuỗi thời gian không tĩnh

3.2.2.3. Xác định đối số p bằng biểu đồ PACF

- Biểu đồ PACF biểu diễn các hệ số tương quan một phần (partial correlation coefficients) giữa chuỗi thời gian và độ trễ của chính nó.
- Dựa vào biểu đồ, ta có thể quan sát thấy thứ tự của các lag vượt qua ngưỡng quan trọng, đó chính là giá trị của tham số p.



Hình 3.12: Biểu đồ PACF của một chuỗi thời gian

- Trong trường hợp của biểu đồ trên, giá trị của p có thể chọn là 1 hoặc 3.

3.2.2.4. Hàm ndiffs

Trong Python, thư viện pmdarima cung cấp hàm ndiffs nhằm tìm ra đối số d cho mô hình ARIMA. Với đầu vào là mảng các giá trị của chuỗi thời gian, hàm trả về giá trị của đối số d (differencing).

3.3. Kết luận

Trong đồ án này, chúng em đã nghiên cứu và áp dụng các phương pháp học máy và học sâu cho bộ dữ liệu giá chứng khoán. Sau quá trình cài đặt và chạy thử nghiệm, chúng em đã so sánh hiệu suất phân loại của các mô hình với nhau và rút ra một số kinh nghiệm cho bản thân, từ đó thực hiện các phương pháp để giúp cải thiện kết quả của các mô hình. Nghiên cứu, cài đặt các mô hình học sâu thì không thể phủ nhận được kết quả mà nó mang lại đối với bộ dữ liệu. Rõ ràng khi chọn các khoảng thời gian time_slice để tạo thành 1 điểm dữ liệu dạng vectơ 3 chiều (tensor) thì cần chọn để thể hiện được ý nghĩa như 1 tuần (5 ngày), 1 tháng (20 ngày), 3 tháng (60 ngày),….

Chương 4 CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ

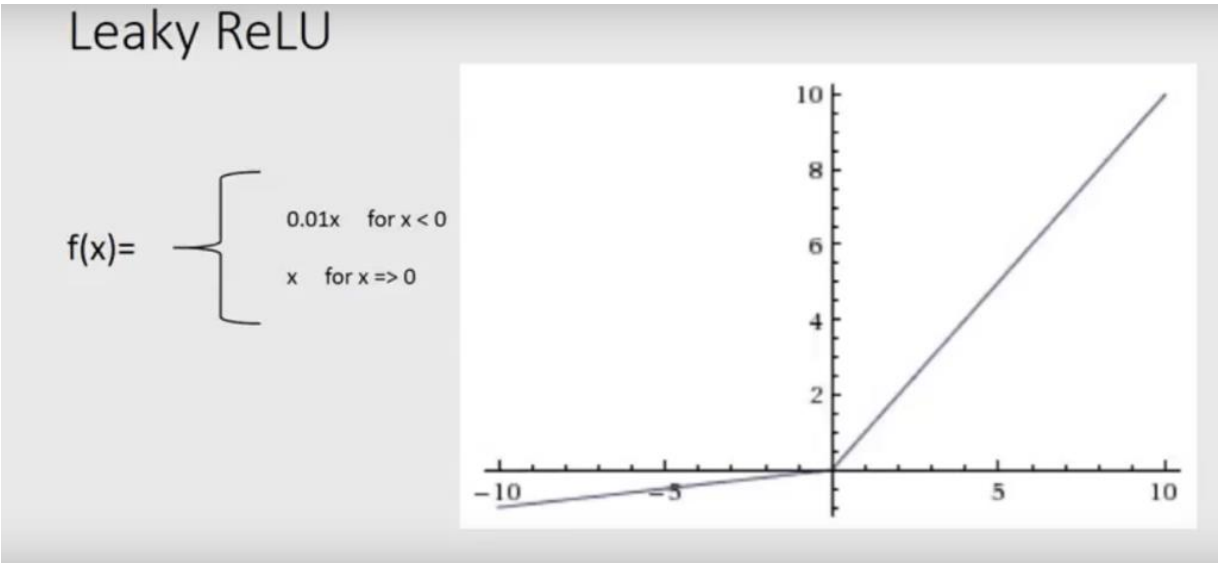
Trong chương này, chúng em sẽ tiến hành cài đặt theo các bước đã giới thiệu trong kiến trúc tổng quan của mô hình, sau khi có kết quả của các thử nghiệm, chúng em tiến hành phân tích kết quả của các mô hình tốt nhất tương ứng cho bộ dữ liệu.

4.1. Cài đặt, thử nghiệm

Trong đồ án này, chúng em cài đặt các mô hình thông dụng hay được sử dụng trong dự đoán giá trị trong chuỗi thời gian như: LSTM, ARIMA trên bộ dữ liệu.

4.1.1. Long Short-Term Memory

Đối với mô hình học sâu LSTM, chúng em thiết kế 6 layer, trong đó có 2 LSTM layer và 4 Dense layer. Ở LSTM layer đầu tiên, chúng em tạo ra 200 node với input shape có dạng vectơ (5, 1), gán tham số return_sequences = True để trả về kết quả output cuối cùng của layer và hàm kích hoạt. Ở LSTM layer thứ 2 ta chỉ truyền vào số node. Dense layer là 1 layer được sử dụng như một layer neural network bình thường dùng để kết nối các node của layer trước với node của layer hiện tại. Tất cả các layer đều dùng hàm kích hoạt Leaky ReLU thuộc module nn của thư viện TensorFlow.



Hình 4.1: Hàm kích hoạt Leaky ReLU

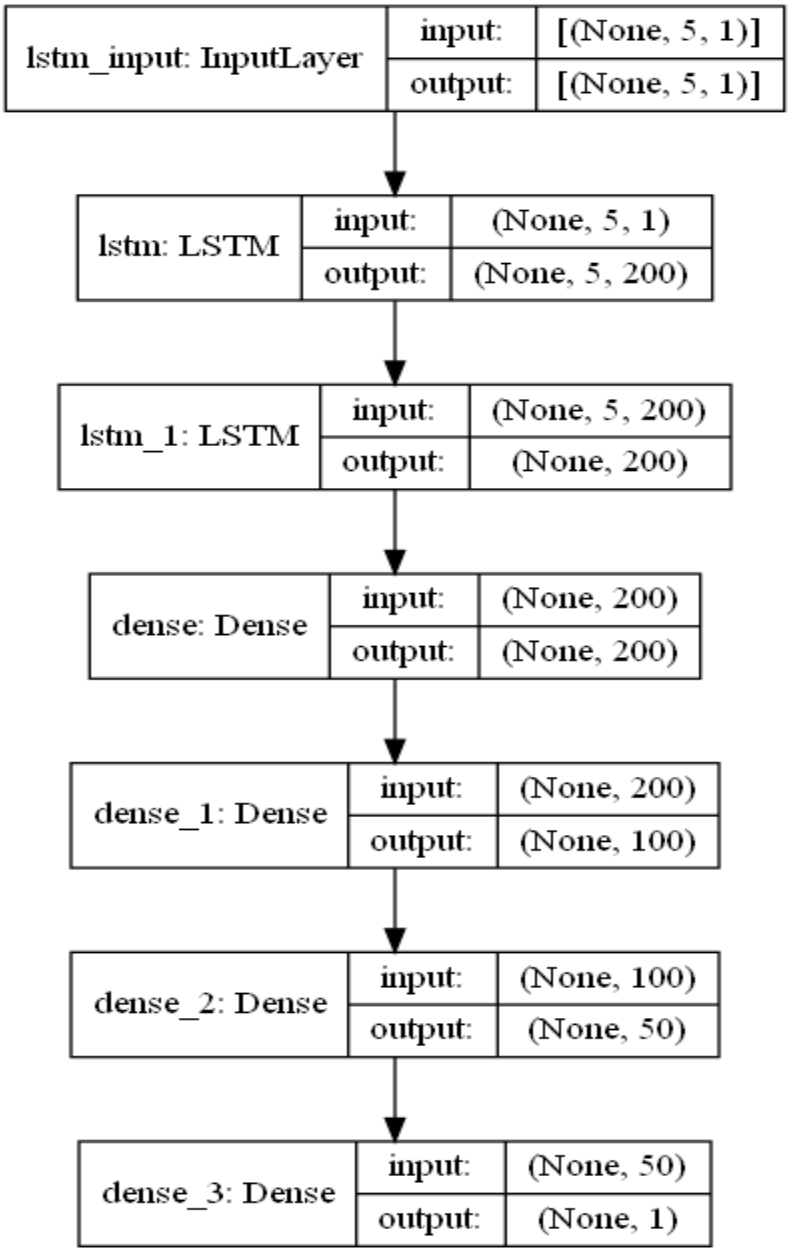
Hàm Leaky ReLU có các điểm tốt của hàm ReLU và giải quyết được vấn đề Dying ReLU bằng cách xét một độ dốc nhỏ cho các giá trị âm thay vì để giá trị là 0.

