

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

# **ĐỒ ÁN TỐT NGHIỆP**

**Dự đoán xu hướng thị trường chứng khoán dựa trên  
các mô hình deep learning**

**LÊ TRỌNG ĐẠT**

dat.lt183493@sis.hust.edu.vn

**Ngành: Công nghệ thông tin**

**Chuyên ngành: Khoa học máy tính**

**Giảng viên hướng dẫn: TS. Trịnh Anh Phúc**

\_\_\_\_\_  
Chữ kí GVHD

**Khoa:** Khoa học máy tính

**Trường:** Công nghệ thông tin và Truyền thông

**HÀ NỘI, 08/2022**

# LỜI CẢM ƠN

Trải qua thời gian nghiên cứu tôi đã hoàn thành đồ án “Dự đoán xu hướng thị trường chứng khoán dựa trên các mô hình deep learning”. Song chúng tôi đã không thể hoàn thành đề tài này nếu như không có sự hướng dẫn của thầy cô cũng như là sự giúp đỡ của bạn bè.

Tôi xin bày tỏ lòng biết ơn chân thành tới thầy giáo TS. Trịnh Anh Phúc đã trực tiếp giúp đỡ tôi hoàn thành đồ án này. Thầy đã cung cấp cho tôi tất cả những kỹ năng cần thiết, tài liệu quan trọng cũng như là những lời khuyên bổ ích để phục vụ cho đề tài này. Dưới sự hướng dẫn tận tình của thầy, tôi đã có thể hoàn thiện đề tài này tốt nhất trong khả năng của bản thân.

Tôi chân thành cảm ơn quý Thầy, Cô trong Trường Công nghệ Thông tin và Truyền thông - Đại học Bách khoa Hà Nội đã tận tình truyền đạt kiến thức trong những năm tôi học tập. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu đồ án tốt nghiệp mà còn là hành trang quý báu để tôi bước vào đời một cách vững chắc và tự tin.

Do kiến thức còn hạn hẹp nên không tránh khỏi những thiếu sót trong cách diễn đạt và trình bày. Tôi rất mong nhận được sự đóng góp ý kiến của quý thầy cô để báo cáo đồ án này đạt được kết quả tốt nhất.

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Thị trường cổ phiếu có một vai trò quan trọng trong sự phát triển của xã hội hiện đại. Chúng cho phép triển khai các nguồn lực kinh tế. Sự thay đổi giá cổ phiếu phản ánh những thay đổi trên thị trường. Do đó, dự đoán thị trường cổ phiếu được xem là một trong những lĩnh vực khá phổ biến và quý giá nhất trong lĩnh vực tài chính. Với khả năng xử lý dữ liệu mạnh mẽ trên nhiều lĩnh vực, học sâu cũng được sử dụng một cách rộng rãi trong lĩnh vực tài chính.

Trong nghiên cứu này, chúng tôi trình bày một số đề xuất cải tiến cho mô hình Transformer cơ bản để giải quyết nhiệm vụ dự đoán giá cổ phiếu. Thứ nhất, chúng tôi sử dụng mô hình Time2Vec giúp mô hình hóa đặc trưng về thời gian để nâng cao khả năng biểu diễn thời gian của mô hình Transformer. Thứ hai, chúng tôi phát triển một phương pháp hiệu chỉnh tham số để tránh học các đặc trưng dư thừa trong cơ chế Multi-Head Attention.

Chúng tôi thực nghiệm trên cổ phiếu IBM (International Business Machines) và một số cổ phiếu khác có chuỗi ngày giao dịch trong phạm vi rộng và dùng chúng để thử dự đoán giá đóng cửa hàng ngày. Kết quả thực nghiệm cho thấy phương pháp Transformer với các đề xuất của chúng tôi có thể đạt được kết quả tốt hơn trong việc dự đoán giá cổ phiếu so với nhiều mô hình dự đoán khác.

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Các giải pháp hiện tại và hạn chế .....	2
1.3 Mục tiêu và định hướng giải pháp .....	3
1.4 Đóng góp của đồ án .....	4
1.5 Bố cục đồ án .....	4
<b>CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT .....</b>	<b>6</b>
2.1 Ngữ cảnh của bài toán.....	6
2.2 Các kết quả nghiên cứu tương tự .....	6
2.3 Cổ phiếu và phân tích thị trường.....	7
2.3.1 Các khái niệm.....	7
2.3.2 Phân tích kỹ thuật .....	8
2.3.3 Đường trung bình động.....	11
2.3.4 Bollinger Bands .....	11
2.4 Học máy .....	12
2.5 Học sâu .....	13
2.5.1 Khái niệm .....	13
2.5.2 Transformer.....	14
2.5.3 Time2Vec.....	18
<b>CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT.....</b>	<b>21</b>
3.1 Kiến trúc mạng đề xuất .....	21
3.2 Tiền xử lý dữ liệu.....	22
3.2.1 Chuyển đổi dữ liệu thành phần trăm thay đổi .....	22
3.2.2 Moving Average.....	23

3.3 Time2Vec .....	23
3.4 Transformer Encoder .....	25
3.5 Hiệu chỉnh trực giao tham số .....	27
<b>CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM.....</b>	<b>29</b>
4.1 Tổng quan giải pháp.....	29
4.2 Môi trường phát triển .....	29
4.3 Tiền xử lý dữ liệu.....	30
4.3.1 Thu thập dữ liệu.....	30
4.3.2 Biến đổi dữ liệu .....	30
4.3.3 Chuẩn hóa dữ liệu .....	31
4.3.4 Tạo dữ liệu đào tạo.....	32
4.4 Phương pháp đánh giá.....	33
4.5 Các tham số đánh giá .....	34
4.6 Phương pháp thí nghiệm.....	34
4.7 Kết quả thí nghiệm.....	34
4.7.1 LSTM .....	34
4.7.2 Mô hình LSTM kết hợp CNN.....	36
4.7.3 Mô hình Transformer cơ bản .....	37
4.7.4 Mô hình Transformer đề xuất .....	38
4.8 Đánh giá .....	39
<b>CHƯƠNG 5. KẾT LUẬN .....</b>	<b>41</b>
5.1 Kết luận.....	41
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>44</b>

## DANH MỤC HÌNH VẼ

Hình 2.1	Dữ liệu giao dịch lịch sử của công ty Tesla cung cấp bởi Yahoo Finance . . . . .	9
Hình 2.2	Biểu đồ thanh nền . . . . .	10
Hình 2.3	Giá cổ phiếu và các chỉ báo kỹ thuật . . . . .	11
Hình 2.4	Lược sử phát triển của Trí tuệ nhân tạo . . . . .	13
Hình 2.5	Mô hình Transformer . . . . .	15
Hình 2.6	Cơ chế Self-Attention . . . . .	17
Hình 2.7	Cơ chế Multi-Head Attention . . . . .	18
Hình 3.1	Kiến trúc của mô hình đề xuất . . . . .	21
Hình 3.2	Dữ liệu cổ phiếu IBM thu thập từ thư viện Yfinance . . . . .	23
Hình 3.3	Cấu trúc lớp Encoder . . . . .	26
Hình 3.4	Trọng số chú ý của cơ chế Multi-Head Attention . . . . .	27
Hình 4.1	Quy trình thực hiện . . . . .	29
Hình 4.2	Dữ liệu cổ phiếu IBM thu thập từ thư viện Yfinance . . . . .	30
Hình 4.3	Đường trung bình động 10 ngày . . . . .	31
Hình 4.4	Phân bố dữ liệu sau khi biến đổi thành phần trăm thay đổi . . . . .	31
Hình 4.5	Cửa sổ trượt kích thước cửa sổ 5 và bước nhảy 1 . . . . .	32
Hình 4.6	Dữ liệu huấn luyện, validation và test . . . . .	33
Hình 4.7	Kiến trúc mô hình LSTM [26] . . . . .	35
Hình 4.8	Hàm mất mát của mô hình LSTM . . . . .	35
Hình 4.9	Biểu đồ kết quả dự đoán của mô hình LSTM . . . . .	36
Hình 4.10	Kiến trúc mô hình CNN kết hợp LSTM [27] . . . . .	36
Hình 4.11	Hàm mất mát của mô hình LSTM kết hợp CNN . . . . .	37
Hình 4.12	Biểu đồ kết quả dự đoán của mô hình LSTM kết hợp CNN . . . . .	37
Hình 4.13	Biểu đồ kết quả dự đoán của mô hình Transformer cơ bản . . . . .	38
Hình 4.14	Hàm mất mát của mô hình Transformer cơ bản . . . . .	38
Hình 4.15	Biểu đồ kết quả dự đoán của mô hình Transformer đề xuất . . . . .	39
Hình 4.16	Hàm mất mát của mô hình Transformer đề xuất . . . . .	39
Hình 4.17	Sai số độ đo MSE của các mô hình . . . . .	40

## DANH MỤC BẢNG BIỂU

Bảng 4.1	Tham số mô hình . . . . .	34
Bảng 4.2	Kết quả độ đo của mô hình LSTM . . . . .	35
Bảng 4.3	Kết quả độ đo của các mô hình LSTM kết hợp CNN . . . . .	37
Bảng 4.4	Kết quả độ đo của mô hình Transformer cơ bản . . . . .	37
Bảng 4.5	Kết quả độ đo của mô hình Transformer đề xuất . . . . .	38
Bảng 4.6	Kết quả độ đo của các mô hình . . . . .	39

## DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
Conv-LSTM	Mạng tích chập kết hợp mạng bộ nhớ dài ngắn
IBM	Tập đoàn International Business Machines
LSTM	Mạng bộ nhớ dài - ngắn (Long-Short Term Memory)
MA	Đường trung bình động (Moving Average)
MAE	Mean Absolute Error
MSE	Mean Squared Error
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNN	Mạng nơ-ron hồi quy (Recurrent Neural Network)
SMA	Đường trung bình động đơn giản (Simple Moving Average)
SVM	Máy vec tơ hỗ trợ (Support Vector Machine)



# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Thị trường chứng khoán chính là thước đo sức mạnh nền kinh tế của nhiều nước trên thế giới và là công cụ phản ánh sự thay đổi của nền kinh tế trong tương lai. Cụ thể là giá chứng khoán tăng sẽ cho thấy nền kinh tế đang phát triển và ngược lại, giá chứng khoán giảm lại là dự báo không mấy tốt đẹp về triển vọng của nền kinh tế trong tương lai gần. Việc quan sát xu hướng phát triển của thị trường chứng khoán sẽ giúp các nhà kinh tế, ngân hàng, doanh nghiệp... chuẩn bị những thứ cần thiết trước sự biến động của nền kinh tế.

Bên cạnh đó, thị trường chứng khoán luôn thu hút các nhà đầu tư do khả năng sinh lời cao nhưng cũng đi kèm nhiều rủi ro. Lợi nhuận của các nhà đầu tư chủ yếu được tạo ra từ việc dự đoán trước sự biến động của thị trường từ đó có các hành động mua vào hay bán phù hợp. Việc lựa chọn mua hay bán cổ phiếu nào phụ thuộc vào kỳ vọng phát triển của một công ty trong tương lai. Do đó các nhà đầu tư khi tham gia thị trường chứng khoán cần phải dự đoán được xu hướng thị trường chứng khoán trong ngắn hạn hoặc dài hạn để giảm thiểu rủi ro và đưa ra các quyết định đúng đắn.

Tuy nhiên, dự đoán về xu hướng của thị trường chứng khoán là một nhiệm vụ đầy thách thức do tính chất phi tuyến tính và đặc trưng bởi một hành vi đám đông của thị trường. Các yếu tố tạo nên sự biến động trong lĩnh vực này rất phức tạp và bắt nguồn từ nhiều nguyên nhân với các bản chất khác nhau. Từ các quyết định kinh tế, chính trị, hiệu quả hoạt động cụ thể của công ty cho đến những nguyên nhân không rõ ràng như tin đồn, quyết định của các nhà đầu tư nổi tiếng, các quy luật của thị trường... khiến cho hầu như không thể tính đến tất cả các yếu tố liên quan khi đưa ra quyết định giao dịch. Rất khó để phân tích các yếu tố này theo cách thủ công, vì vậy sẽ tốt hơn nếu có các công cụ hỗ trợ phân tích những dữ liệu này.

Trong trí tuệ nhân tạo, dự đoán về biến động giá cổ phiếu là bài toán thuộc lĩnh vực phân tích chuỗi thời gian. Các cách tiếp cận truyền thống để dự đoán giá cổ phiếu chủ yếu dựa trên các chỉ số kỹ thuật cơ bản, từ đó dự đoán các mẫu biến đổi lặp đi lặp lại một cách rõ ràng hoặc tiềm ẩn từ dữ liệu giao dịch lịch sử. Tuy nhiên, hiệu quả của những phương pháp đó bị hạn chế bởi hai khía cạnh. Thứ nhất, chúng thường yêu cầu kiến thức chuyên môn về tài chính. Thứ hai, các phương pháp này chỉ nắm bắt được các mô hình dữ liệu đơn giản và cấu trúc phụ thuộc đơn giản của chuỗi thời gian. Với sự phát triển của công nghệ trí tuệ nhân tạo, ngày càng có nhiều nhà nghiên cứu cố gắng giải quyết vấn đề này bằng cách sử dụng các thuật

toán học máy, chẳng hạn như Cortes và cộng sự sử dụng mô hình SVM [1], Altman sử dụng Nearest Neighbors [2], Breiman và cộng sự sử dụng Random Forest [3]. Gần đây, kể từ khi mạng nơ-ron sâu thể hiện khả năng mạnh mẽ của nó trong việc giải quyết các vấn đề phi tuyến tính, nghiên cứu dự đoán cổ phiếu dựa trên kỹ thuật học sâu ngày càng trở nên phổ biến hơn trong những năm gần đây và cho thấy những lợi thế đáng kể so với các phương pháp tiếp cận truyền thống.

Đồ án này sẽ khám phá một ứng dụng của học sâu trong dự đoán xu hướng thị trường chứng khoán. Cụ thể, tôi sẽ sử dụng mô hình dựa trên Transformer để dự báo xu hướng thị trường chứng khoán trong ngắn hạn. Kết quả cho thấy mô hình đề xuất của tôi đã đạt được một số kết quả khả quan trong nghiên cứu kỹ thuật học sâu so với một số mô hình học sâu khác. Tuy nhiên, luận văn không thể tránh khỏi những thiếu sót, rất mong nhận được những ý kiến đóng góp, nhận xét để tôi tiếp tục hoàn thiện và nâng cao kết quả nghiên cứu.

## **1.2 Các giải pháp hiện tại và hạn chế**

Các nghiên cứu dự đoán chứng khoán dựa trên kỹ thuật học sâu có thể được nhóm lại thành hai loại: (1) phân tích cơ bản (Fundamental analysis) và (2) phân tích kỹ thuật (Technical analysis). Phân tích cơ bản xây dựng các tín hiệu dự đoán bằng cách sử dụng thông tin cơ bản như văn bản tin tức, báo cáo tài chính và báo cáo của nhà phân tích. Từ các văn bản này, các nhà phân tích sẽ cố gắng dự đoán thái độ của phần lớn các nhà đầu tư đối với một cổ phiếu. Nếu các nhà đầu tư có thái độ tích cực thì có khả năng cao cổ phiếu sẽ tăng và ngược lại nếu thái độ các nhà đầu tư tiêu cực thì giá cổ phiếu sẽ giảm. Ví dụ, Schumaker và Chen [4]; Xu và Cohen [5]; Chen và cộng sự [6] sử dụng các phương pháp xử lý ngôn ngữ tự nhiên để dự đoán biến động giá cổ phiếu bằng cách trích xuất các đặc trưng tiềm ẩn từ các văn bản có liên quan đến thị trường, chẳng hạn như tin tức, báo cáo và thậm chí cả tin đồn. Mặt khác, phân tích kỹ thuật dự đoán thị trường tài chính bằng cách sử dụng dữ liệu lịch sử của cổ phiếu. Phương pháp này cho rằng các chỉ số của một cổ phiếu thường sẽ giao động theo một khuôn mẫu nào đó. Nếu nắm bắt được quy luật biến đổi đó thì có thể dự đoán được giá cổ phiếu và tạo ra lợi nhuận. Một cách tiếp cận tự nhiên cho phương pháp này là sử dụng các mô hình tuần tự để phân tích giá cổ phiếu, chẳng hạn như RNN của Rumelhart và cộng sự [7], LSTM của Hochreiter và cộng sự [8], Conv-LSTM của Xingjian [9] và ALSTM của Qin và cộng sự [10]. Tuy nhiên, nhược điểm chính của các phương pháp này các mô hình được áp dụng cho các phương pháp này là chúng khó nắm bắt được các phụ thuộc dài hạn [11], chẳng hạn như các phụ thuộc trong vài tháng trong chuỗi thời gian tài chính.

Gần đây, một mô hình học sâu xử lý dữ liệu chuỗi thời gian nổi tiếng có tên

là Transformer của Vaswani và cộng sự [12] đã đạt được thành công lớn trong các nhiệm vụ xử lý ngôn ngữ tự nhiên. Khác biệt với các mô hình dựa trên RNN, Transformer sử dụng cơ chế tự chú ý nhiều đầu để tìm hiểu mối quan hệ giữa các vị trí khác nhau trên toàn cầu, do đó khả năng học tập các phụ thuộc lâu dài được nâng cao. Từ khi Transformer ra đời đến nay, đã có nhiều nhà nghiên cứu đã áp dụng mô hình Transformer cho bài toán dự đoán giá cổ phiếu như J.Liu và cộng sự [13] đã sử dụng một biến thể của mô hình Transformer kết hợp mạng Capsule để dự đoán giá cổ phiếu, hay Q. Ding và cộng sự sử dụng cơ chế phân cấp thời gian giao dịch cổ phiếu theo ngày và theo tháng để huấn luyện mô hình Transformer [14]. Tuy nhiên, Transformer chuẩn được thiết kế cho các tác vụ ngôn ngữ tự nhiên và do đó nó có một số hạn chế trong việc giải quyết dự đoán tài chính: (1) Khả năng học các thông tin về thời gian yếu trong khi thời gian là một đặc trưng quan trọng trong xử lý dữ liệu cổ phiếu hàng ngày. (2) cơ chế Multi-Head Attention có xu hướng học các head dư thừa [15][16][17], phân bố tham số của các head thường tập trung nên khả năng học các đặc trưng khác nhau bị hạn chế.

### **1.3 Mục tiêu và định hướng giải pháp**

Đối với các nhà phân tích thị trường chứng khoán, yêu cầu cần phải phát triển các công cụ mới kết hợp với các mô hình dự đoán truyền thống để điều chỉnh các dự đoán bằng cách tính đến các yếu tố không bắt nguồn từ công ty, đặc biệt là các yếu tố phi tuyến. Công cụ này sẽ giúp cho các nhà phân tích hay các nhà đầu tư phân tích một lượng lớn dữ liệu và có thêm cách nhìn khách quan vào sự biến động của giá cổ phiếu. Mục đích của luận văn này là thúc đẩy cách tiếp cận Học sâu với mạng thần kinh nhân tạo để cố gắng giải quyết vấn đề này.

Trong nghiên cứu này, tôi đề xuất một phương pháp dựa trên mô hình Transformer để dự đoán giá đóng cửa của cổ phiếu trong ngày kế tiếp từ lịch sử giao dịch của một số ngày trước đó. Điểm nổi bật chính của mô hình được đề xuất là khả năng nắm bắt các phụ thuộc dài hạn, ngắn hạn của chuỗi thời gian tài chính. Để giải quyết hai vấn đề của mô hình Transformer đã nêu ở 1.2, tôi đề xuất áp dụng một số cải tiến cho mô hình Transformer: (1) Sử dụng mô hình Time2Vec [18] để mã hóa đặc trưng về thời gian, tăng khả năng học các phụ thuộc về thời gian của mô hình Transformer. (2) Hiệu chỉnh trực giao tham số cho các head tránh học các head dư thừa trong cơ chế Multi-Head Attention.

Hơn nữa, các mô hình theo phương pháp phân tích kỹ thuật thường cung cấp đầu vào là dữ liệu giá cổ phiếu thô. Tôi cho rằng đây là một ý tưởng tồi do giá của cổ phiếu phụ thuộc vào đơn vị tiền tệ. Khi nhìn vào biểu đồ giá cổ phiếu của IBM, bạn có thể thấy giá từ năm 2005 đến năm 2008 (\$ 50- \$ 100) ở một mức hoàn

toàn khác so với giá của những năm 2020 và 2022 (\$ 140- \$ 220). Về bản chất, hai khoảng giá này không liên quan gì đến nhau. Mặt khác, khi dự đoán xu hướng cổ phiếu, người ta thường quan tâm đến tỉ lệ tăng hay giảm giá cổ phiếu hơn là mức giá cụ thể. Do đó, để đưa các mức giá trong quá khứ ngang bằng với mức giá trong hiện tại và hữu ích hơn cho việc đào tạo mạng nơ-ron, tôi chuyển đổi giá của một ngày thành phần trăm thay đổi so với ngày trước đó. Lợi ích của việc chuyển đổi này là dữ liệu đầu vào không còn phụ thuộc vào đơn vị tiền tệ đồng thời thể hiện rõ hơn xu hướng tăng hay giảm của cổ phiếu.

Nghiên cứu này là sử dụng dữ liệu giao dịch của cổ phiếu để dự đoán giá đóng cửa trong ngày kế tiếp. Tôi thực nghiệm trên cổ phiếu IBM (International Business Machines) và một số cổ phiếu khác trong phạm vi 30 năm (từ 1992 đến 2022) để huấn luyện mô hình. Kết quả thực nghiệm cho thấy phương pháp Transformer với các cải tiến có thể đạt được kết quả tốt trong việc dự đoán giá cổ phiếu so với nhiều mô hình dự đoán khác.

#### **1.4 Đóng góp của đề án**

Đề án này có 3 đóng góp chính như sau:

1. Đề án đề xuất một phương pháp tiền xử lý dữ liệu nhằm chuẩn hóa dữ liệu, loại bỏ ảnh hưởng của mức giá đến quá trình học của mô hình. Điều này không chỉ giúp loại bỏ các ảnh hưởng tiêu cực của đơn vị tiền tệ, tăng khả năng dự đoán của mô hình mà còn giúp cho quá trình huấn luyện mô hình hiệu quả hơn.
2. Đề án đề xuất một phương pháp dựa trên mô hình Transformer với những cải tiến: sử dụng mô hình Time2Vec để biểu diễn đặc trưng về thời gian tăng khả năng học các thông tin về thời gian của mô hình, hiệu chỉnh trực giao tham số giúp cơ chế Multi-Head Attention tránh học các head dư thừa.
3. Trong các thử nghiệm, mô hình được đề xuất cho kết quả tốt hơn đáng kể so với các mô hình như LSTM, LSTM kết hợp CNN.

#### **1.5 Bố cục đề án**

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau.

Chương 2 giới thiệu về các kiến thức nền tảng mà đề án sử dụng. Cụ thể, chương này sẽ giới thiệu về các khái niệm về cổ phiếu, các chỉ báo kỹ thuật phân tích biến động giá cổ phiếu được sử dụng trong đề án. Sau đó, tôi trình bày các kiến thức nền tảng của học sâu và chi tiết lý thuyết cơ bản của các mô hình học sâu sẽ được sử dụng.

Trong Chương 3, tôi trình bày đóng góp chính của đề án, một mô hình dựa trên

Transformer với những đề xuất cải tiến của tôi. Trước tiên, tôi mô tả bước tiền xử lý dữ liệu được đề xuất. Đó là biến đổi dữ liệu giao dịch cổ phiếu về phần trăm thay đổi nhằm loại bỏ các ảnh hưởng tiêu cực của sức mua đồng tiền đối với giá cổ phiếu. Sau đó tôi trình bày mô hình Transformer với các đề xuất cải tiến được giải thích chi tiết và ứng dụng chúng vào bài toán dự đoán giá cổ phiếu. Các cải tiến được trình bày bao gồm sử dụng mô hình Time2Vec để mã hóa đặc trưng về thời gian tăng khả năng học các thông tin chuỗi tài chính cho mô hình và hiệu chỉnh trực giao cho tham số của mô hình.

Chương 4 trình bày các kết quả thực nghiệm trên bộ dữ liệu cổ phiếu IBM. Đầu tiên, tôi mô tả cách thu thập và tiền xử lý dữ liệu và các kịch bản thí nghiệm. Tôi tiến hành 2 thí nghiệm: Thứ nhất, tôi áp dụng 1 trong các cải tiến vào mô hình để đánh giá từng cải tiến; Thứ 2, tôi so sánh hiệu suất của mô hình đề xuất và các mô hình khác. Sau đó tôi đánh giá mô hình và so sánh kết quả của các mô hình trong từng thí nghiệm, từ đó rút ra được các nhận xét về các mô hình đã cài đặt. Kết quả cho thấy mô hình Transformer với các cải tiến đề xuất của tôi có khả năng dự đoán tốt hơn một số mô hình khác như LSTM, LSTM kết hợp CNN.

## CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

Chương này trình bày ngắn gọn những kiến thức nền tảng mà nghiên cứu này sử dụng, chẳng hạn như các chỉ số tài chính thường được sử dụng, phương pháp phân tích kỹ thuật và các mô hình học sâu. Để trả lời cho bài toán ở phần trước, tôi bắt đầu với việc trình bày ngữ cảnh của bài toán và các nghiên cứu đã có. Sau đó các mô hình Transformer cơ bản được trình bày chi tiết.

### 2.1 Ngữ cảnh của bài toán

Giá cổ phiếu biến động chủ yếu do cung và cầu của thị trường, theo lý thuyết kinh tế vi mô. Dự báo giá trị của cổ phiếu rất phức tạp do tính không cố định, không tuyến tính và phụ thuộc vào nhiều yếu tố, do đó ảnh hưởng đến tính biến động của cổ phiếu. Nhiều yếu tố liên quan đến giá trị dự đoán của cổ phiếu, chẳng hạn như điều kiện kinh tế chung, sự ổn định chính trị, giá trị của khách hàng, đánh giá của khách hàng của công ty, kỳ vọng của thương nhân và phương tiện truyền thông xã hội,...

Cách mạng Công nghiệp 4.0 mang đến những thay đổi toàn diện trong mọi mặt của cuộc sống. “Thế giới vận hành bằng công nghệ” trở thành xu thế tất yếu, là mục tiêu tranh đua cho mọi nền kinh tế, mọi doanh nghiệp cùng hướng đến. Trong cuộc cách mạng Công nghiệp lần thứ IV, Trí tuệ nhân tạo và Học sâu là những thành phần quan trọng, nổi bật, đã và đang được nghiên cứu, ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau trong đó có ngành ngân hàng nói chung và lĩnh vực công nghệ tài chính nói riêng, tạo ra những bước phá mạnh mẽ về tính hiệu quả, kịp thời, chính xác trong công tác giám sát, dự báo và hỗ trợ ra quyết định.

Dự đoán giá trị cổ phiếu bằng các mô hình toán học và máy tính đã phổ biến trong những thập kỷ qua. Với Học máy, và Học sâu, lượng dữ liệu khổng lồ có thể được xử lý và phân tích để dự đoán các mẫu xu hướng cổ phiếu theo thời gian. Các mẫu biến đổi giá và yếu tố quá phức tạp có thể được xác định với các công cụ Học máy và Học sâu.

### 2.2 Các kết quả nghiên cứu tương tự

Nhiều bài báo sử dụng thành công các phương pháp Học sâu trong việc phát triển các mô hình dự đoán xu hướng thị trường chứng khoán. Có thể chia các phương pháp này thành 2 hướng nghiên cứu là phân tích cơ bản (Fundamental Analysis) và phân tích kỹ thuật (Technical Analysis).

Phân tích cơ bản dựa trên những báo cáo, bài viết hoặc những trạng thái trên mạng xã hội để dự báo xu hướng cổ phiếu trong tương lai. Năm 2011, một bài báo

do Bollen và cộng sự [19] đã gây ấn tượng với các nhà nghiên cứu trên toàn thế giới bằng cách dự đoán giá cổ phiếu bằng cách sử dụng phân tích cảm xúc từ các bài đăng trên Twitter, bài báo này vẫn được nhiều người trích dẫn cho đến ngày nay. H. K. Sul và cộng sự [20] đã tìm thấy mối tương quan chặt chẽ giữa giá trị cảm xúc, tức là tâm trạng công khai trên Twitter và lợi nhuận cổ phiếu. V. S. Pagolu và cộng sự [21] đã chỉ ra nhược điểm trong [19] và đề xuất hai phương pháp khác: Word2Vec và N-gram. Schumaker và Chen [4] đề xuất một phương pháp học máy để phân tích các bài báo tài chính bằng cách sử dụng một số biểu diễn văn bản khác nhau.

Mặt khác, các phương pháp phân tích kỹ thuật trích xuất thông tin khối lượng giá từ dữ liệu giao dịch lịch sử và sử dụng các thuật toán máy học để dự đoán. Ví dụ Bao và cộng sự [22] đã sử dụng các phép biến đổi wavelet để loại bỏ nhiễu khỏi chuỗi giá chứng khoán trước khi đưa chúng vào một lớp AutoEncoder và lớp Long-short Term Memory (LSTM) để đưa ra dự đoán giá trong một ngày. Hiransha và cộng sự [23] đã so sánh Convolutional Neural Network (CNN) với Recurrent Neural Network (RNN) về dự đoán giá cổ phiếu của các công ty trong lĩnh vực công nghệ thông tin. Trong thử nghiệm của họ, mạng CNN cho kết quả tốt hơn so với RNN và LSTM. Sự khác biệt về hiệu suất được cho là do CNN không dựa trên dữ liệu lịch sử như trường hợp của các mô hình tuần tự. Ngoài ra, nhiều nhà nghiên cứu tập trung vào việc sử dụng các kỹ thuật Học tăng cường để giải quyết vấn đề giao dịch theo thuật toán. Ví dụ, Moody và Saell [24] đã giới thiệu một thuật toán học tăng cường để xác định các chính sách đầu tư sinh lời mà không cần xây dựng mô hình dự báo, và Dempster và Leemans [25] đã sử dụng học tăng cường để giao dịch trên thị trường ngoại hối.

## **2.3 Cổ phiếu và phân tích thị trường**

### **2.3.1 Các khái niệm**

Cổ phiếu là giấy chứng nhận sự góp vốn kinh doanh vào các công ty cổ phần. Giá trị ban đầu ghi trên cổ phiếu là mệnh giá của cổ phiếu. Mệnh giá là giá trị danh nghĩa. Số tiền nhận được từ khoản góp vốn gọi là cổ tức. Giá cổ phiếu giao động qua các phiên giao dịch trên thị trường chứng khoán và tách rời so với mệnh giá.

Khi nắm giữ chứng khoán thực chất là đang sở hữu một phần doanh nghiệp. Mức lợi nhuận thu được khi gửi tiết kiệm là cố định và chỉ khoảng 6-7%/năm, tuy nhiên tốc độ tăng trưởng trung bình của các công ty trên thị trường chứng khoán là 20%/năm. Việc của nhà đầu tư là chọn ra những công ty có tiềm năng tăng trưởng cao để mua cổ phiếu của các công ty đó, khi công ty làm ăn có lãi sẽ trả cổ tức cho nhà đầu tư, hoặc khi giá cổ phiếu tăng lên, nhà đầu tư sẽ thu được lợi nhuận từ

chênh lệch giá mua cổ phiếu và giá bán cổ phiếu.