Bảng 4.1 giúp chúng ta tổng hợp lại model xem model có bao nhiêu layer, tổng số tham số bao nhiêu, shape của mỗi layer..

Layer (type)	Output Shape	Parameters
lstm (LSTM)	(None, 5, 200)	161600
lstm_1 (LSTM)	(None, 200)	320800
dense (Dense)	(None, 200)	40200
dense_1 (Dense)	(None, 100)	20100
dense_2 (Dense)	(None, 50)	5050
dense_3 (Dense)	(None, 1)	51
Total params: 547,801 Trainable params: 547,801 Non-trainable params: 0		

Bảng 4.1: Tóm tắt mô hình LSTM

Hình 4.1 cho chúng ta thấy được kiến trúc tổng thể của mô hình với InputLayer của lstm_input và output của Dense layer cuối cùng



Hình 4.2: Cấu trúc của mô hình LSTM

Để hạn chế tình trạng vanishing gradient, ta điều chỉnh learning rate cho mô hình. Số lượng trọng số được cập nhật trong quá trình huấn luyện gọi là step size hoặc learning rate

Cụ thể, learning rate là một siêu tham số có thể cấu hình được sử dụng trong việc huấn luyện mạng nơ-ron có giá trị dương nhỏ, thường nằm trong khoảng từ 10^{-6} đến 1.0. Trong quá trình huấn luyện, sự lan truyền ngược của lỗi ước tính mà trọng số của một nút trong mạng phải chịu trách nhiệm. Thay vì cập nhật trọng số với số lượng đầy đủ, nó được chia tỷ lệ theo learning rate.

Điều này có nghĩa là tốc độ học 0.1, một giá trị mặc định phổ biến, có nghĩa là các trọng số trong mạng được cập nhật bằng 0.1 × (ước tính trọng số lỗi) hoặc 10% trọng số ước tính lỗi mỗi khi các trọng số được cập nhật.

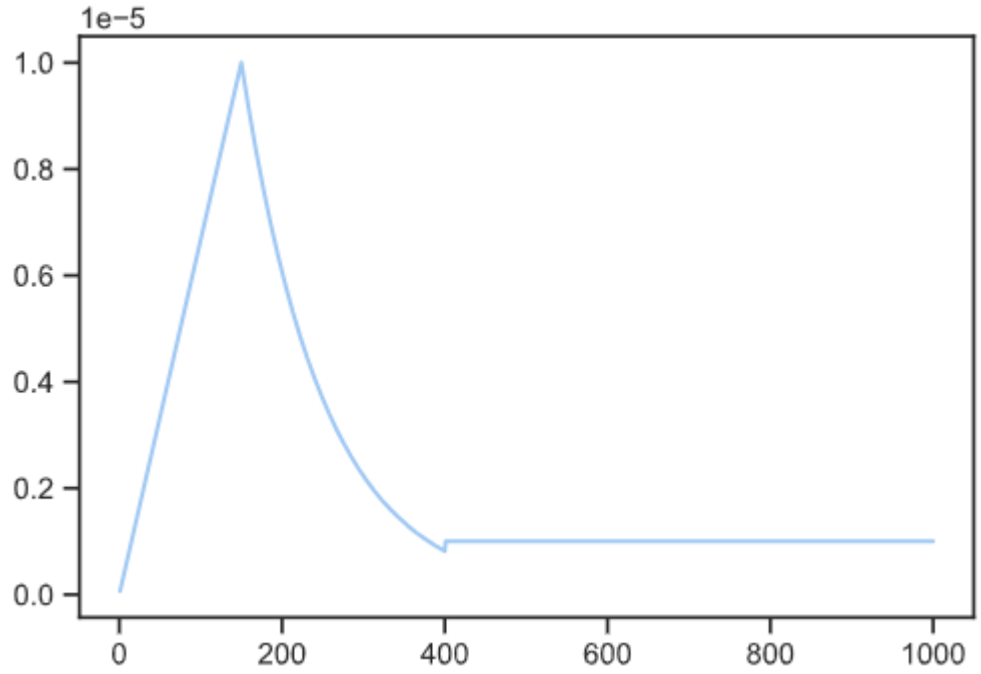
Với learning rate được cấu hình hoàn hảo, mô hình sẽ học cách gần đúng nhất chức năng được cung cấp các tài nguyên có sẵn (số lớp và số nút trên mỗi lớp) trong một số epochs huấn luyện nhất định.

Ở mức cực đoan, learning rate quá lớn sẽ dẫn đến việc cập nhật trọng số sẽ quá lớn và hiệu suất của mô hình (chẳng hạn như mất mát trên tập dữ liệu huấn luyện) sẽ dao động trong các epochs huấn luyện. Hiệu suất dao động được cho là do các trọng số phân kỳ gây ra. Learning rate quá nhỏ có thể không bao giờ hội tụ hoặc có thể gặp khó khăn trong một giải pháp không tối ưu.

Trong trường hợp xấu nhất, việc cập nhật trọng số quá lớn có thể khiến trọng số bị phá huỷ (tức là dẫn đến tràn số).

Phạm vi giá trị cần xem xét cho learning rate là nhỏ hơn 1.0 và lớn hơn 10^{-6}

Nhìn chung, learning rate nhỏ hơn sẽ đòi hỏi nhiều epoch huấn luyện hơn. Ngược lại, learning rate lớn hơn sẽ yêu cầu thời gian huấn luyện ít hơn. Hơn nữa, batch-size nhỏ hơn phù hợp hơn với learning rate nhỏ hơn với ước tính nhiễu của gradient lỗi.



Hình 4.3: Learning rate của từng khoảng epochs

Vì vậy, chúng em tinh chỉnh learning rate để giúp mô hình huấn luyện hiệu quả hơn

```
def scheduler(epoch):

  if epoch <= 150:
    lrate = (10 ** -5) * (epoch / 150)
  elif epoch <= 400:
    initial_lrate = (10 ** -5)
    k = 0.01
    lrate = initial_lrate * math.exp(-k * (epoch - 150))
  else:
    lrate = (10 ** -6)

  return lrate
```

Hình 4.4: Code tuỳ biến learning rate trong LSTM

Sau khi build model xong thì compile nó có tác dụng biên tập lại toàn bộ model của chúng ta đã build. Ở đây chúng ta có thể chọn các tham số để training model như: thuật toán huấn luyện thông qua tham số optimizer, hàm loss của mô hình chúng ta có thể sử dụng mặc định hoặc tự xây dưng thông qua tham số loss, chọn metrics hiển thị khi mô

hình được huấn luyện. Ở đồ án này, chúng em sử dụng hàm Adam để thực hiện optimizer.