Thị trường chứng khoán là nơi mà các loại chứng khoán được mua và bán. Thông qua các sàn giao dịch chứng khoán các nhà đầu tư có thể thực hiện giao dịch từ đó trao đổi quyền sở hữu các chứng khoán. Căn cứ vào việc mua, bán chứng khoán lần đầu hay mua bán lại, thị trường chứng khoán được chia làm hai bộ phận: thị trường sơ cấp và thị trường thứ cấp. Thị trường chứng khoán được chia thành 2 loại đó là thị trường sơ cấp và thị trường thứ cấp. Thị trường sơ cấp là nơi cổ phiếu lần đầu phát hành từ công ty để thu hút một nguồn vốn đầu tư, điều này giúp họ có thể huy động một số vốn trên thị trường chứng khoán. Phần lớn những người mua trên thị trường sơ cấp là các tổ chức lớn hay quỹ đầu tư. Với thị trường chứng khoán thứ cấp, cổ phiếu được mua bán lại sau khi phát hành. Người mua tại thị trường sơ cấp sẽ tiến hành mua bán đối với các nhà đầu tư chứng khoán khác trên thị trường. Chính vì thế sẽ không có tiền mới được sinh ra mà chỉ là thay đổi quyền sở hữu cổ phiếu giữa người mua và bán. Đây cũng là nơi các nhà đầu tư cá nhân có thể tham gia giao dịch chứng khoán.

Giá trị cổ phiếu biến động mạnh trên thị trường thứ cấp vì nó chịu tác động của nhiều nhân tố, đặc biệt là kết quả kinh doanh của công ty. Nếu công ty gặp khó khăn hay làm ăn thua lỗ, cổ tức có thể rất thấp hoặc không có cổ tức. Khi công ty phát hành cổ phiếu rơi vào phá sản, cổ đông sẽ nhận được những giá trị còn lại của các tài sản thanh lý.

### **2.3.2 Phân tích kỹ thuật**

Ngay từ khi xuất hiện thị trường chứng khoán, nhiều nhà đầu tư đã tìm cách để phân tích xu hướng của thị trường một cách khoa học tạo ra các trường phái lý luận khác nhau. Mỗi trường phái đều có những ưu nhược điểm riêng và đều đưa ra những lý luận chặt chẽ để bảo vệ quan điểm của mình.

Phân tích kỹ thuật là một phương pháp nghiên cứu chuyển động của cổ phiếu và dữ liệu giao dịch để dự đoán giá cả trong tương lai, ngược lại với việc nghiên cứu dữ liệu tài chính hoặc tin tức của nó. Những nhà đầu tư theo trường phái phân tích kỹ thuật cho rằng dữ liệu giao dịch của cổ phiếu phản ánh đầy đủ những thông tin của cổ phiếu ở hiện tại và trong một tương lai gần. Dữ liệu giao dịch có thể sẽ chứa những mẫu biến đổi được lặp lại nhiều lần và nếu có thể nhận thấy dấu hiệu của các mẫu này thì có thể dự đoán được tương lai. Dữ liệu giao dịch cũng cho thấy tâm lý của đám đông trong thị trường. Nếu số khối lượng giao dịch tăng nhanh thì cũng cho thấy thị trường đang lạc quan.

Lấy ví dụ đơn giản về phân tích kỹ thuật, tháng 6 năm 2020 trong hình 2.1, cổ phiếu TSLA mở cửa ở mức 858 USD / cổ phiếu và giá đã từng giảm 0,45% xuống



854,1 USD / cổ phiếu, đây là mức giá thấp nhất trong ngày. Sau đó, cổ phiếu được giao dịch tăng 7,6% so với ngày hôm trước, lên 899 USD / cổ phiếu và đóng cửa ngày giao dịch hôm đó ở mức 898,1 USD / cổ phiếu. Chúng ta có thể thấy rằng người mua sẵn sàng trả ngày càng nhiều giá hơn cho mỗi cổ phiếu. Những người bán không sẵn sàng bán cổ phiếu của họ ở cùng một mức giá, thay vì đòi hỏi giá ngày càng nhiều hơn. Mô hình giao dịch này có thể được coi là thị trường thống nhất hơn ở một vị thế mua (có nghĩa là đi lên). Khối lượng giao dịch 14.939.500 cổ phiếu trong ngày hôm đó cũng có thể cho thấy tâm trạng của công chúng đang lạc quan hơn là thận trọng.

**Tesla, Inc. (TSLA)**  
NasdaqGS - NasdaqGS Real Time Price. Currency in USD [Add to watchlist](#)

**885.66** +21.28 (+2.46%)  
At close: June 5 4:00PM EDT

Summary Company Outlook [Chart](#) Conversations Statistics **Historical Data** Profile Financials Analysis Options

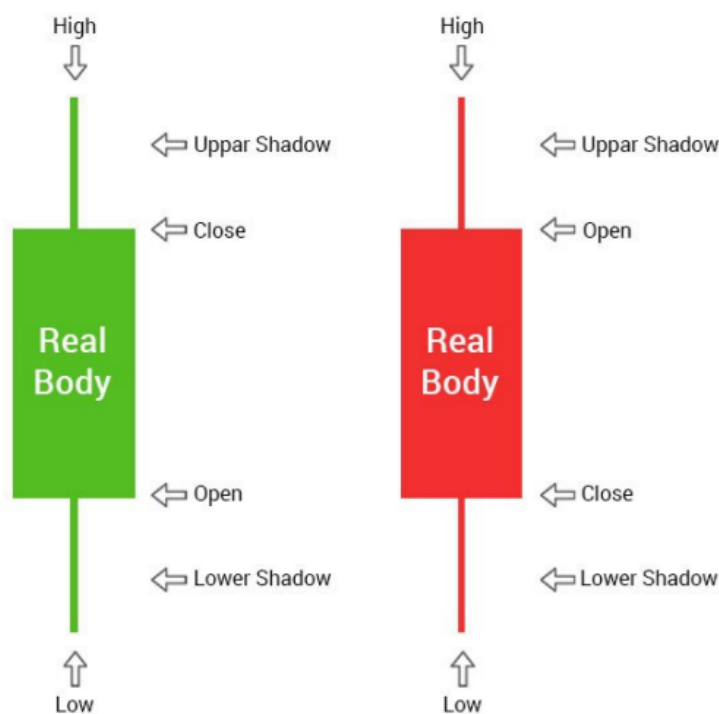
Time Period: Jun 29, 2010 - Jun 07, 2020 Show: Historical Prices Frequency: Daily [Apply](#)

Currency in USD [Download](#)

Date	Open	High	Low	Close*	Adj Close**	Volume
Jun 05, 2020	877.84	886.52	866.20	885.66	885.66	7,796,100
Jun 04, 2020	889.88	895.75	858.44	864.38	864.38	8,887,700
Jun 03, 2020	888.12	897.94	880.10	882.96	882.96	7,949,500
Jun 02, 2020	894.70	908.66	871.00	881.56	881.56	13,565,600
Jun 01, 2020	858.00	899.00	854.10	898.10	898.10	14,939,500
May 29, 2020	808.75	835.00	804.21	835.00	835.00	11,812,500

**Hình 2.1:** Dữ liệu giao dịch lịch sử của công ty Tesla cung cấp bởi Yahoo Finance

Thông thường, các nhà môi giới trực tuyến cung cấp phần mềm giao dịch cho các nhà đầu tư với một số chỉ báo kỹ thuật được tích hợp sẵn để thực hiện phân tích kỹ thuật. Một trong những nền tảng cho điều này là biểu đồ thanh nến, được thể hiện trong hình 2.2. Biểu đồ kết hợp thông tin về giá cổ phiếu và trông giống như một cây nến, do đó có tên là biểu đồ thanh nến. Ở bên trái, chúng ta có một cây nến xanh tương ứng giá cổ phiếu đi lên. Phần được tô màu xanh lá cây được gọi là phần thân. Thanh ở trên và dưới được gọi là bóng trên và bóng dưới. Điểm thấp nhất và điểm cao nhất cho biết giá thấp nhất và giá cao nhất trong một thời kỳ nhất định. Phần trên và phần dưới của phần thân biểu thị giá đóng và mở trong một khoảng thời gian nhất định. Ở bên phải, có một nến màu đỏ tương ứng giá cổ phiếu đi xuống. Nó hoạt động tương tự như một cây nến xanh, ngoại trừ một điểm khác biệt: phần trên và phần dưới của thân cho biết giá mở và đóng cửa trong một khoảng thời gian nhất định.



**Hình 2.2:** Biểu đồ thanh nến

Hình 2.3 cho thấy biến động giá cổ phiếu trong vòng 39 ngày giao dịch bằng cách sử dụng 39 biểu đồ nến, mỗi biểu đồ đại diện cho 1 ngày giao dịch. Ngoài ra còn có các đường có màu sắc khác nhau, chẳng hạn như xanh lá cây, xanh lam, tím, cam, lục lam đại diện cho SMA (đường trung bình động đơn giản, xem chương 2.5.1)

Ngoài ra còn có các biểu đồ đánh dấu hàng tuần, biểu đồ đánh dấu hàng tháng, v.v. cho từng chứng khoán, vì chứng khoán đó đã được liệt kê đủ lâu. Đối với đầu tư ngắn hạn, xu hướng thường được phân tích bằng cách sử dụng biểu đồ đánh dấu 1 ngày. Có nhiều lý thuyết về việc vẽ đường hỗ trợ và đường kháng cự, cũng như cách sử dụng chúng để dự đoán xu hướng. Một cách để làm điều đó được thể hiện trong hình 2.3. Nó thể hiện một phân tích kỹ thuật đơn giản. Cực đại và cực tiểu cục bộ được xác định và chúng tạo thành một hình bình hành được gọi là kênh xu hướng [39, trang 40]. Điều đó có nghĩa là cổ phiếu có khả năng cao sẽ di chuyển vào bên trong kênh. Nếu giá đóng cửa của cổ phiếu là trên đường trung bình động (n) thì người ta tin rằng cổ phiếu sẽ tăng giá trong khoảng thời gian n và ngược lại. Việc phá vỡ mức kháng cự cho thấy rằng cổ phiếu sắp đi lên và phá vỡ mức hỗ trợ cho thấy điều ngược lại. Trong trường hợp cụ thể này, chúng ta có thể quan sát rằng phân tích kỹ thuật đã dự đoán thành công chuyển động của cổ phiếu trong ngày hoặc vài ngày tới.

Các chỉ báo kỹ thuật, hoặc bộ dao động kỹ thuật, là các tín hiệu được tính toán toán học được sử dụng trong phân tích kỹ thuật để dự đoán biến động giá cổ phiếu trong tương lai. Một số chỉ báo kỹ thuật được sử dụng rộng rãi được trình bày dưới đây.



**Hình 2.3:** Giá cổ phiếu và các chỉ báo kỹ thuật

### 2.3.3 Đường trung bình động

Đường trung bình động là công cụ phân tích kỹ thuật rất phổ biến, với khả năng đơn giản hóa các biến động giá, tạo thành đường xu hướng cũng như dễ dàng diễn giải các hình dạng của nó. Đường trung bình động thường được sử dụng kèm với các chỉ báo khác để bổ sung các khiếm khuyết của nhau. Đường trung bình động có thể hỗ trợ trong việc tối giản hóa các biến động của thị trường. Đường trung bình động đơn giản (Simple Moving Average) là dạng đơn giản nhất của đường MA. Đường trung bình động đơn giản được dùng để xác định xu hướng giá cổ phiếu sẽ tiếp tục hay đảo chiều ngược lại, từ đó tạo ra các tín hiệu giúp nhà đầu tư nhận biết được thời điểm thích hợp để đưa ra quyết định mua vào và bán ra. Đường trung bình động đơn giản được tính bằng cách tính tổng tỷ giá của một khoảng thời gian (ví dụ 20 nến trước đó) và chia cho số nến (hay còn hiểu là giá trung bình cộng của khoảng thời gian đó).

$$SMA = \frac{A_1 + A_2 + \dots + A_n}{n} \quad (2.1)$$

### 2.3.4 Bollinger Bands

Bollinger bands là một chỉ báo được phát minh bởi John Bollinger – nhà phân tích tài chính tài ba trên thế giới vào đầu năm 1980. Chỉ báo này cũng được đặt theo tên của nhà phát minh là Bollinger (dải Bollinger). Chỉ báo này được cấu tạo từ đường trung bình động MA và độ lệch chuẩn giá. Cụ thể, dải Bollinger gồm 3 phần chính, một đường trung bình động chu kỳ 20 ngày (SMA20) là dải nằm giữa và hai dải di động được đặt bên trên và bên dưới SMA20. Khi thị trường biến động mạnh, hai dải trên và dưới sẽ mở rộng ra, ngược lại, khi sự biến động giảm, độ rộng của hai đường sẽ thu hẹp lại.

Thông thường các nhà đầu tư sử dụng chỉ báo Bollinger Bands để xác định xu

hướng của thị trường, dự đoán về khả năng tiếp tục hay dừng lại xu hướng đó. Ngoài ra, Bollinger Bands cũng giúp trader xác định thị trường có đang trong giai đoạn đi ngang, hay đang bắt đầu cho một giai đoạn tích lũy. Dựa vào xu hướng này các nhà đầu tư sẽ đưa ra chiến lược giao dịch phù hợp.

$$MB = SMA(n) \quad (2.2)$$

$$UB = SMA(n) + m * \sigma(n) \quad (2.3)$$

$$LB = SMA(n) - m * \sigma(n) \quad (2.4)$$

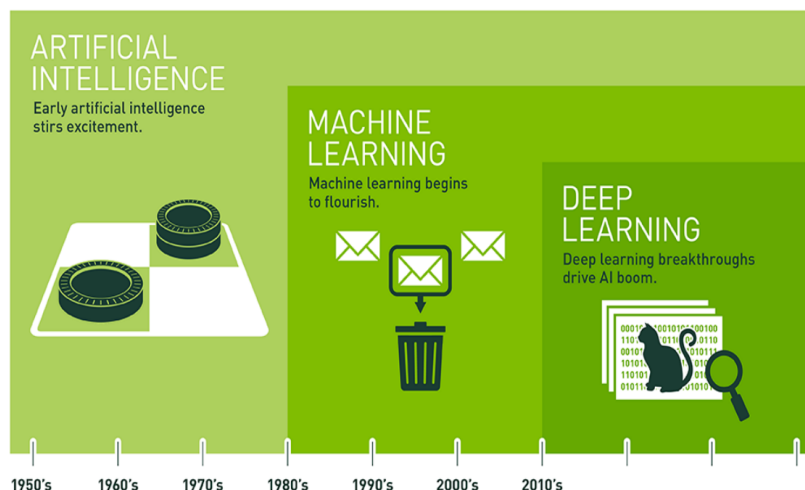
Trong đó: SMA là đường trung bình động đơn giản và n thường là 20  
UB là dải trên  
LB là dải dưới  
 $\sigma(n)$  là độ lệch của giá trong n ngày  
m là hằng số độ lệch thường bằng 2

## 2.4 Học máy

Học máy là một lĩnh vực con của trí tuệ nhân tạo tập trung vào việc dạy cho máy tính thực hiện một tác vụ cụ thể, chẳng hạn như nhận dạng chữ viết tay, phân loại ảnh mèo hoặc chó, tái tạo hình ảnh và dự báo tương lai gần, để bắt chước các kỹ năng nhất định của con người. Bên cạnh đó, nó cũng có thể được sử dụng để giải quyết các vấn đề thường khó đối với con người hoặc đòi hỏi chuyên môn cao. Ví dụ bác sĩ có thể mất nhiều năm kinh nghiệm để có thể xác định khối u xấu bằng cách xem hình ảnh X-quang, nhưng một chương trình máy tính có thể sử dụng kỹ thuật Máy học để đạt đến trình độ đó với thời gian đào tạo ngắn hơn nhiều. Ngay cả khi chương trình máy tính không thể thay thế bác sĩ, nó có thể giúp phát hiện ung thư sớm hơn và khả năng phát hiện tốt hơn (ít âm tính giả hơn). Nó cũng có thể được sử dụng để tìm các mô hình dữ liệu trong một lượng lớn dữ liệu mà con người khó thực hiện hoặc thực hiện trong thời gian rất lớn.