Hàm Adam là sự kết hợp của Momentum và RMSprop . Nếu giải thích theo hiện tượng vật lí thì Momentum giống như 1 quả cầu lao xuống dốc, còn Adam như 1 quả cầu rất nặng có ma sát, vì vậy nó dễ dàng vượt qua local minimum tới global minimum và khi tới global minimum nó không mất nhiều thời gian dao động qua lại quanh đích vì nó có ma sát nên dễ dừng lại hơn.

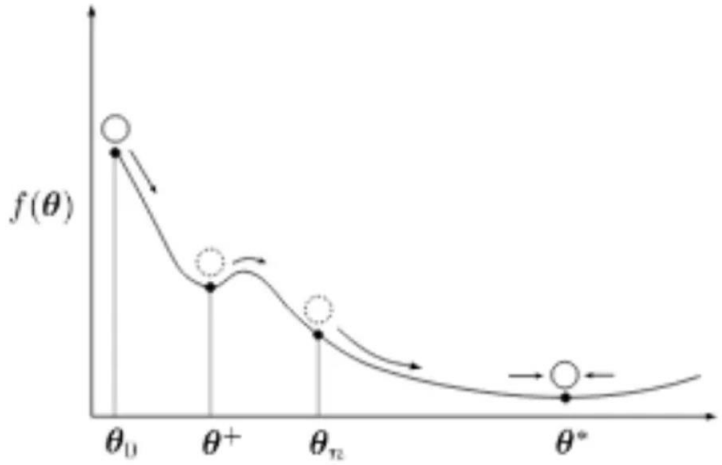


Figure 2: Heavy Ball with Friction, where the ball with mass overshoots the local minimum θ^+ and settles at the flat minimum θ^*.

Hình 4.5: Minh hoạ hàm Adam trong optimizer

Công thức:

$$
\begin{aligned}
\boldsymbol{g}_n &\longleftarrow \nabla f(\boldsymbol{\theta}_{n-1}) \\
\boldsymbol{m}_n &\longleftarrow (\beta_1/(1-\beta_1^n))\,\boldsymbol{m}_{n-1} + ((1-\beta_1)/(1-\beta_1^n))\,\boldsymbol{g}_n \\
\boldsymbol{v}_n &\longleftarrow (\beta_2/(1-\beta_2^n))\,\boldsymbol{v}_{n-1} + ((1-\beta_2)/(1-\beta_2^n))\,\boldsymbol{g}_n \odot \boldsymbol{g}_n \\
\boldsymbol{\theta}_n &\longleftarrow \boldsymbol{\theta}_{n-1} - a\,\boldsymbol{m}_n/(\sqrt{\boldsymbol{v}_n} + \epsilon)\,,
\end{aligned}
$$

Hình 4.6: Công thức hàm Adam

Ở đây, chúng ta sử dụng độ đo mean squared error làm hàm loss và tf.keras.metrics.RootMeanSquaredError() làm hàm metrics để trực quan hoá hiệu suất huấn luyện của mô hình

Sau khi đã complie xong mô hình hình, chúng ta tiến hành gọi thực hiện hàm fit để đưa dữ liệu vào huấn luyện để tìm tham số mô hình (tương tự như sklearn). Với việc gán epochs = 400 sẽ giúp mô hình cập nhật được các learning rate qua các khoảng epochs, batch-size = 64 để đưa qua 1 node của layer theo lô 64 điểm dữ liệu, gán dữ liệu kiểm thử vào validation_data để kiểm tra loss và metrics ngay trên dữ liệu kiểm thử. Khi mô hình chúng ta lớn, có khi huấn luyện thì gặp sự cố ta muốn lưu lại mô hình để chạy lại thì callback giúp ta làm điều này. Nhưng ở đây ta sử dụng callback để cập nhật learning rate cho mô hình. Vào đầu mỗi epoch, lệnh callback này nhận giá trị learning rate được cập nhật từ hàm lập lịch cung cấp, với learning rate hiện tại và thời điểm hiện tại, đồng thời áp dụng learning rate cập nhật trên trình tối ưu hóa.

4.1.2. Autoregressive Integrated Moving Average Model

Đối với mô hình ARIMA, ta có bảng tóm tắt kết quả mô hình của mỗi mô hình được huấn luyện. Em sẽ chọn minh hoạ bảng tóm tắt của mô hình FB.

ARIMA RESULT	
Model	ARIMA(3,1,0)
No.Observations	1379

Bảng 4.2: Minh họa khái quát loại model và số điểm dữ liệu được đưa vào model

	coeff	P>|z|
AR1	0.0181	0.501
AR2	-0.0174	0.520

AR3	-0.0751	0.005

Bảng 4.3: Một số thông tin quan trọng trong kết quả mô hình

Số điểm dữ liệu được mô hình quan sát là 1739, ít hơn số điểm dữ liệu của cả tập dữ liệu FB (gồm 1381 điểm dữ liệu) là 3, do kết quả dự đoán của một điểm dữ liệu được dựa trên giá trị của 3 điểm dữ liệu trước đó (bậc của AR là 3). Cột P>|z| chỉ giá trị p của từng hệ số (coeff). Giá trị p của hệ số ứng với AR=3 là 0.005, nhỏ hơn ngưỡng tin cậy là 0.05 (theo phép kiểm tra Augmented Dickey–Fuller). Qua đó, cho thấy AR = 3 là giá trị tốt cho mô hình.

4.2. Phương pháp đánh giá

Chúng em sử dụng phương pháp đánh giá độ chính xác của mô hình khi huấn luyện trong bộ dữ liệu bằng các độ đo là RMSE (sai số bình phương trung bình gốc), MAPE (sai số phần trăm trung bình tuyệt đối).

Trong đó:

- RMSE (sai số bình phương trung bình gốc):

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

Trong đó: $\hat{y}_i$ là giá trị dự đoán, y_i là giá trị thật, n là số lượng mẫu.

- MAPE(sai số phần trăm trung bình tuyệt đối):

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

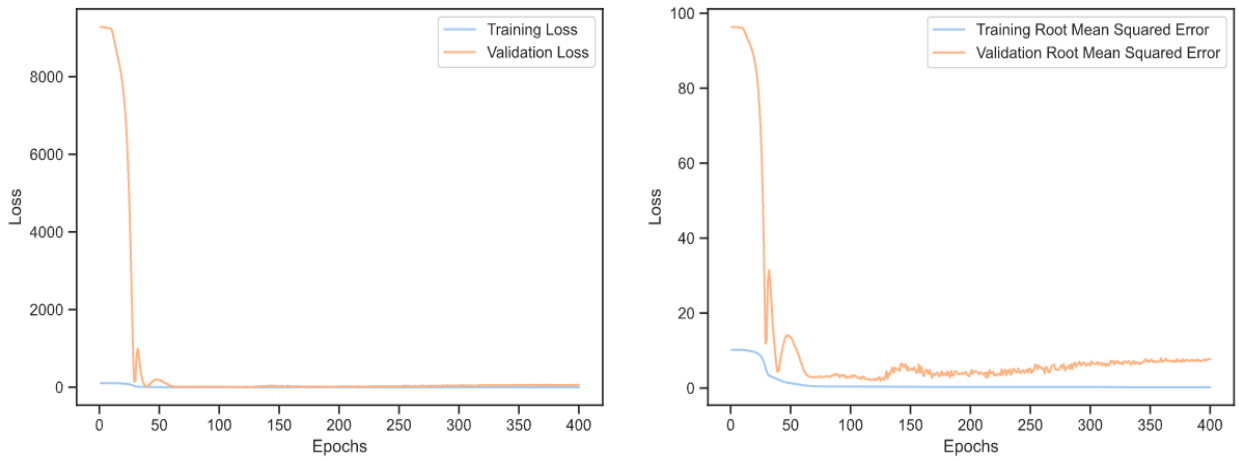
Trong đó: $\hat{y}_i$ là giá trị dự đoán, y_i là giá trị thật, n là số lượng mẫu.

4.3. Kết quả thử nghiệm và đánh giá

	FB	MSFT	TSLA	AAPL	Average
RMSE	2.85502153	1.064712786	6.702129751	7.658400514	4.570066145
MAPE	0.015065904	0.015580243	0.01755862	0.054227389	0.025608039

Bảng 4.4: Kết quả thử nghiệm mô hình LSTM trên bộ dữ liệu

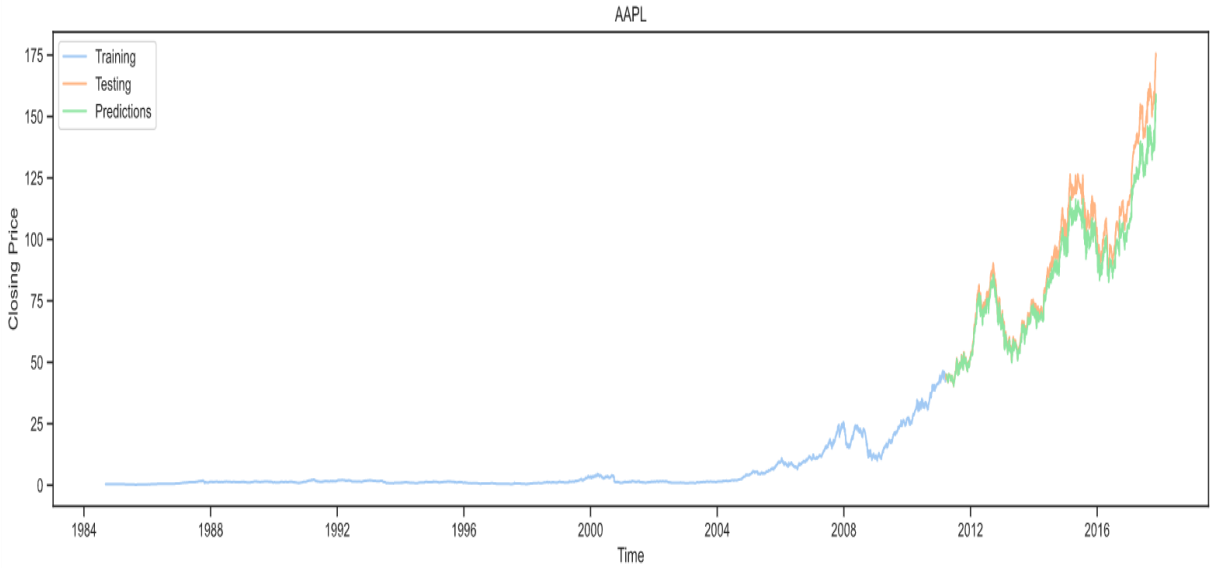
Bảng trên minh hoạ kết quả thử nghiệm mà chúng em đạt được khi cài đặt mô hình LSTM, các kết quả được đánh giá dựa vào độ đo RMSE và MAPE để thuận tiện cho việc so sánh, đối chiếu giữa 2 mô hình. Các thử nghiệm áp dụng với từng bộ dữ liệu được thực hiện trên dữ liệu đã trải qua tiền xử lý. Mô hình dự đoán giá cổ phiếu của AAPL và TSLA có RMSE, MAPE cao hơn 2 mô hình còn lại do dữ liệu huấn luyện có ít sự giao động hoặc tăng/giảm không thể hiện rõ hơn so với dữ liệu kiểm thử nên dẫn đến có các chỉ số lỗi cao hơn.



Hình 4.7: Kết quả loss trên tập train, test của mô hình LSTM trên tập dữ liệu AAPL

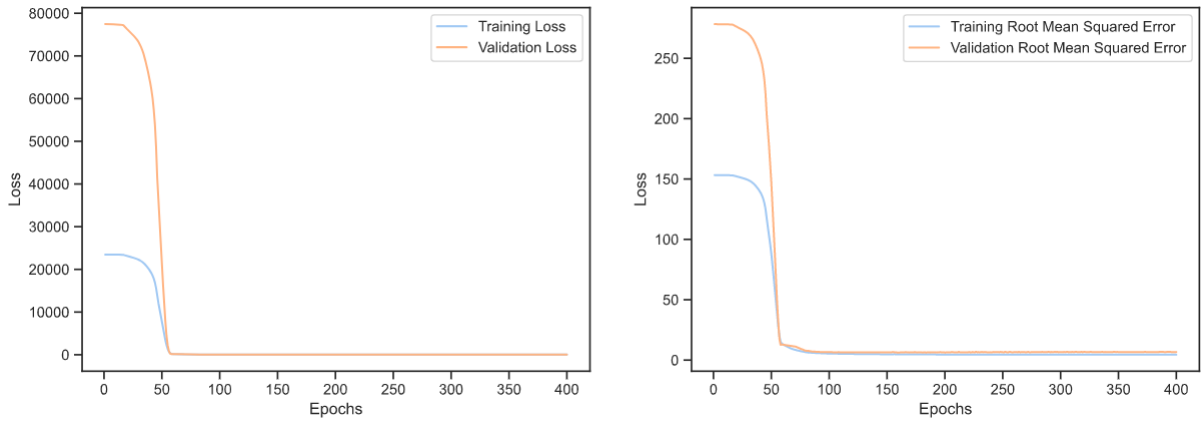
Hình trên minh hoạ kết quả lỗi của mô hình khi huấn luyện trên tập dữ liệu (training) và tập dữ liệu kiểm thử khi đưa tất cả 2 tập dữ liệu qua 400 Epochs. Ta nhận

thấy rõ rằng đối với chỉ số lỗi Loss (MSE) giảm rõ rệt, từ epoch 60 trở đi lỗi trên tập huấn luyện và kiểm thử của mô hình tương đương nhau, tiến gần về 0. Đối với độ đo RMSE, kết quả tiến dần về 0 nhanh hơn trên tập huấn luyện, tập kiểm thử lỗi giảm đạt đến giá trị nhỏ nhất trong đoạn epoch [100; 140] có xu hướng tiến dần về 0, sau đó tăng lên theo epochs.

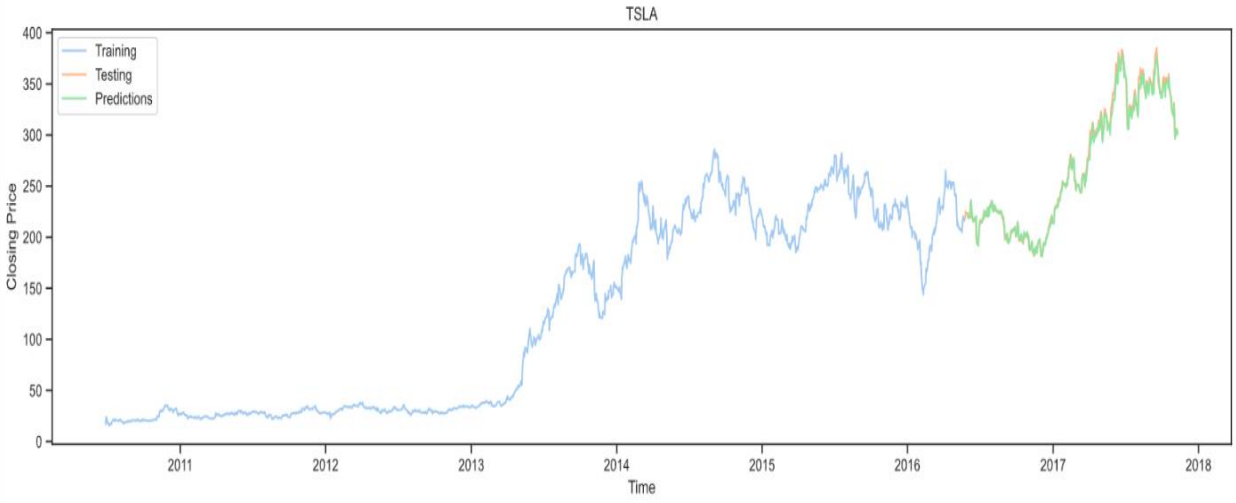


Hình 4.8: Kết quả dự đoán của mô hình LSTM trên tập dữ liệu AAPL

Hình trên cho thấy mô hình dự đoán được xu hướng tăng trưởng của cổ phiếu AAPL. Tuy nhiên, do tập dữ liệu huấn luyện không cân bằng mà tập dữ liệu huấn luyện lại phức tạp nên khi dự đoán giá đóng cửa có sự sai lệch nhiều so với giá thực tế.