Hình 2.4 minh họa một tế bào thần kinh trong não người. Nó được kết nối với các tế bào thần kinh khác và có thể giao tiếp với chúng bằng cách sử dụng các chất hóa học. Học sâu và cụ thể là Mạng thần kinh nhân tạo (Artificial Neural Network) được lấy cảm hứng từ cách bộ não con người hoạt động cơ bản trong việc ra quyết định và tính toán. Cả hai đều được xây dựng từ các tế bào thần kinh đang chuyển tín hiệu và quyết định tín hiệu nào sẽ phát ra phía trước trong quá trình tính toán hoặc quyết định. Mạng nơ-ron được xây dựng từ các lớp khác nhau với các chức năng tính toán khác nhau. Một số ví dụ về các lớp mạng là lớp kết nối đầy đủ (fully connected), các lớp quên (dropout) và các lớp LSTM. Đối với mạng nơ-ron để đưa

ra quyết định chính xác hoặc thực hiện các chức năng cụ thể với độ chính xác, nó thường yêu cầu một lượng lớn dữ liệu huấn luyện mà từ đó nó sẽ học cách nhận ra các mô hình dữ liệu. Có nhiều thuật toán huấn luyện như Perceptron, Học sâu và Học tăng cường. Do Google khởi xướng, mạng nơ-ron đã được phát triển và cải tiến đáng kể trong vài năm qua và các ý tưởng sáng tạo mới liên tục xuất hiện và đạt được nhiều thành tựu to lớn.



**Hình 2.4:** Lược sử phát triển của Trí tuệ nhân tạo

## 2.5 Học sâu

### 2.5.1 Khái niệm

Những năm gần đây, khi mà khả năng tính toán của máy tính được nâng lên một tầm cao mới cùng với lượng dữ liệu khổng lồ được thu thập, Học máy đã tiến thêm một bước dài và Học sâu một lĩnh vực mới được ra đời. Học sâu là một kỹ thuật tiên tiến của học máy dựa trên các thuật toán mạng nơ-ron nhân tạo. Là một nhánh đầy triển vọng của Trí tuệ nhân tạo, Học sâu đã thu hút được sự chú ý lớn từ cộng đồng công nghệ. So với các kỹ thuật Học máy thông thường như máy vector hỗ trợ (SVM) và k-láng giềng gần nhất (kNN), các thuật toán Học sâu sở hữu lợi thế của việc học tính năng không giám sát, khả năng tổng quát hóa mạnh mẽ và khả năng đào tạo mạnh mẽ với dữ liệu lớn. Hiện tại, Học sâu đã được ứng dụng toàn diện trong các nhiệm vụ phân loại và dự đoán, thị giác máy tính, xử lý hình ảnh và âm thanh. Mặc dù Học sâu được phát triển trong lĩnh vực khoa học máy tính, nhưng các ứng dụng của nó đã thâm nhập vào các lĩnh vực đa dạng như y học, khoa học thần kinh, vật lý và thiên văn học, tài chính và ngân hàng. Học sâu được lấy cảm hứng từ mạng nơ-ron sinh học bao gồm nhiều nơ-ron liên kết với nhau. Mạng nơ-ron Học sâu, hoặc mạng nơ-ron nhân tạo, cố gắng mô phỏng hệ cách hệ thần kinh hoạt động thông qua sự kết hợp của dữ liệu đầu vào, các trọng số và các đại

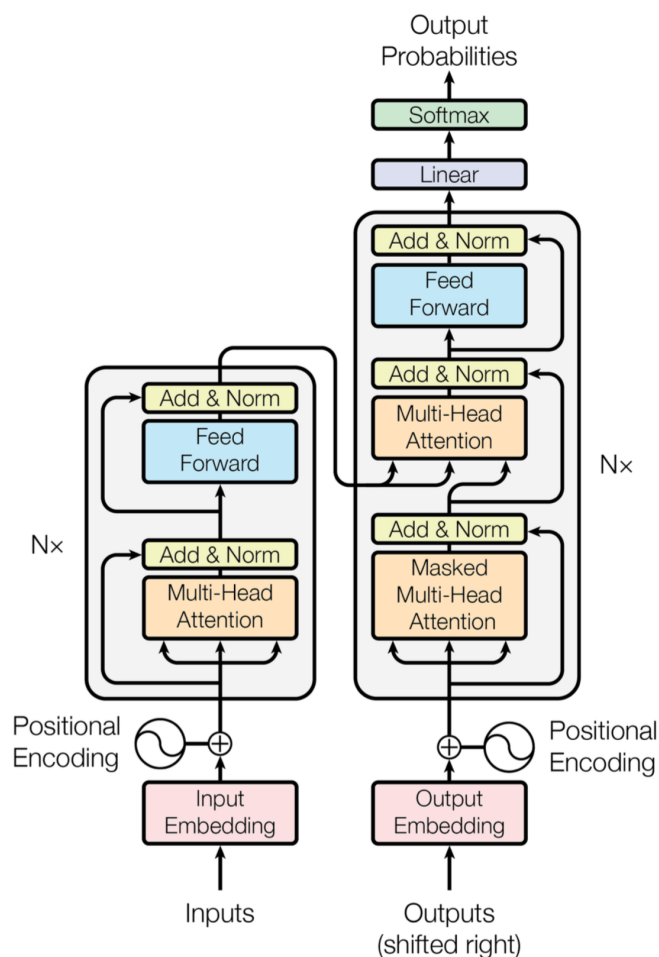
lượng hiệu chỉnh. Các yếu tố này làm việc cùng nhau để nhận biết, phân loại và mô tả chính xác các đối tượng trong dữ liệu.

Trong phần tiếp theo tôi sẽ trình bày các mô hình Học sâu được nghiên cứu trong đồ án này: Time2Vec và mô hình Transformer.

### **2.5.2 Transformer**

Transformer là một mô hình Học sâu được thiết kế để phục vụ giải quyết nhiều bài toán trong xử lý ngôn ngữ và tiếng nói, ví dụ như bài toán dịch tự động, bài toán sinh ngôn ngữ, phân loại, nhận dạng thực thể, nhận dạng tiếng nói, chuyển văn bản thành tiếng nói. Tuy nhiên, khác với RNNs, Transformer không xử lý các phần tử trong một chuỗi một cách tuần tự mà sử dụng cơ chế Self-Attention. Nếu dữ liệu đầu vào là một câu ngôn ngữ tự nhiên, Transformer không cần phải xử lý phần đầu câu trước rồi mới tới phần cuối câu. Do tính năng này, Transformer có thể tận dụng khả năng tính toán song song của GPU và giảm thời gian xử lý đáng kể.

Kiến trúc của mô hình Transformer được thể hiện trong hình 2.5. Tương tự như những mô hình dịch máy khác, kiến trúc của mô hình Transformer cũng bao gồm 2 thành phần là Encoder và Decoder. Encoder sẽ học cách biểu diễn câu thành một vector mong muốn rằng vector này mang đầy đủ thông tin của câu đó. Decoder có chức năng chuyển đổi vector biểu diễn của câu thành ngôn ngữ đích.

**Hình 2.5:** Mô hình Transformer

Khác với các mô hình xử lý tuần tự, mô hình Transformer có khả năng xử lý song song cho các từ, từ đó có khả năng tăng tốc độ huấn luyện mô hình. Lớp mã hóa của mô hình Transformer là một dạng mạng lan truyền tiến và có thể xử lý dữ liệu một cách song song. Hơn nữa các lớp Encoder có thể xếp chồng lên nhau tạo ra khả năng mã hóa cao. Do đó nó có thể tăng tốc độ học bằng cách tính toán song song. Nhờ sử dụng cơ chế tự chú ý mà không cần tuyến tính qua các nút mạng, mô hình Transformer còn có khả năng xử lý đầu vào theo 2 hướng mà không cần phải xếp chồng thêm một lớp nữa như trong kiến trúc Bidirectional LSTM.

Mô hình Transformer có thể xếp chồng nhiều lớp Encoder lên nhau để tăng hiệu quả mã hóa. Mỗi lớp Encoder sẽ bao gồm một lớp Multi-Head Attention, một lớp Feed Forward và một số lớp hiệu chỉnh khác. Đầu vào của mạng sẽ là ma trận biểu diễn của các từ trong câu được cộng với thông tin vị trí đã được mã hóa thông qua một lớp Positional Encoding. Sau đó, ma trận này sẽ được đưa qua một lớp Multi-Head Attention để học các phụ thuộc của đầu vào. Multi-Head Attention thật chất là Self-Attention, nhưng mà để mạng có thể có chú ý nhiều mô hình dữ liệu khác

nhau, tác giả đơn giản là sử dụng nhiều Self-Attention.

Self-Attention là cơ chế giúp mô hình nhìn vào các từ khác trong lúc mã hóa một từ cụ thể, vì vậy, Transformer Encoder có thể hiểu được sự liên quan giữa các từ trong một câu, kể cả khi chúng có khoảng cách xa. Các Decoder cũng có kiến trúc giống như vậy nhưng giữa chúng có một lớp Attention để nó có thể tập trung vào các phần liên quan của đầu vào.

Có thể tưởng tượng cơ chế Self-Attention giống như cơ chế tìm kiếm. Với một từ cho trước, cơ chế này sẽ cho phép mô hình tìm kiếm trong cách từ còn lại, từ nào “giống” để sau đó thông tin sẽ được mã hóa dựa trên tất cả các từ trên. Đầu tiên, với mỗi từ chúng ta cần tạo ra 3 vector: Query, Key, Value bằng cách nhân ma trận biểu diễn các từ đầu vào với ma trận tham số tương ứng.

- Query: vector dùng để chứa thông tin của từ được tìm kiếm, so sánh. Vector này được sử dụng như một câu truy vấn khi bạn nhập từ khóa vào ô tìm kiếm của Google.
- Key: vector đại diện cho câu, được dùng để so sánh với Query ở trên để tìm ra câu phù hợp với Query. Ví dụ như các thẻ meta chứa các từ khóa của trang web, Google sẽ so sánh từ truy vấn Query với các từ khóa Key này và trả về kết quả có độ tương đồng cao nhất.
- Value: vector chứa đầy đủ thông tin của câu, mỗi vector đại diện cho 1 câu. Đây chính là nội dung của trang web mà người dùng tìm kiếm khi người dùng truy vấn bằng vector Query.

Để tính tương quan, chúng ta đơn giản chỉ cần tính tích vô hướng dựa vào các vector Query và Key. Sau đó dùng hàm softmax để chuẩn hóa chỉ số tương quan trong đoạn[0-1], và cuối cùng, tính trung bình cộng có trọng số giữa các vector Values sử dụng chỉ số tương quan mới tính được. Cơ chế Self-Attention được mô tả trong hình 2.6. Cho đầu vào của mạng là chuỗi thời gian dữ liệu giao dịch  $X = [x_1, x_2, \dots, x_n] \in R^{N \times F}$ . Quá trình tính toán các vector trọng số trên có thể được tóm tắt làm 3 bước như sau:

Bước 1: Tính ma trận Query, Key, Value. Trước hết, khởi tạo 3 ma trận trọng số Query, Key, Value, sau đó nhân input với các ma trận trọng số này để tạo thành 3 ma trận tương ứng.

$$Q = W^{(Q)}X; K = W^{(K)}X; V = W^{(V)}X \quad (2.5)$$

Trong đó  $W^{(Q)}$ ,  $W^{(K)}$ ,  $W^{(V)}$  là ma trận trọng số của Query, Key, Value.



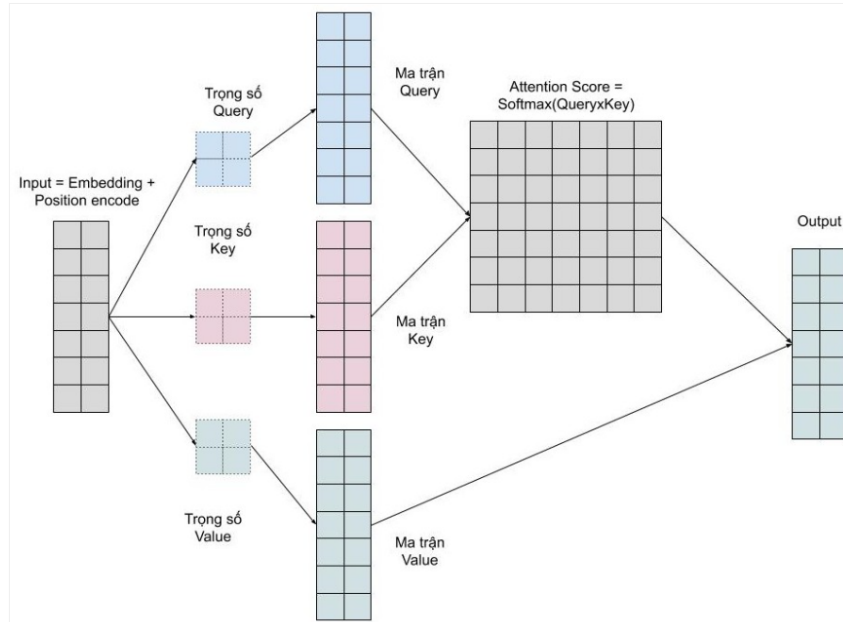
Bước 2: Tính ma trận trọng số chú ý  $a \in R^{N \times N}$  bằng cách nhân 2 ma trận Key, Query vừa được tính ở trên với nhau nhằm mục đích so sánh vector Query và Key để học sự phụ thuộc giữa các từ trong câu. Sau đó, ta chuẩn hóa giá trị của ma trận trọng số về trong đoạn [0-1] bằng cách sử dụng hàm softmax. Giá trị 1 có nghĩa là câu Query giống với Key, giá trị 0 có nghĩa là không giống.

$$a = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot M \quad (2.6)$$

Trong đó M là ma trận mặt nạ để xác định những giá trị nào sẽ chú ý đến nhau.

Bước 3: Tính output. Trọng số chú ý cho biết sự phụ thuộc giữa các từ ở trong câu, tiếp đó ta nhân trọng số chú ý với ma trận Value để thu được vector mã hóa của câu. Khi đó, các giá trị trong ma trận Value sẽ được đánh trọng số bởi ma trận trọng số từ đó các từ sẽ có thể chú ý đến nhau.