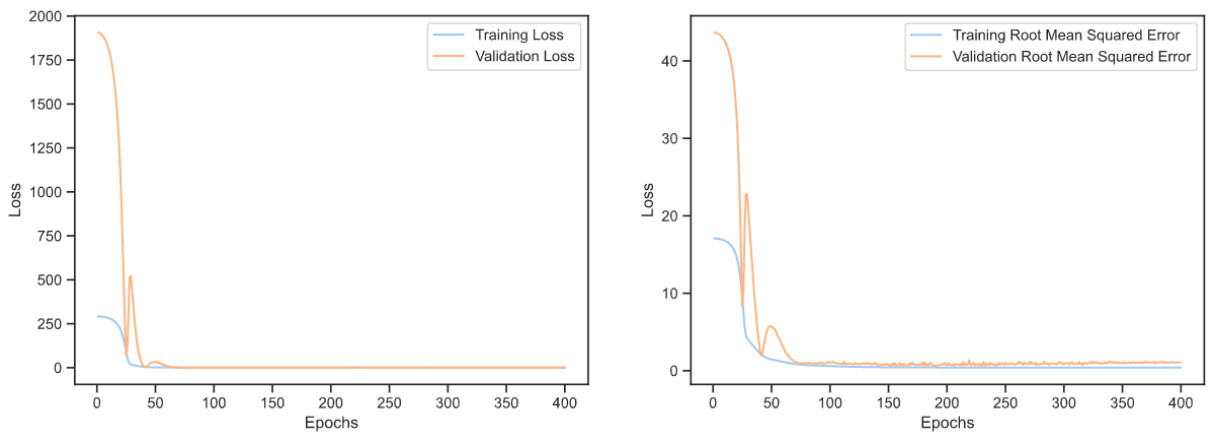


Hình 4.9: Kết quả loss trên tập train, test của mô hình LSTM trên tập dữ liệu TSLA

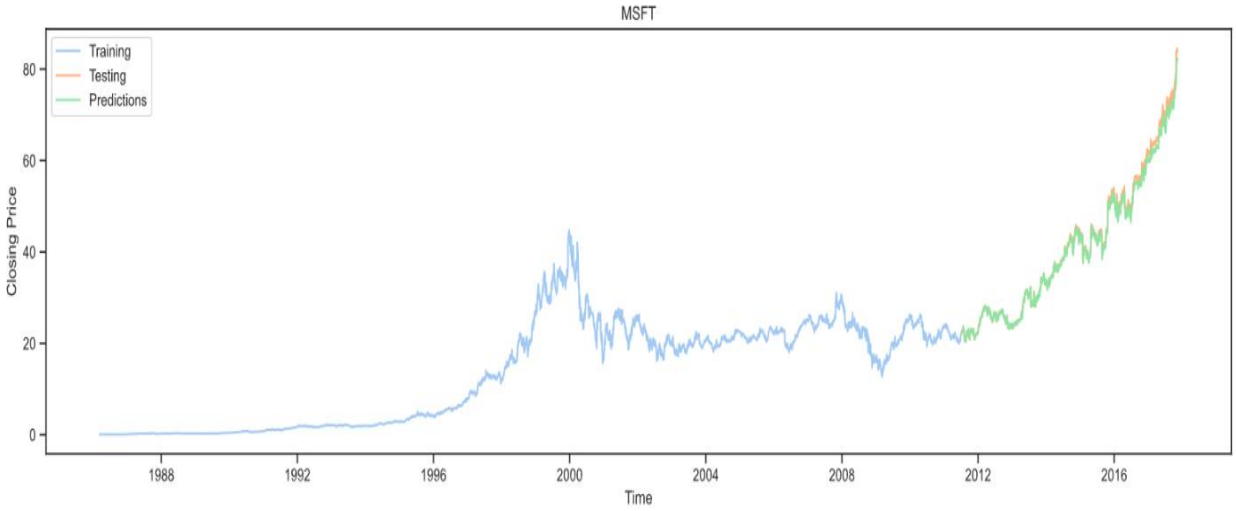


Hình 4.10: Kết quả dự đoán của mô hình LSTM trên tập dữ liệu TSLA

Khi mô hình được huấn luyện trên tập dữ liệu TSLA, giá đóng cửa dự đoán có độ lệch so với giá thực tế không nhiều. Ta thấy rằng, mô hình dự gần chính xác do tập dữ liệu liệu huấn luyện có sự phân bố đa dạng nên khi dự đoán trên tập dữ liệu kiểm thử cho kết quả tốt hơn hẳn.

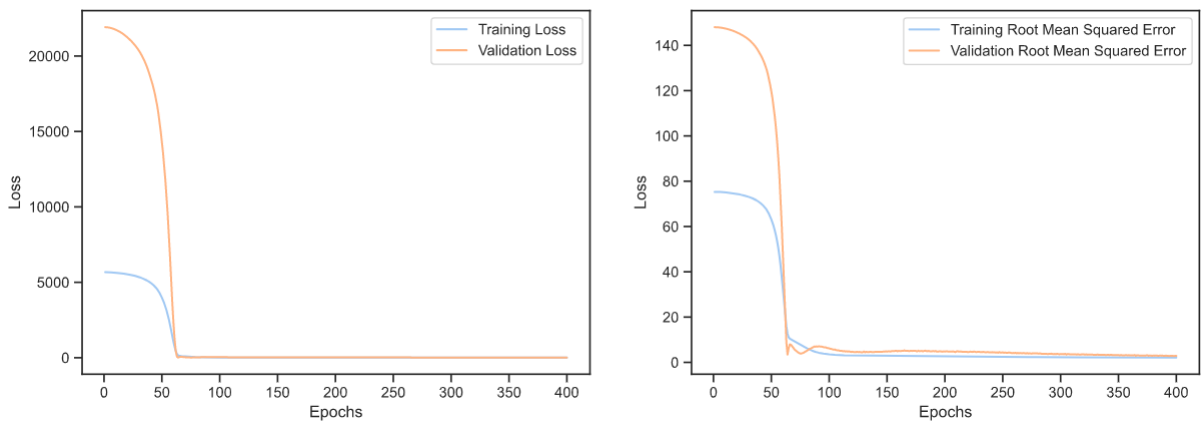


Hình 4.11: Kết quả loss trên tập train, test của mô hình LSTM trên tập dữ liệu MSFT

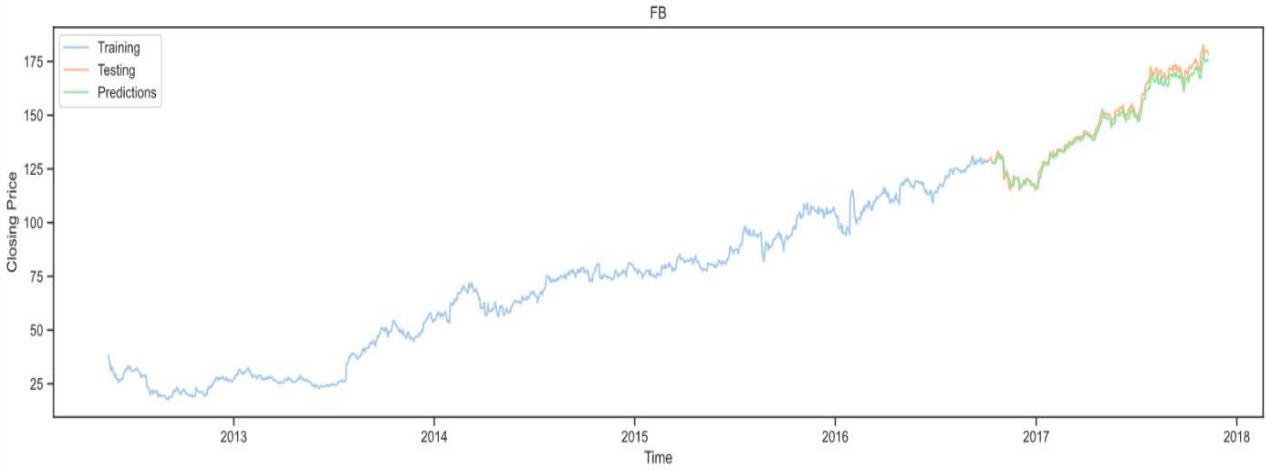


Hình 4.12: Kết quả dự đoán của mô hình LSTM trên tập dữ liệu MSFT

Trên tập dữ liệu kiểm thử của MSFT, giá cổ phiếu tăng dần đều từ năm 2011 đến 2016 mô hình dự đoán gần như khớp giá thực tế. Sau năm 2016, giá cổ phiếu của MSFT bắt đầu tăng trưởng thẳng đứng, mô hình lúc này vẫn dự đoán đúng xu hướng tăng trưởng của cổ phiếu nhưng bắt đầu cho thấy sự sai lệch không nhiều.



Hình 4.13: Kết quả loss trên tập train, test của mô hình LSTM trên tập dữ liệu FB



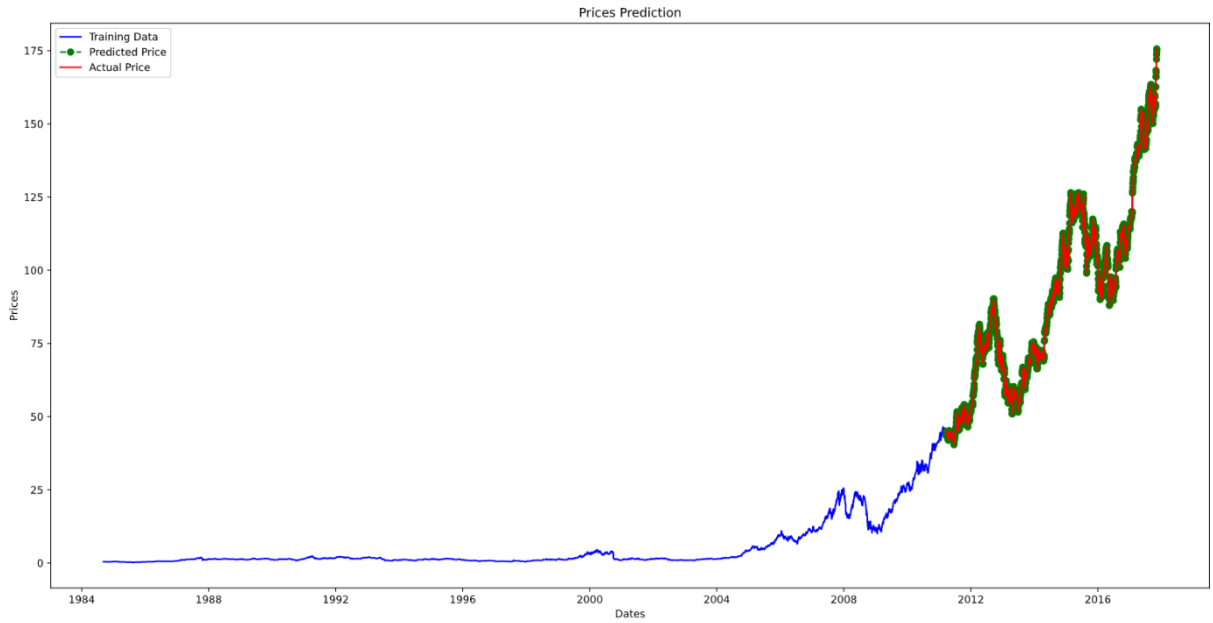
Hình 4.14: Kết quả dự đoán của mô hình LSTM trên tập dữ liệu FB

Mô hình LSTM cho thấy sự hiệu quả đối với các bài toán dạng chuỗi thời gian (time-series) trên tập dữ liệu FB. Điều này cho phép nó dự đoán đúng xu hướng tăng giảm trên tập dữ liệu huấn luyện.

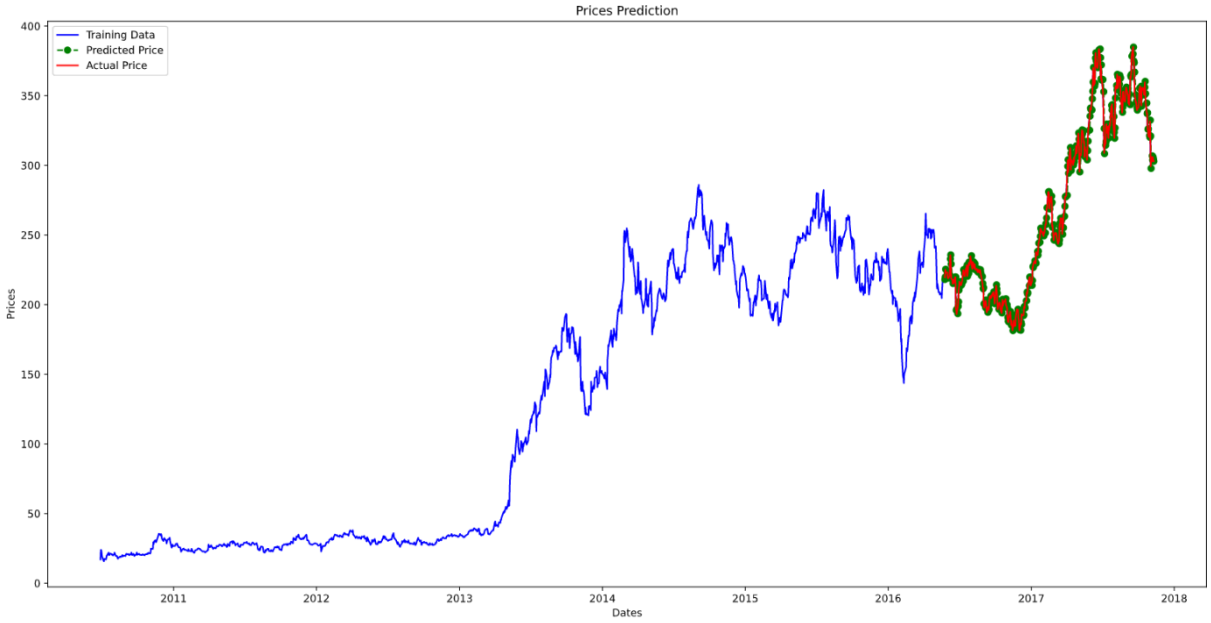
	FB	MSFT	TSLA	AAPL	Average
RMSE	1.68871188	0.579996592	6.14476349	1.398942483	2.453103611
MAPE	0.153720489	0.459809122	0.272148278	0.439125563	0.331200863