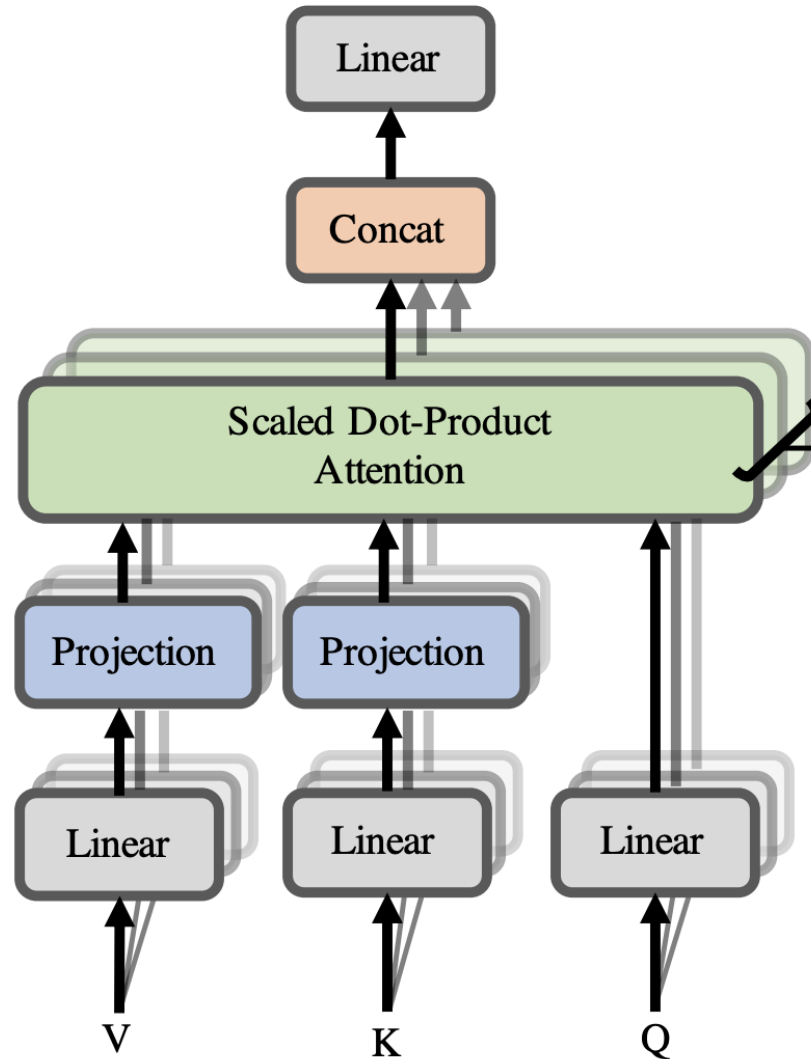
$$\text{Attention}(Q, K, V) = aV \quad (2.7)$$



**Hình 2.6:** Cơ chế Self-Attention

Mỗi lớp Self-Attention sẽ học được một kiểu pattern đặc trưng cho mối liên hệ giữa các từ trong câu với nhau. Để mở rộng khả năng này, tức là muốn mô hình học được nhiều mối liên hệ giữa các từ hơn, cách đơn giản là thêm nhiều lớp Self-Attention. Tức là chúng ta cần nhiều ma trận Query, Key, Value. Hay nói cách khác

ma trận trọng số Key, Query, Value sẽ có thêm 1 chiều depth nữa và depth là số lớp Self-Attention. Cơ chế này được tác giả đặt tên là Multi-Head Attention. Multi-Head Attention cho phép mô hình chú ý đến đồng thời những pattern dễ quan sát được như chú ý đến những từ phía trước 1 từ, chú ý đến những từ phía sau 1 từ, chú ý đến những từ liên quan,...Kiến trúc Multi-Head Attention được mô tả như hình 2.7



Hình 2.7: Cơ chế Multi-Head Attention

### 2.5.3 Time2Vec

Trong việc xây dựng các mô hình học sâu, thời gian thường là một thuộc tính quan trọng. Đầu vào cho các vấn đề liên quan đến thời gian có thể được coi là một chuỗi trong đó, thay vì được phân phối giống nhau và độc lập, tồn tại sự phụ thuộc theo thời gian và không gian giữa các điểm dữ liệu. Chuỗi các điểm dữ liệu này có thể là đồng bộ, tức là được lấy mẫu theo các khoảng thời gian đều đặn hoặc không đồng bộ, tức là được lấy mẫu tại các thời điểm khác nhau. Trong cả hai trường hợp,

thời gian có thể là một đặc trưng quan trọng.

Mô hình Transformer xử lý dữ liệu một cách song song, do đó, những đặc trưng về thời gian của dữ liệu bị mất đi khi sử dụng mô hình Transformer. Để sử dụng các thông tin về thời gian của dữ liệu, người ta thường coi thời gian như là một đặc trưng, nó thường được đưa vào như một chiều của điểm dữ liệu. Để giúp sử dụng thời gian tốt hơn, một số nhà nghiên cứu thiết kế các tính năng thời gian được làm thủ công phù hợp với vấn đề cụ thể của họ và đưa các tính năng đó vào dữ liệu. Tuy nhiên, các tính năng thủ công có thể tốn kém và đòi hỏi chuyên gia về vấn đề này. Do đó mô hình Time2Vec ra đời để giải quyết vấn đề đưa những thông tin về thời gian vào các mô hình học sâu như một đặc trưng.

Mô hình Time2Vec được đề xuất trong bài báo Time2Vec: Learning a vector Representation of Time [2]. Trong bài báo, tác giả chia đặc trưng về thời gian thành 2 loại: 1-sự kiện xảy ra theo chu kỳ, 2-sự kiện xảy ra không theo chu kỳ. Sự kiện xảy ra theo chu kỳ như số lượng bán hàng của một cửa hàng có thể cao hơn vào cuối tuần hoặc ngày lễ. Thời tiết thường tuần hoàn theo các mùa khác nhau trong năm. Sự kiện xảy ra không theo chu kỳ là những sự kiện có xác suất cao xảy ra trong khoảng thời gian như một số bệnh có nhiều khả năng xảy ra đối với lứa tuổi lớn hơn. Các mẫu có chu kỳ và không có chu kỳ như vậy giúp phân biệt thời gian với các đặc trưng khác đồng thời mã hóa tốt hơn các đặc trưng về thời gian. Điều quan trọng là sử dụng một phép biến đổi cho phép biểu diễn các mẫu thời gian tuần hoàn và không tuần hoàn.

Để mô hình hóa các đặc trưng về thời gian, tác giả đề xuất sử dụng một hàm biến đổi và đặt tên cho vector mô hình hóa thời gian này là Time2Vec:

$$t2v(\tau)[i] = \begin{cases} \omega_i\tau + \phi_i & i = 0 \\ \sin(\omega_i\tau + \phi_i) & i \neq 0 \end{cases} \quad (2.8)$$

Trong đó:  $t2v(\tau)[i]$  là phần tử thứ  $i$  của vector  $t2v(\tau)$   
 $\omega_i\tau$  và  $\phi_i$  là các tham số mô hình cần học

Như vậy thời gian tại mỗi điểm dữ liệu sẽ được biểu diễn bằng một hàm tuyến tính  $\omega_0\tau + \phi_0$  và  $k - 1$  hàm sin có dạng  $\sin(\omega_i\tau + \phi_i)$  trong đó  $\omega_i$  và  $\phi_i$  là các tham số mô hình cần học. Bằng cách sử dụng hàm biến đổi trên và thêm vào sau một lớp tuyến tính thì vector đầu ra sẽ có dạng biến đổi Fourier với chuỗi hàm sin, theo đó bất kỳ hàm một biến nào cũng có thể được xấp xỉ bởi tổng một chuỗi hàm sin nhân với trọng số. Giả sử vector Time2Vec có dạng  $t2v = [\omega_0\tau + \phi_0; \sin(\omega_1\tau + \phi_1); \dots; \sin(\omega_k\tau + \phi_k)]$  và lớp tuyến tính được thêm vào sau có dạng

$A = [a_0; a_1; \dots; a_k]$ . Khi đó đầu ra của mô hình Time2Vec sẽ có dạng:

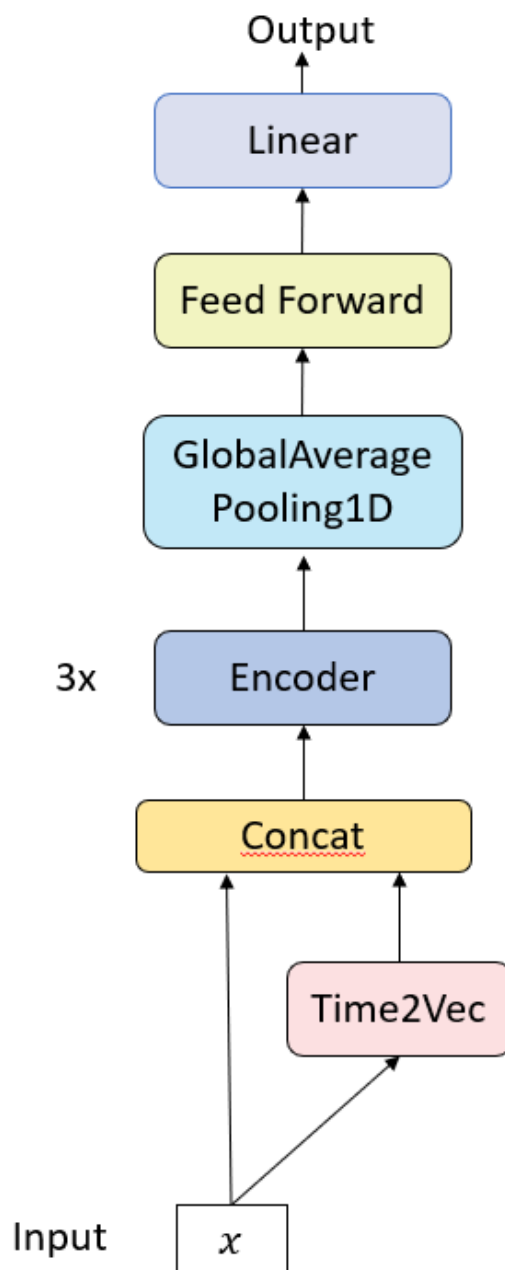
$$O = t2v.A^T = a_0(\omega_0\tau + \phi_0) + a_1 \sin(\omega_1\tau + \phi_1) + \dots + a_k \sin(\omega_k\tau + \phi_k) \quad (2.9)$$

Các trọng số khác nhau của lớp tuyến tính phía sau được học bởi mô hình sẽ kết hợp với hàm biến đổi và tạo ra các chuỗi Fourier xấp xỉ gần đúng với đặc trưng về thời gian. Việc sử dụng hàm sin có thể giúp mã hóa các thời gian theo chu kỳ. Vector đặc trưng thời gian sẽ được nối vào vector điểm dữ liệu để đưa vào mô hình  $[x, \sin(\omega_1\tau + \phi_1); \dots; \sin(\omega_k\tau + \phi_k)]$ .

## CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

Trong chương trước, tôi đã trình bày về các cơ sở lý thuyết, các khái niệm được sử dụng trong nghiên cứu này. Chương này sẽ trình bày cách thức nghiên cứu được thực hiện và mô hình đề xuất của tôi. Tổng quan về mô hình đề xuất được trình bày trước tiên, sau đó là phần giải thích về phương pháp tiền xử lý dữ liệu và chi tiết về mô hình đề xuất.

### 3.1 Kiến trúc mạng đề xuất



**Hình 3.1:** Kiến trúc của mô hình đề xuất

Hình 3.1 thể hiện kiến trúc của mô hình đề xuất. Dữ liệu ban đầu của mô hình là 5 chỉ số: giá đóng cửa, giá mở cửa, giá cao nhất, giá thấp nhất và khối lượng giao dịch của  $n$  ngày liên tiếp. Mạng sẽ cố gắng học các phụ thuộc của dữ liệu để cho output đầu ra là giá đóng cửa của ngày tiếp theo tức ngày thứ  $n+1$ . Đầu tiên, lớp Input nhận dữ liệu đã biến đổi về phần trăm thay đổi và qua tiền xử lý như đầu vào của mạng. Sau đó dữ liệu được đưa qua một lớp Time2Vec để thu được vector đặc trưng thời gian. Từ 5 đặc trưng được đưa vào, Time2Vec cho ra kết quả là một vector một chiều có kích thước là 2 và được nối vào đầu vào. Như vậy, từ dữ liệu có kích thước  $(L, n, 5)$  sau khi qua mã hóa đặc trưng thời gian ta thu được bộ dữ liệu có kích thước  $(L, n, 7)$  trong đó có 2 chiều về thời gian được thêm vào. Tiếp theo, 3 lớp Transformer Encoder liên tiếp nhau được sử dụng để học mối liên hệ giữa các đặc trưng. Sau đó, một lớp GlobalAveragePooling1D được sử dụng để tính giá trị trung bình trong mỗi timestep. Lớp GlobalAveragePooling1D có tác dụng giảm kích thước của dữ liệu và giảm thời gian tính toán. Cuối cùng là một mạng Feed Forward để đưa ra dự đoán về tỷ lệ tăng hoặc giảm giá cổ phiếu trong ngày tiếp theo.

Tiếp theo tôi sẽ đi sâu vào giải thích các thành phần của mô hình và các cải tiến được đề xuất áp dụng.

## 3.2 Tiền xử lý dữ liệu

### 3.2.1 Chuyển đổi dữ liệu thành phần trăm thay đổi

Cung cấp dữ liệu giá thô vào mô hình học sâu thường là một ý tưởng tồi do giá của cổ phiếu phụ thuộc vào đơn vị tiền tệ. Khi nhìn vào biểu đồ giá của IBM trong hình 3.2, bạn có thể thấy giá từ năm 2005 đến năm 2008 (50– 100) ở một mức hoàn toàn khác so với giá của những năm 2020 và 2022 (140– 220). Về bản chất, hai khoảng giá này không liên quan gì đến nhau. Trong 2 khoảng thời gian này, giá cổ phiếu năm 2022 cao hơn nhiều năm 2008 không đồng nghĩa với công ty phát triển mạnh hơn hay giá trị thực của cổ phiếu tăng bởi vì sức mua của đồng đô la ở hai thời điểm này khác nhau.

Trong dự đoán giá cổ phiếu, mức độ tăng hay giảm bao nhiêu phần trăm thường sẽ được quan tâm hơn giá thị trường của cổ phiếu. Mức độ tăng hay giảm của cổ phiếu sẽ ảnh hưởng trực tiếp đến tỷ suất lợi nhuận của nhà đầu tư - điều mà nhà đầu tư mong muốn. Do đó, để đưa các điểm giá trong quá khứ ngang bằng với điểm giá trong hiện tại và hữu ích hơn cho việc đào tạo mạng nơ-ron, tôi chuyển đổi các đặc trưng đã chọn của một ngày thành phần trăm thay đổi so với ngày trước đó. Việc biến đổi này có hai lợi ích rõ rệt: Thứ nhất biến đổi này đưa giá trong quá khứ và hiện tại về trong cùng một khoảng để giá không còn phụ thuộc vào sức mua của

tiền tệ. Thứ 2 biến đổi này giúp thể hiện rõ hơn xu hướng tăng giảm của cổ phiếu và giúp huấn luyện mô hình hiệu quả hơn.



**Hình 3.2:** Dữ liệu cổ phiếu IBM thu thập từ thư viện Yfinance

Theo đó tại ngày thứ  $t+1$  dữ liệu sẽ được biến đổi thành:

$$x_{t+1} = \frac{X_{t+1} - X_t}{X_t} \quad (3.1)$$

Trong đó:  $x_{t+1}$  là vector điểm dữ liệu sau khi biến đổi  
 $X_t$  là vector điểm dữ liệu trước khi biến đổi

### 3.2.2 Moving Average

Mục tiêu của đề tài là phát hiện các xu hướng của cổ phiếu tuy nhiên dữ liệu theo ngày thường có nhiều nhiễu do biến động giá. Để giảm nhiễu và làm biểu đồ mượt hơn, một kỹ thuật thường được sử dụng là đường trung bình động. Nhìn vào hướng của đường trung bình có thể thấy được cơ bản về cách giá đang di chuyển. Nếu nó nghiêng lên, giá đang trong tăng (hoặc gần đây), nếu nó nghiêng xuống và giá đang giảm, đi ngang và giá có khả năng biến động ổn định trong 1 phạm vi. Đường trung bình động (Moving Average) là một công cụ phân tích kỹ thuật đơn giản giúp làm mịn dữ liệu giá bằng cách tạo ra mức giá trung bình. Mức trung bình được lấy trong một khoảng thời gian cụ thể, như 10 ngày, 20 phút, 30 tuần hoặc bất kỳ khoảng thời gian nào mà nhà giao dịch chọn.