Bảng 4.5: Kết quả thử nghiệm mô hình ARIMA trên bộ dữ liệu

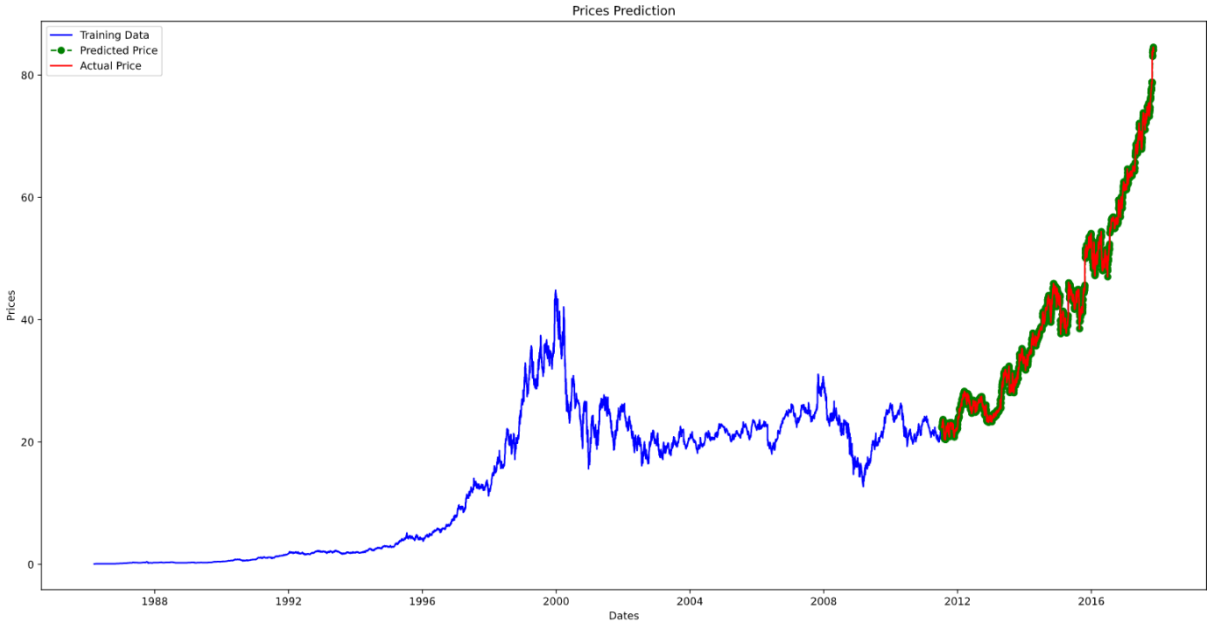
Bảng trên minh hoạ kết quả thử nghiệm mà chúng em đạt được khi cài đặt mô hình ARIMA, các kết quả được đánh giá dựa vào độ đo RMSE và MAPE để thuận tiện cho việc so sánh, đối chiếu giữa 2 mô hình. Mô hình dự đoán giá cổ phiếu của AAPL, TSLA và MSFT có RMSE, MAPE cao hơn so với mô hình dự đoán của FB do dữ liệu huấn luyện có xu hướng bất ổn định hơn.



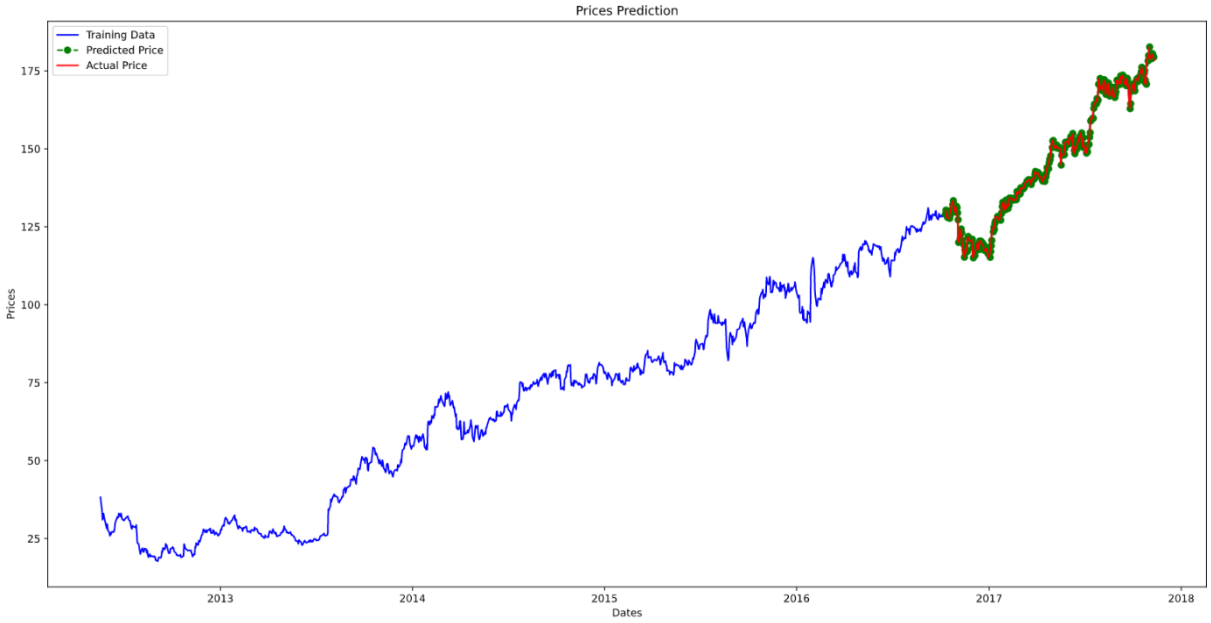
Hình 4.15: Kết quả dự đoán của mô hình ARIMA trên tập dữ liệu AAPL



Hình 4.16: Kết quả dự đoán của mô hình ARIMA trên tập dữ liệu TSLA



Hình 4.17: Kết quả dự đoán của mô hình ARIMA trên tập dữ liệu MSFT



Hình 4.18: Kết quả dự đoán của mô hình ARIMA trên tập dữ liệu FB

Sau khi trực quan kết quả, kết quả dự đoán của mô hình ARIMA trên tập dữ liệu FB là tốt hơn so với kết quả trên những tập dữ liệu khác, do dữ liệu trong suốt khoảng thời gian của tập dữ liệu FB có xu hướng rõ ràng và ổn định, đồng thời tập huấn luyện và tập kiểm thử có sự tương đồng cao so với công ty còn lại.

4.4. Kết luận

Với hai phương pháp chúng em đã chọn để dự đoán giá cổ phiếu, kết quả dự đoán là tương đối tốt khi mô hình đã dự đoán được xu hướng khá tương đồng so với dữ liệu kiểm thử, mặc dù vẫn có sự chênh lệch tuỳ theo sự khác biệt giữa dữ liệu huấn luyện và dữ liệu kiểm thử. LSTM chắc chắn là phức tạp hơn và khó huấn luyện hơn và trong hầu hết các trường hợp không vượt quá hiệu suất của một mô hình ARIMA đơn giản. Các phương pháp cổ điển như ARIMA tập trung vào sự phụ thuộc vào thời gian cố định: mối quan hệ giữa các quan sát tại các thời điểm khác nhau, đòi hỏi phân tích và đặc tả số lượng quan sát trễ được cung cấp làm đầu vào.

Chương 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Dự đoán thị trường chứng khoán là nhu cầu thực tế nhằm tạo sự thuận lợi cho việc đầu tư. Những dự đoán chắc chắn luôn rất hữu ích nhằm giảm thiểu các yếu tố rủi ro trong bất kỳ loại cổ phiếu hay cách thức đầu tư nào. Yếu tố rủi ro có thể được phân tích trên cơ sở dữ liệu lịch sử và xu hướng đầu tư của các nhà đầu tư trước đó.

Trước khi áp dụng kỹ thuật học máy, học sâu, chúng em đã tiến hành thực hiện các thao tác: tiền xử lý dữ liệu, phân tích, trực quan dữ liệu, lựa chọn cách chia dữ liệu phù hợp để huấn luyện mô hình và kiểm tra kết quả đạt được dựa trên những độ đo RMSE và MAPE. Bộ dữ liệu mà chúng em sử dụng là bộ dữ liệu về cổ phiếu của bốn công ty hàng đầu thế giới và có giá cổ phiếu biến động trong khoảng thời gian nhiều năm: Apple, Microsoft, Tesla, Facebook.

Đồ án lần này của chúng em đã phân tích và cài đặt thực nghiệm một kỹ thuật học máy (Auto Regressive Integrated Moving Average), một kỹ thuật học sâu (Long Short-Term Memory) tiêu biểu trong bài toán dự đoán giá cổ phiếu.

Dựa trên kết quả thu được khi giải quyết bài toán, phương pháp LSTM có thể đưa ra những dự đoán chính xác về giá cổ phiếu tương đương với ARIMA.

Nhìn chung, điểm số lỗi của mô hình LSTM xấp xỉ mô hình ARIMA vì phương pháp LSTM có hiệu quả cao khi dự đoán giá trị dài hạn, còn phương pháp ARIMA có hiệu quả cao khi dự đoán giá trị ngắn hạn. Trong bài toán này, việc dự đoán thực hiện trong khoảng thời gian ngắn, vì thế hiệu quả mà hai mô hình thể hiện là tương đương nhau.

5.2. Hạn chế

Bên cạnh các kết quả đạt được, đồ án của chúng em cũng tồn tại một số hạn chế. Kết quả RMSE và MAPE của mô hình được đào tạo từ dữ liệu của hai công ty Apple và Tesla vẫn tương đối cao, nguyên do là đặc tính có sự khác biệt lớn giữa tập dữ liệu

huấn luyện và tập dữ liệu kiểm thử, dẫn đến giá trị dự đoán mắc lỗi nhiều hơn. Đây là hạn chế lớn nhất cần được khắc phục.

LSTM đòi hỏi nhiều tài nguyên và thời gian để huấn luyện, cũng như áp dụng trong thế giới thực. Về mặt kỹ thuật, chúng cần băng thông bộ nhớ vì các lớp tuyến tính hiện diện trong mỗi ô mà hệ thống thường không cung cấp được. Do đó, về mặt phần cứng, LSTM trở nên khá tốn kém.

Với sự phát triển trong lĩnh vực khai thác dữ liệu (data mining), các nhà phát triển đang tìm kiếm một mô hình có thể nhớ thông tin trong quá khứ lâu hơn LSTM. Nguồn cảm hứng cho loại mô hình như vậy là thói quen của con người khi chia một phần thông tin nhất định thành các phần nhỏ để dễ nhớ hơn.

LSTM dễ bị overfitting, và cũng rất khó áp dụng thuật toán nào để loại bỏ tình trạng này.

Hạn chế của việc sử dụng mô hình ARIMA là mô hình này chỉ mang tính chất dự báo ngắn hạn. Trên thực tế, trong một số trường hợp, các nhà nghiên cứu cần đưa ra dự báo dài hạn. Trong tương lai, mô hình này còn có thể được triển khai cho bất kỳ loại dữ liệu nào khác, chẳng hạn như dữ liệu lượng mưa.

5.3. Hướng phát triển

Dự đoán thị trường chứng khoán là một lĩnh vực nghiên cứu tiềm năng và mang lại lợi ích tiền tệ vì tổng vốn hóa thị trường của nó là rất lớn.

Để đầu tư thành công, các nhà đầu tư cần quan tâm đến việc dự báo tình hình tương lai của thị trường chứng khoán có thể mang lại lợi nhuận rất cao. Một hệ thống dự đoán tốt sẽ giúp các nhà đầu tư thực hiện đầu tư chính xác hơn và sinh lời nhiều hơn bằng cách cung cấp các thông tin hỗ trợ như xu hướng tương lai của giá cổ phiếu.

Bên cạnh giá trị của chính nó trong quá khứ, giá cổ phiếu trong tương lai còn có thể bị tác động bởi các yếu tố như chính trị, biến động của nền kinh tế, các tin tức tài chính cũng như tác động của các phương tiện truyền thông xã hội. Do đó, nếu kết hợp các kỹ

thuật phân tích bằng học máy và các phương pháp truyền thống sẽ mang lại hiệu quả dự đoán cao hơn.

TÀI LIỆU THAM KHẢO

Danh mục Tài liệu Tiếng Anh

[1] "Adam," [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam. [Accessed 20 06 2021].

[2] "Dense layer," [Online]. Available: https://keras.io/api/layers/core_layers/dense/. [Accessed 18 06 2021].

[3] "LearningRateScheduler," [Online]. Available: https://keras.io/api/callbacks/learning_rate_scheduler/. [Accessed 20 06 2021].

[4] "Mean absolute percentage error," [Online]. Available: https://en.wikipedia.org/wiki/Mean_absolute_percentage_error. [Accessed 18 06 2021].

[5] "Root-mean-square error," [Online]. Available: https://en.wikipedia.org/wiki/Root-mean-square_deviation. [Accessed 18 06 2021].

[6] J. Brownlee, "Using Learning Rate Schedules for Deep Learning Models in Python with Keras," [Online]. Available: https://machinelearningmastery.com/using-learning-rate-schedules-deep-learning-models-python-keras/. [Accessed 20 06 2021].

[7] N. S. Chauhan, "Stock Market Forecasting Using Time Series Analysis," [Online]. Available: https://www.kdnuggets.com/2020/01/stock-market-forecasting-time-series-analysis.html. [Accessed 18 06 2021].

[8] B. Chen, "Learning Rate Schedule in Practice: an example with Keras and TensorFlow 2.0," [Online]. Available:

https://towardsdatascience.com/learning-rate-schedule-in-practice-an-example-with-keras-and-tensorflow-2-0-2f48b2888a0c. [Accessed 20 06 2021].

[9] U. Dev, "EDA of Stock Market using Time Series," [Online]. Available: https://usharbudha-dev09.medium.com/eda-of-stock-market-using-time-series-9662fd18bfc5. [Accessed 18 06 2021].

[10] S. Loukas, "Time-Series Forecasting: Predicting Stock Prices Using An ARIMA Model," [Online]. Available: https://towardsdatascience.com/time-series-forecasting-predicting-stock-prices-using-an-arima-model-2e3b3080bd70. [Accessed 18 06 2021].

[11] S. Oguntayo, "Preprocessing Time Series Data for Supervised Machine Learning," [Online]. Available: https://towardsdatascience.com/preprocessing-time-series-data-for-supervised-learning-2e27493f44ae. [Accessed 20 06 2021].

[12] C. Olah, "Understanding LSTM Networks," [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed 18 06 2021].

[13] R. Orac, "LSTM for time series prediction," [Online]. Available: https://towardsdatascience.com/lstm-for-time-series-prediction-de8aeb26f2ca. [Accessed 20 06 2021].

[14] J. S. A. D. Sidra Mehtab, "Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models," [Online]. Available: https://arxiv.org/abs/2009.10819. [Accessed 18 06 2021].

[15] P. P. Ippolito, "Dash Online Data Science Dashboard using Plotly," [Online]. Available: https://github.com/pierpaolo28/Data-Visualization/tree/master/Dash. [Accessed 21 06 2021].

Danh mục Tài liệu Tiếng Việt

[1] "Giới thiệu tổng quan về keras," [Online]. Available: https://trituenhantao.github.io/2020/08/21/keras-la-gi-gioi-thieu-ve-keras/. [Accessed 18 06 2021].

[2] N. C. Thắng, "Keras Callbacks," [Online]. Available: https://miai.vn/2020/09/05/keras-callbacks-tro-thu-dac-luc-khi-train-models/. [Accessed 20 06 2021].