Tuy nhiên, điểm bất lợi của việc áp dụng hiệu ứng trung bình động là tập dữ liệu mới không phản ánh dữ liệu ban đầu nữa. Đồng thời, xu hướng của đường trung bình động sẽ thay đổi chậm hơn xu hướng giá thực, điều này khiến cho những dự đoán dựa trên đường trung bình động có thể chậm hơn so với thay đổi thực tế của giá.

### 3.3 Time2Vec

Khi xử lý dữ liệu dạng chuỗi thời gian, thời gian là một đặc trưng rất quan trọng. Tuy nhiên trong kiến trúc Transformer, các chuỗi dữ liệu được chuyển tiếp cùng

một lúc qua mô hình, gây khó khăn trong việc trích xuất các đặc trưng về thời gian. Do đó Transformer có xu hướng sử dụng thêm các cơ chế khác mã hóa đặc trưng về thời gian trước khi đưa vào mô hình để cung cấp các thông tin về thời gian hay thứ tự cho mô hình. Trong bài báo gốc [12] tác giả sử dụng cơ chế Position Embedding để mã hóa các đặc trưng vào vị trí cho mô hình. Tuy nhiên cơ chế này chỉ mã hóa vị trí của một từ trong từ điển mà không mã hóa được vị trí của từ toàn bộ ngữ cảnh của văn bản đầu vào.

Trong dự đoán giá cổ phiếu đặc trưng về thời gian là không thể thiếu. Nếu không có thông tin về thứ tự hay thời gian, giá cổ phiếu của năm 2020 có thể ảnh hưởng đến giá cổ phiếu của năm 2008, điều này là không thực tế. Do đó cần sử dụng một phương pháp khác để mã hóa thông tin về thời gian trước khi đưa vào huấn luyện mô hình.

Để mã hóa các đặc trưng về thời gian, tôi sử dụng phương pháp được giới thiệu trong bài báo Time2Vec: Learning a Vector Representation of Time [18]. Ý tưởng chính trong bài báo là chia đặc trưng về thời gian thành 2 loại có chu kỳ và không có chu kỳ. Không có chu kỳ là những đặc trưng có xác suất cao xảy ra trong một khoảng thời gian. Các đặc trưng này được trích xuất qua một phép biến đổi tuyến tính  $t2v(\tau) = \omega_i \tau + \phi_i$ . Có chu kỳ là những đặc trưng có tính lặp đi lặp lại theo thời gian như 4 mùa trong 1 năm. Loại thời gian này được biểu diễn bằng một hàm tuần hoàn  $t2v(\tau) = \sin(\omega_i \tau + \phi_i)$ . Tác giả đặt tên cho vector biểu diễn đặc trưng về thời gian này là Time2Vec:

$$t2v(\tau)[i] = \begin{cases} \omega_i \tau + \phi_i & i = 0 \\ \sin(\omega_i \tau + \phi_i) & i \neq 0 \end{cases} \quad (3.2)$$

Trong đó:  $t2v(\tau)[i]$  là phần tử thứ  $i$  của vector  $t2v(\tau)$   
 $\omega_i \tau$  và  $\phi_i$  là các tham số mô hình cần học

Chi tiết về phương thức này đã được trình bày trong 2.5.3.

Trong nghiên cứu này, tôi sử dụng vector  $t2v$  có kích thước 2 để mô hình hóa đặc trưng về thời gian. Trong đó, thành phần đầu tiên của vector là thành phần tuyến tính dùng để biểu diễn đặc trưng thời gian không có chu kỳ. Thành phần thứ 2 là hàm tuần hoàn sin dùng để biểu diễn đặc trưng thời gian có chu kỳ.

Dữ liệu đầu vào là dữ liệu giao dịch của  $n$  ngày liên tiếp  $X \in R^{L \times F}$  với  $L$  là số timestep và  $F = 5$  là số đặc trưng (giá đóng cửa, mở cửa, cao nhất, thấp nhất và khối lượng giao dịch) được biến đổi về vector một chiều  $x_t = R^L$  bằng cách thay 5 đặc trưng bằng trung bình của chúng. Sau đó mô hình khởi tạo các trọng số  $\omega_0, \phi_0$



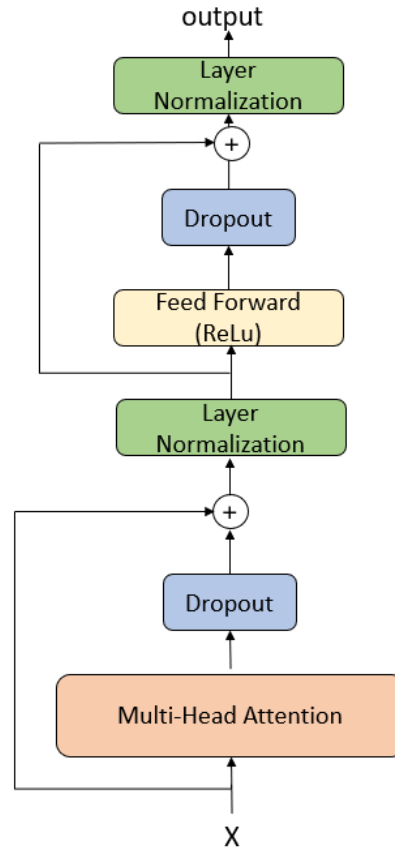
tương ứng với thành phần tuyến tính  $t2v(x)[0] = \omega_0 x + \phi_0$  và  $\omega_1, \phi_1$  tương ứng với thành phần tuần hoàn  $t2v(x)[1] = \sin(\omega_1 x + \phi_1)$ . Hai thành phần này được ghép với nhau tạo thành vector đặc trưng thời gian  $t2v = [\omega_0 x_t + \phi_0; \sin(\omega_1 x_t + \phi_1)]$ .

Như vậy sau lớp Time2Vec, đặc trưng về thời gian của mỗi timestep ban đầu sẽ được biểu diễn bằng vector  $t2v \in R^{L \times 2}$ . Sau đó các đặc trưng về thời gian được thêm vào mỗi timesteps của dữ liệu ban đầu. Vector dữ liệu có dạng  $[X; \omega_1 \tau + \phi_1; \sin(\omega_2 \tau + \phi_2)]$  và có số đặc trưng là 7.

### 3.4 Transformer Encoder

Sau khi mô hình hóa các đặc trưng thời gian bằng mô hình Time2Vec, bước tiếp theo là học các mẫu dữ liệu bằng mô hình Transformer Encoder. Transformer là một kiến trúc mạng nơ-ron sử dụng cơ chế Self-Attention, cho phép mô hình tập trung vào các phần có liên quan đến nhau của chuỗi dữ liệu để cải thiện chất lượng dự đoán. Cơ chế Self-Attention cho phép kết nối tất cả các bước trong chuỗi dữ liệu với nhau cùng một lúc, dẫn đến mô hình có thể ghi nhớ được sự phụ thuộc lâu dài. Tất cả các quá trình này được song song hóa trong kiến trúc Transformer, cho phép tăng tốc quá trình học.

Trong đề án này, tôi sử dụng 3 lớp Encoder xếp chồng lên nhau để học sự phụ thuộc của các đặc trưng. Mỗi lớp Encoder sẽ bao gồm một lớp Multi-Head Attention và một lớp Feed Forward. Lớp Multi-Head Attention bao gồm 8 head, mỗi head là một cơ chế Self-Attention riêng biệt. Với mong muốn rằng mỗi head sẽ học được một sự phụ thuộc khác nhau trong tập dữ liệu để tăng hiệu quả tự chú ý của mô hình. Lớp Feed Forward sẽ bao gồm một lớp ẩn với hàm kích hoạt ReLU và một lớp tuyến tính để biến đổi đầu ra của Multi-Head Attention về số chiều như đầu vào ban đầu. Trước và sau lớp Feed Forward sẽ là 2 lớp Dropout và Normalization để chuẩn hóa tham số và tăng hiệu quả học của mô hình. Kiến trúc của lớp Transformer Encoder đề xuất được mô tả trong hình 3.3.

**Hình 3.3:** Cấu trúc lớp Encoder

Đầu vào của lớp Encoder đầu tiên bao gồm các đặc trưng của dữ liệu giao dịch và 2 đặc trưng thời gian. Giả sử đầu vào là dữ liệu chuỗi thời gian  $X = [x_1, x_2, \dots, x_n] \in R^{N \times F}$ . Đầu tiên tôi tính toán các vector Query, Key, Value của lớp Multi-Head Attention như sau:

$$Q_h = W_h^{(Q)} X; K_h = W_h^{(K)} X; V_h = W_h^{(V)} X \quad (3.3)$$

Trong đó  $h = 1, \dots, H$  là chỉ số của head và  $W_h^{(Q)}, W_h^{(K)}, W_h^{(V)}$  là các ma trận tham số cần học.

Trọng số chú ý được tính toán và nhân với vector  $V_h$  tạo thành đầu ra của lớp Multi-Head Attention.

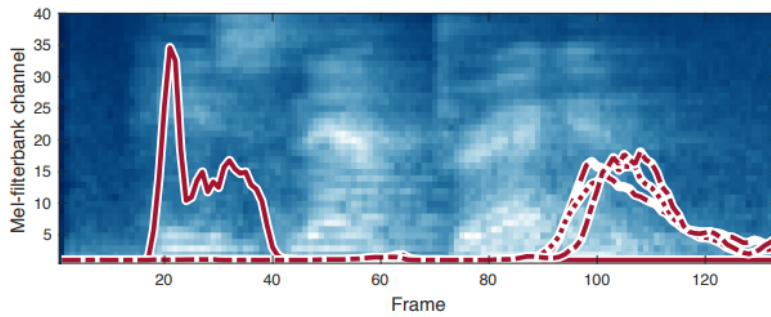
$$O_h = softmax\left(\frac{QK^T}{\sqrt{d_k}}.M\right)V_h \quad (3.4)$$

Sau đó đầu ra của các head được ghép lại với nhau thành vector  $O = [O_0, O_1, \dots, O_H]$  và được đưa qua một lớp tuyến tính như đầu ra của lớp Multi-Head Attention. Cuối

cùng, dữ liệu được truyền qua một mạng Feed Forward 1 lớp ẩn với hàm kích hoạt ReLU để tạo thành đầu ra của Encoder.

### 3.5 Hiệu chỉnh trực giao tham số

Sử dụng Multi-Head Attention cho phép mỗi head học một mô hình thời gian khác nhau trên các không gian con khác nhau. Tuy nhiên, một số nghiên cứu trước đây [15] [16] [17] chỉ ra rằng cơ chế Multi-Head Attention có xu hướng tạo các head dư thừa. Tức là phân bố đầu ra của các head thường tập trung về một số điểm làm giảm khả năng học của mô hình. Hình 3.4 là một ví dụ trong bài báo của Lee và cộng sự năm 2019 về trọng số chú ý của Multi-Head Attention trong đó 3 head có phân bố gần như trùng nhau.



**Hình 3.4:** Trọng số chú ý của cơ chế Multi-Head Attention

Tôi cho rằng để mạng có thể nắm bắt được các thông tin trong dữ liệu, các head của Multi-Head Attention nên chú ý vào các phần khác nhau của chuỗi dữ liệu đầu vào. Để đạt được điều đó tôi sử dụng một phương pháp trực giao hóa trọng số đầu ra  $W^V$  của vector  $V$  trong phương trình (3.3). Đầu tiên, tôi tính toán một vector các trọng số  $W^V = [W_1^V, W_2^V, \dots, W_H^V]$  bằng cách ghép các trọng số  $W_h^V$  của các head với nhau. Kích thước của  $W^V$  là  $H \times F \times d_v$  trong đó  $d_v$  là kích thước của  $V_h$ . Sau đó tôi làm phẳng vector  $W^V$  thành ma trận  $A$  có kích thước  $H \times (F \times d_v)$  và chuẩn hóa ma trận  $A$  thành  $\hat{A} = A / \|A\|_2$ . Hàm phạt được tính như sau:

$$L_p = \|\hat{A}\hat{A}^T - I\|_F \quad (3.5)$$

Trong đó  $\|\cdot\|$  là chuẩn Frobenius, là căn bậc hai của tổng bình phương các phần tử của ma trận và  $I$  là ma trận đơn vị. Sau đó thêm hàm phạt trọng số tất cả các head của các lớp Multi-Head Attention vào trong hàm mất mát của mô hình.

$$L = L_{CE} + \sum_{i=1}^{(B)} \gamma L_{i,p} \quad (3.6)$$

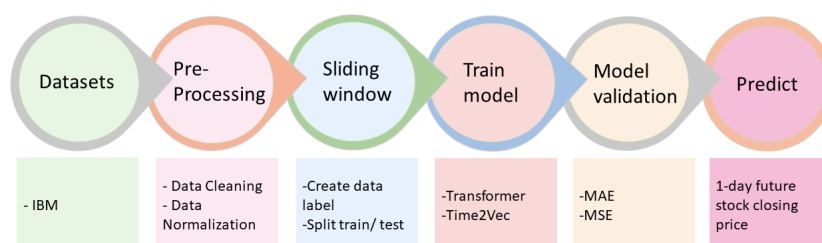
Trong đó B là số Encoder,  $\gamma$  là trọng số của hàm phạt và  $L_{CE}$  là hàm mất mát ban đầu.

## CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM

Trong chương này, tôi sẽ trình bày kết quả thực nghiệm của chương trình cài đặt mô hình đề xuất đã trình bày ở chương trước.

### 4.1 Tổng quan giải pháp

Quy trình thực hiện của đề án gồm 6 pha chính được mô tả trong hình 4.1. Đầu tiên, dữ liệu giao dịch cổ phiếu IBM sẽ được thu thập từ thư viện yfinance của python. Tiếp theo, tôi chọn 5 đặc trưng (giá đóng cửa, giá mở cửa, giá cao nhất, giá thấp nhất, khối lượng giao dịch) để đưa vào huấn luyện mô hình. Sau đó dữ liệu sẽ được biến đổi về phần trăm thay đổi, làm sạch và chuẩn hóa dữ liệu. Dữ liệu sau khi tiền xử lý sẽ được chia thành các timestep n ngày liên tiếp với nhãn là tỷ lệ tăng của giá đóng cửa cổ phiếu của ngày tiếp theo so với ngày liền trước đó. Sau quá trình này, tôi thu được dữ liệu phù hợp để đưa vào huấn luyện mô hình học sâu. Đầu ra của mô hình sẽ là tỷ lệ thay đổi giá đóng cửa của ngày tiếp theo so với ngày liền trước. Cuối cùng, tôi tiến hành đánh giá và thử nghiệm mô hình đã đào tạo trên các độ đo khác nhau.



Hình 4.1: Quy trình thực hiện

### 4.2 Môi trường phát triển

Tôi xây dựng mạng nơ-ron của mình bằng cách sử dụng thư viện Tensorflow. Việc phát triển và thử nghiệm được thực hiện bằng PyCharm, một IDE Python (Môi trường phát triển tích hợp). Thư viện Tensorflow cung cấp các loại "nơ-ron" cơ bản để chúng ta có thể tập trung vào việc thiết kế kiến trúc của mạng nơ-ron, điều chỉnh các siêu tham số và hiểu đặc điểm dữ liệu của bài toán. Quá trình xử lý dữ liệu cũng như phân tích kỹ thuật được thực hiện từ đầu bằng numpy, một thư viện cung cấp các chức năng xử lý dữ liệu dạng ma trận. Cuối cùng, dữ liệu đã được trực quan hóa bằng cách sử dụng thư viện Matplotlib.

Quá trình huấn luyện được thực hiện trên môi trường ảo của colab phiên bản

miễn phí với ram 12GB và GPU Tesla K80.

### 4.3 Tiền xử lý dữ liệu

#### 4.3.1 Thu thập dữ liệu

Trong nghiên cứu này, tôi sử dụng giá đóng cửa của cổ phiếu IBM trong 30 năm (từ ngày 01-01-1992 đến ngày 01-01-2022) để huấn luyện các mô hình và đánh giá kết quả. Bộ dữ liệu bao gồm 7549 ngày giao dịch với 5 yếu tố là giá đóng cửa, giá mở cửa, giá cao nhất, giá thấp nhất và khối lượng giao dịch. IBM là công ty có lịch sử lâu đời và là công ty công nghệ hàng đầu thế giới. Mọi biến động về công nghệ trên thế giới đều được phản ánh qua giá cổ phiếu của IBM. Cách dễ nhất để có được giá lịch sử của IBM là chỉ cần tải xuống bộ dữ liệu từ trang yahoo của IBM bằng thư viện Yfinance. Yfinance là một trong những thư viện nổi tiếng của Python về tài chính, được sử dụng để thu thập dữ liệu trực tuyến. Hình 4.2 hiển thị dữ liệu giao dịch của một số ngày thu thập được từ thư viện yfinance. Đối với mỗi ngày giao dịch, tôi sử dụng các thông tin giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa và Khối lượng giao dịch làm các đặc trưng được đưa vào mô hình. Mục đích của tôi là từ 5 yếu tố này để dự đoán giá đóng cửa ngày tiếp theo từ dữ liệu  $n$  ngày trước đó. Lý do sử dụng 5 yếu tố trong  $n$  ngày trước là vì mô hình sẽ hỗ trợ tìm ra sự phân phối của dữ liệu thật và tôi có thể dự đoán giá đóng cửa từ dữ liệu đã được tạo ra. Giả sử đầu vào của mạng là  $X = \{X_1, X_2, \dots, X_L\}$ , bao gồm dữ liệu giao dịch cổ phiếu hàng ngày của  $L$  ngày. Mỗi  $X_k$  trong  $X$  là một vector được tạo ra từ 5 đặc trưng như sau:

$$X_k = [X_{k,close}; X_{k,open}; X_{k,high}; X_{k,low}; X_{k,volume}]$$

Date	Open	High	Low	Close	Adj Close	Volume
1992-01-02	21.271511	21.659895	21.241634	21.570269	11.003518	7451286
1992-01-03	21.570269	21.779398	21.510515	21.600143	11.018756	6069310
1992-01-06	21.749521	22.257408	21.719646	22.048279	11.247362	9198524
1992-01-07	21.988527	22.645794	21.958652	22.615917	11.536928	8255032
1992-01-08	22.466539	22.705545	22.048279	22.078156	11.262602	10526526
...	...	...	...	...	...	...
2021-12-28	131.600006	133.220001	131.589996	132.630005	129.480743	3445200
2021-12-29	132.399994	134.210007	132.300003	133.350006	130.183640	4239900
2021-12-30	133.750000	134.369995	133.330002	133.910004	130.730362	3158100
2021-12-31	134.000000	134.990005	133.610001	133.660004	130.486298	3362100
2022-01-03	134.070007	136.289993	133.630005	136.039993	132.809769	4605900

**Hình 4.2:** Dữ liệu cổ phiếu IBM thu thập từ thư viện Yfinance

#### 4.3.2 Biến đổi dữ liệu

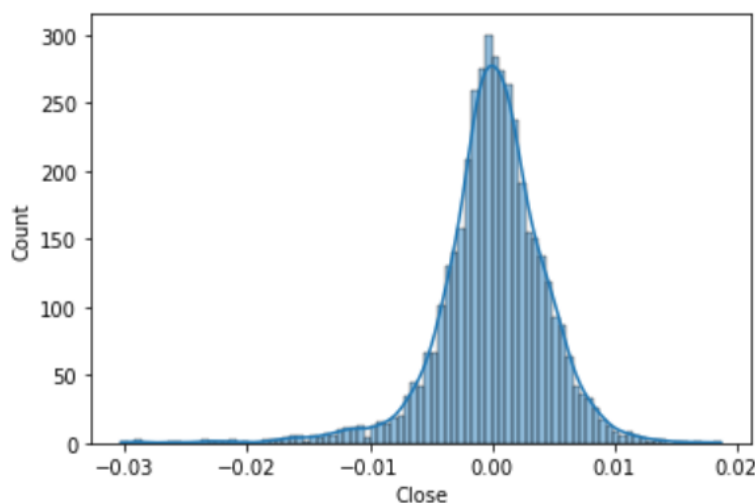
Mục tiêu của đề tài là phát hiện các xu hướng của cổ phiếu tuy nhiên dữ liệu theo ngày thường có nhiều nhiễu do biến động giá ngẫu nhiên. Để giảm nhiễu và làm biểu đồ mượt hơn, một kỹ thuật thường được sử dụng là đường trung bình động.

Trong nghiên cứu này tôi sử dụng đường trung bình động 10 ngày để làm mượt dữ liệu. Hình 4.3 hiển thị giá cổ phiếu và đường trung bình động, nó cho thấy đường trung bình động làm giảm nhiều từ biến động giá cổ phiếu ngẫu nhiên nhưng vẫn giữ được xu hướng tăng giảm của cổ phiếu.



**Hình 4.3:** Đường trung bình động 10 ngày

Sau đó, dữ liệu được biến đổi thành phần trăm thay đổi như đã trình bày ở 3.2.1. Hình 4.4 cho thấy sau khi biến đổi về phần trăm thay đổi, dữ liệu có phân bố chuẩn. Điều này giúp mạng nơ-ron có thể học tốt hơn và tránh được hiện tượng triệt tiêu và bùng nổ gradient.



**Hình 4.4:** Phân bố dữ liệu sau khi biến đổi thành phần trăm thay đổi

### 4.3.3 Chuẩn hóa dữ liệu

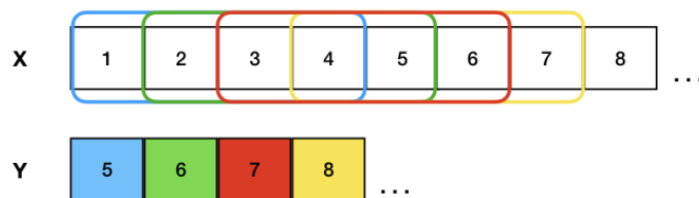
Chuẩn hóa dữ liệu là quá trình thay đổi phạm vi giá trị trong một tập dữ liệu. Khi tôi sử dụng dữ liệu giá và khối lượng, tất cả dữ liệu cổ phiếu phải nằm trong một miền giá trị quy chuẩn. Nói chung, các thuật toán học máy và học sâu hội tụ nhanh hơn hoặc hoạt động tốt hơn khi dữ liệu gần với phân phối chuẩn và trên một miền giá trị tương tự nhau. Ngoài ra, trong thuật toán học máy, hàm kích hoạt, chẳng hạn

như hàm sigmoid, có điểm bão hòa nhỏ mà các giá trị ngoài điểm đó cho kết quả đầu ra là không đổi gây ra hiện tượng vanishing và exploding gradient. Do đó, khi sử dụng các mô hình, dữ liệu phải được chuẩn hóa trước khi sử dụng. Quá trình này được thực hiện bằng các phương thức MinMaxScaler của thư viện scikitlearn. Khi MinMaxScaler được áp dụng cho một đặc trưng, nó sẽ trừ mỗi giá trị cho giá trị nhỏ nhất của đặc trưng đó và chia kết quả cho miền giá trị hiện tại. Bằng cách này, MinMaxScaler bảo toàn hình dạng của phân phối ban đầu. MinMaxScaler chuẩn hóa các giá trị đầu vào nằm trong khoảng  $[0,1]$ .

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

#### 4.3.4 Tạo dữ liệu đào tạo

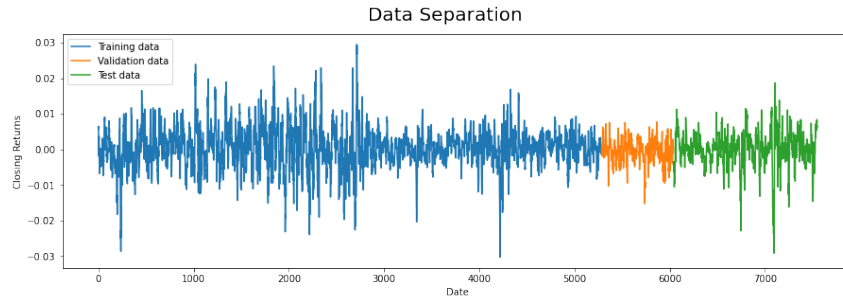
Ý tưởng chính của mô hình là sử dụng sự biến động giá của  $n$  ngày trước đó để dự đoán giá đóng cửa của ngày kế tiếp. Do đó, tôi sử dụng cửa sổ trượt  $n$  ngày ( $n$  là tham số) để dự đoán xu hướng cho ngày thứ  $n + 1$ . Hình 4.5 mô tả quá trình này với độ dài bộ lọc là 5 và bước nhảy là 1. Mỗi màu tương ứng với mỗi điểm dữ liệu đầu vào được đưa vào mạng. 5 phần tử đầu tiên được lấy từ tập dữ liệu, 4 phần tử đầu tiên sẽ là đầu vào ( $x$ ) và phần tử cuối cùng sẽ là nhãn ( $y$ ) của  $x$  đó. Tiếp theo, bộ lọc di chuyển về phía trước với kích thước bước nhảy là 1 và một điểm dữ liệu khác được tạo. Bộ lọc sẽ tiếp tục tiến về phía trước và quá trình này được lặp lại cho đến khi bộ lọc đến cuối tập dữ liệu. Trong quá trình huấn luyện mô hình, mạng sẽ học cách sử dụng dữ liệu đầu vào để dự đoán dữ liệu đầu ra, có nghĩa là, với một chuỗi 4 giá trị liên tiếp, mô hình sẽ xuất ra giá trị mà nó tin rằng giá trị đó là giá trị thứ 5. Cuối cùng, với cửa sổ kích thước bước nhảy là  $n$  và bộ dữ liệu kích thước ( $L, 5$ ) ( $L$  là số ngày lịch sử giao dịch) ta thu được một bộ dữ liệu đào tạo có kích thước ( $L-n, n, 5$ ).



**Hình 4.5:** Cửa sổ trượt kích thước cửa sổ 5 và bước nhảy 1

Trên bộ dữ liệu (IBM), tôi chia tập dữ liệu thành 3 phần với tỷ lệ 70% train 10% validation và 20% test như hình 4.6 Bộ dữ liệu validation được sử dụng để kiểm tra overfitting và underfitting của mô hình.





**Hình 4.6:** Dữ liệu huấn luyện, validation và test

#### 4.4 Phương pháp đánh giá

Tôi đánh giá hiệu quả dự báo của mô hình đề xuất bằng các tiêu chí thống kê như: Sai số bình phương trung bình (MSE), sai số trung bình bình phương gốc (RMSE) và sai số bình quân tuyệt đối (MAE) được tính theo các công thức trình bày sau đây:

MSE được sử dụng để đánh giá hiệu suất của mô hình dựa trên sai số trung bình của dữ liệu dự báo và dữ liệu thật. Ngoài ra MSE được sử dụng làm hàm mất mát cho mô hình. MSE được tính theo công thức sau:

$$MSE = \frac{1}{N} \sum_{k=1}^N (\hat{y}_k - y_k)^2$$

RMSE là một trong những thước đo sai số được sử dụng phổ biến nhất trong các bài toán hồi quy. Nó bằng căn bậc hai của MSE. Dựa trên sai số RMSE, có thể xác định mức độ phân bố của dữ liệu xung quanh giá trị tối ưu:

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (\hat{y}_k - y_k)^2}$$

MAE đo lường mức độ chính xác của dự báo.

$$MAE = \frac{1}{N} \sum_{k=1}^N |\hat{y}_k - y_k|$$

Trong đó:  $y_k$  là dữ liệu cổ phiếu thật.

$\hat{y}_k$  là dữ liệu cổ phiếu dự đoán.

$N$  là tổng số ngày giao dịch của cổ phiếu.

Các độ đo MAE, MSE và RMSE có giá trị thấp có nghĩa là dự đoán giá đóng gần đúng với dữ liệu thật.

Tham số	Giá trị
Epochs	200
Batch size	32
Optimizer	AdamW
Learning rate	0.001
Time steps	32

**Bảng 4.1:** Tham số mô hình

#### 4.5 Các tham số đánh giá

Các tham số được sử dụng trong quá trình đào tạo mô hình được tóm tắt trong Bảng 4.1. Thuật toán tối ưu hóa AdamW được sử dụng trong quá trình đào tạo với tốc độ học tập là 0,001. Chúng tôi đào tạo mô hình trong 200 epochs và batch size bằng 32. Tôi sử dụng cửa sổ trượt kích thước 32 ngày để tạo bộ dữ liệu.

#### 4.6 Phương pháp thí nghiệm

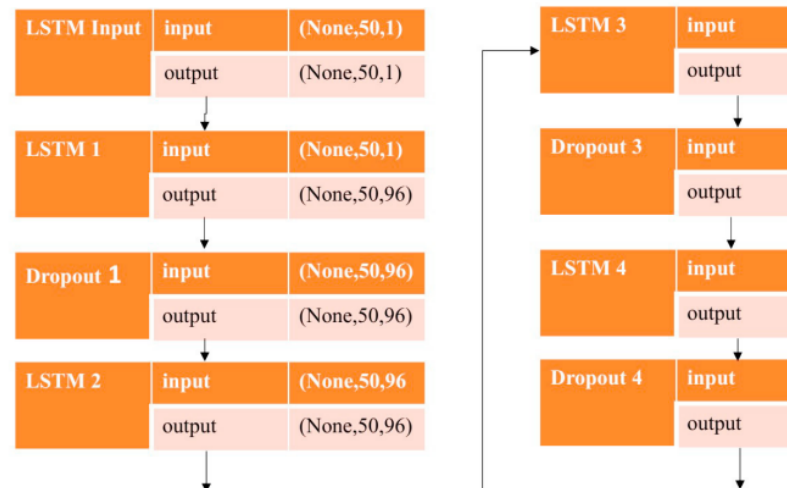
Với mỗi thí nghiệm, tôi tiến hành thực hiện 5 lần với các tham số đã trình bày trong bảng 4.1 và lấy kết quả tốt nhất để so sánh.

Để đánh giá phương pháp được đề xuất tôi so sánh kết quả của mô hình đề xuất với một số mô hình dự đoán đã có như LSTM, CNN kết hợp LSTM. LSTM, LSTM kết hợp CNN là các phương pháp hồi quy cho dự đoán thị trường cổ phiếu và tôi chọn chúng như là những phương pháp cơ bản để so sánh với mô hình đề xuất của tôi. Các phương pháp này là các phương pháp hiện đại và đã đạt được kết quả tốt trong những nghiên cứu trước đây. Kết quả thu được cho thấy mô hình đề xuất của tôi đạt kết quả tốt hơn và trong khoảng thời gian ngắn hơn. Chi tiết kết quả các thí nghiệm được trình bày ở phần sau.

#### 4.7 Kết quả thí nghiệm

##### 4.7.1 LSTM

Tôi sử dụng mô hình LSTM được trình bày trong bài báo của Moghar và cộng sự [26]. Trong đó, các tác giả sử dụng 4 lớp LSTM liên tiếp nhau để dự đoán giá cổ phiếu. Kiến trúc mô hình LSTM được mô tả trong hình 4.7.

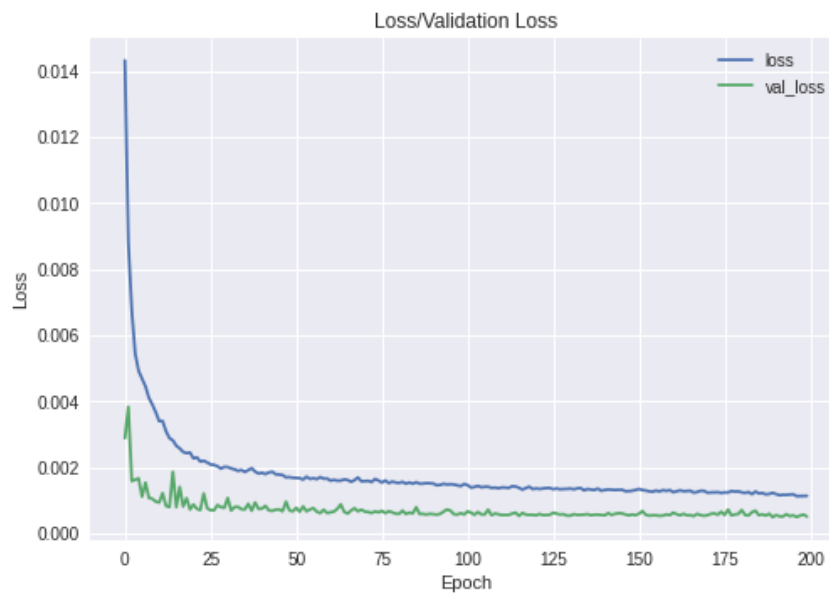


**Hình 4.7:** Kiến trúc mô hình LSTM [26]

Kết quả thu được như sau:

Mô hình	MSE	RMSE	MAE
LSTM	0.0013	0.0356	0.0224

**Bảng 4.2:** Kết quả độ đo của mô hình LSTM



**Hình 4.8:** Hàm mất mát của mô hình LSTM



**Hình 4.9:** Biểu đồ kết quả dự đoán của mô hình LSTM

#### 4.7.2 Mô hình LSTM kết hợp CNN

Tôi sử dụng mô hình LSTM CNN được giới thiệu trong bài báo của Wenjie Lu [27]. Trong đó, các tác giả sử dụng 2 lớp CNN để trích xuất các đặc trưng trước khi đưa vào các lớp LSTM. Kiến trúc mô hình được mô tả trong hình 4.10.

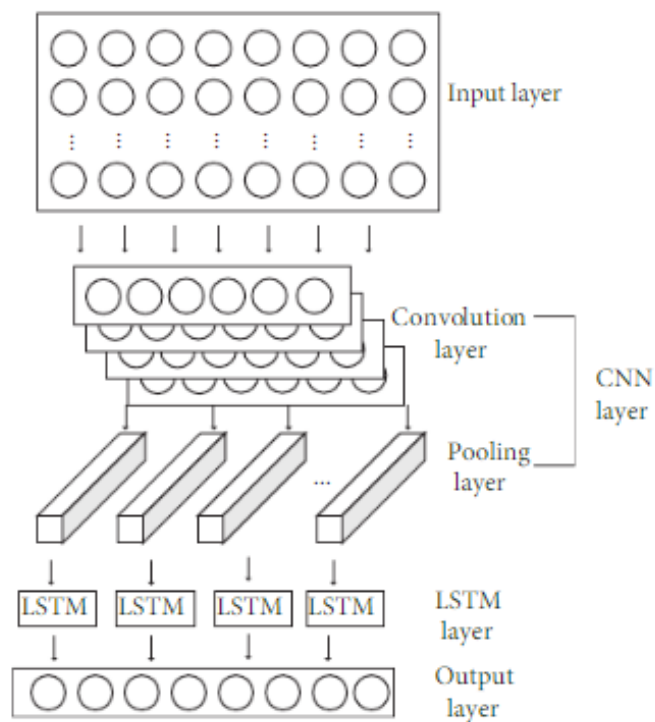


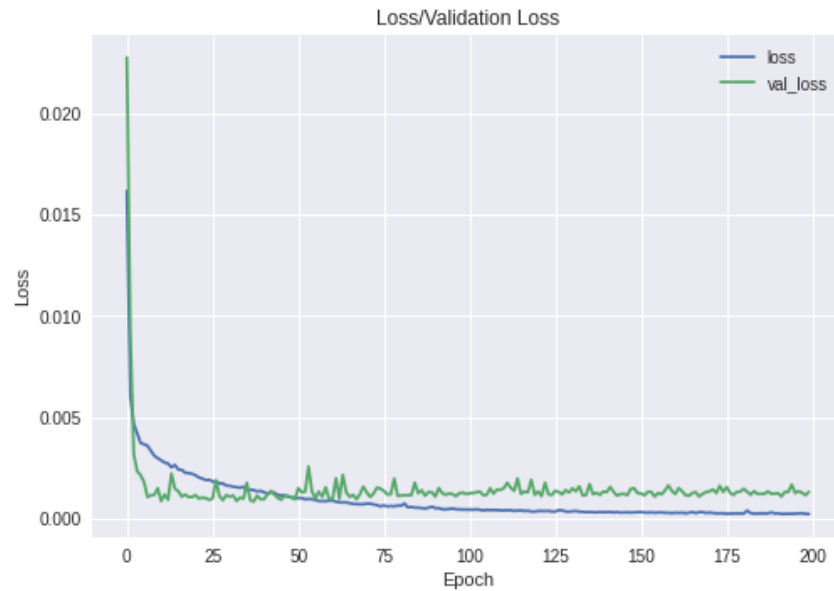
FIGURE 2: Architecture of LSTM memory cell.

**Hình 4.10:** Kiến trúc mô hình CNN kết hợp LSTM [27]

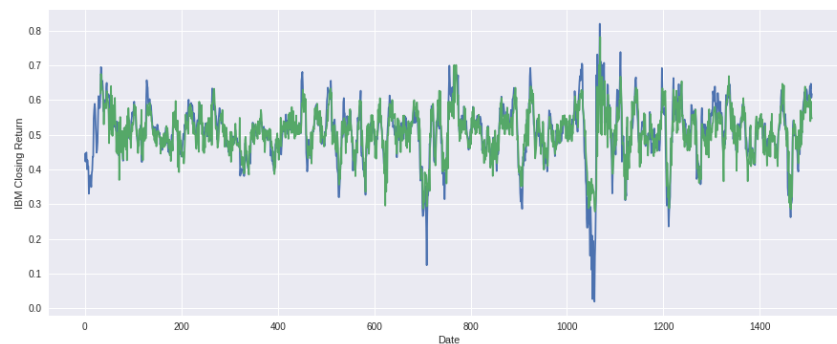
Kết quả thu được như sau:

Mô hình	MSE	RMSE	MAE
LSTM+CNN	0.0023	0.0335	0.0315

**Bảng 4.3:** Kết quả độ đo của các mô hình LSTM kết hợp CNN



**Hình 4.11:** Hàm mất mát của mô hình LSTM kết hợp CNN



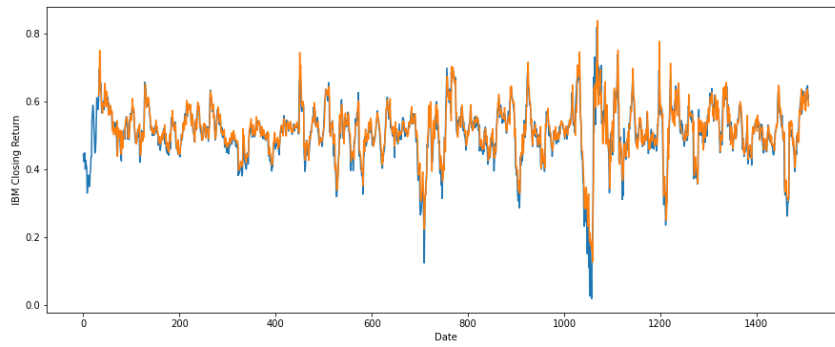
**Hình 4.12:** Biểu đồ kết quả dự đoán của mô hình LSTM kết hợp CNN

### 4.7.3 Mô hình Transformer cơ bản

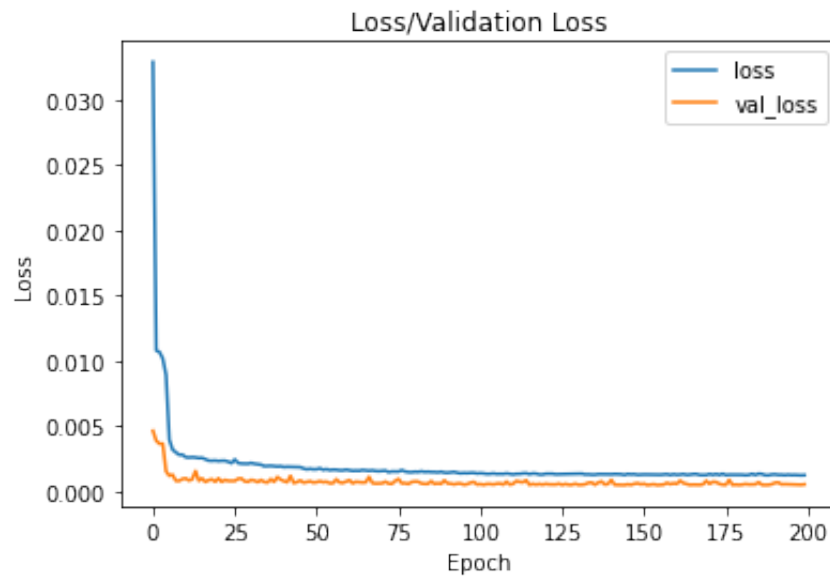
Mô hình Transformer cơ bản chỉ sử dụng Time2Vec để mã hóa đặc trưng về thời gian mà không trực quan hóa tham số.

Mô hình	MSE	RMSE	MAE
Transformer cơ bản	0.0011	0.0326	0.0213

**Bảng 4.4:** Kết quả độ đo của mô hình Transformer cơ bản



**Hình 4.13:** Biểu đồ kết quả dự đoán của mô hình Transformer cơ bản



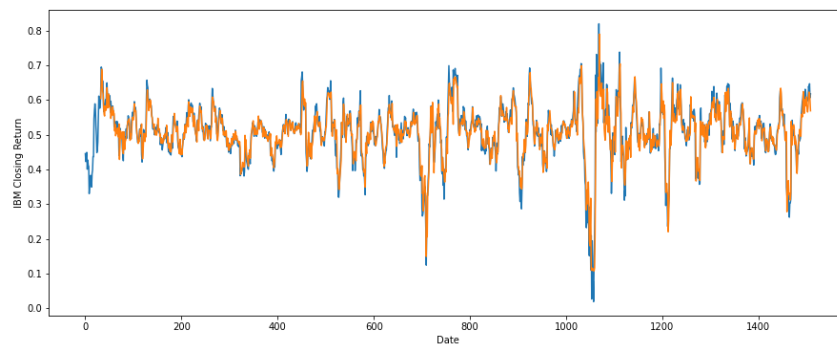
**Hình 4.14:** Hàm mất mát của mô hình Transformer cơ bản

#### 4.7.4 Mô hình Transformer đề xuất

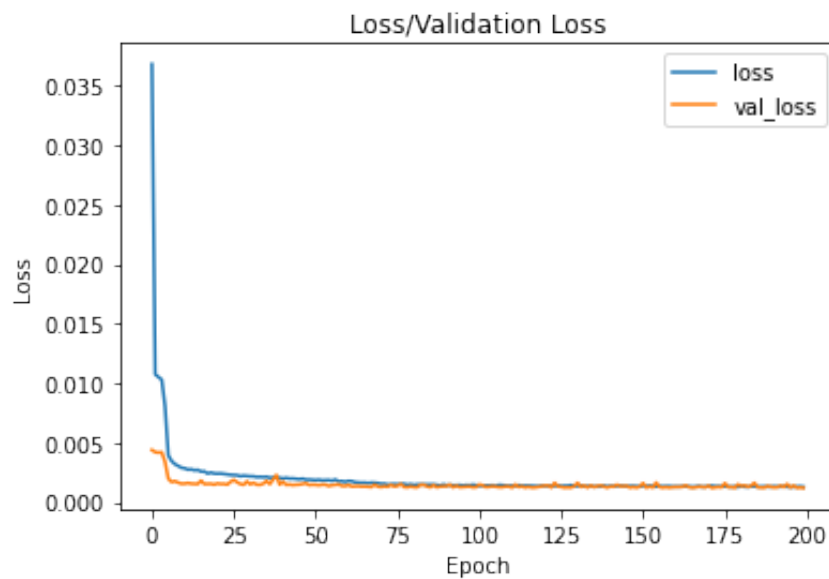
Mô hình Transformer đề xuất sử dụng Time2Vec để mã hóa đặc trưng về thời gian và trực quan hóa tham số.

Mô hình	MSE	RMSE	MAE
T2V-Transformer	0.001	0.0318	0.0211

**Bảng 4.5:** Kết quả độ đo của mô hình Transformer đề xuất



**Hình 4.15:** Biểu đồ kết quả dự đoán của mô hình Transformer đề xuất



**Hình 4.16:** Hàm mất mát của mô hình Transformer đề xuất

#### 4.8 Đánh giá

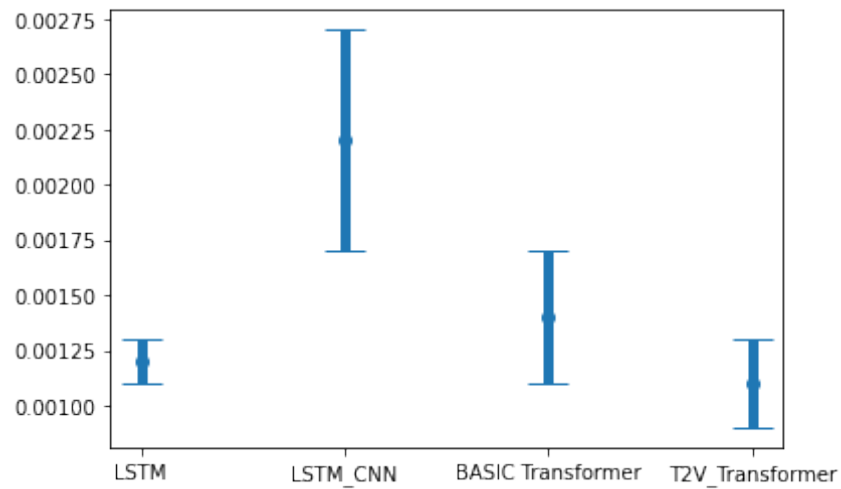
So sánh các mô hình trên với mô hình Transformer đề xuất, tôi thu được bảng kết quả như sau:

Mô hình	MSE	RMSE	MAE
LSTM	0.0013	0.0356	0.0224
LSTM+CNN	0.0023	0.0336	0.0315
Transformer cơ bản	0.0011	0.0326	0.0213
<b>T2V-Transformer</b>	<b>0.001</b>	<b>0.0318</b>	<b>0.0211</b>

**Bảng 4.6:** Kết quả độ đo của các mô hình

Kết quả dự đoán được chỉ ra trong Bảng 4.6 với phần tô đậm là kết quả tốt nhất. Các độ đo MSE, RMSE và MAPE có giá trị thấp có nghĩa là dự đoán giá đóng gần đúng với dữ liệu thật. Từ các biểu đồ, các độ đo của các mô hình, tôi nhận thấy mô hình Transformer của tôi tốt hơn trên cả 3 độ đo được chọn. Qua đó khẳng định

mô hình của tôi đã đạt kết quả tốt trong quá trình huấn luyện mô hình so với những phương pháp khác.



**Hình 4.17:** Sai số độ đo MSE của các mô hình

Hình 4.17 thể hiện sai số của các mô hình sau 5 lần thử nghiệm. Kết quả cho thấy mô hình LSTM có độ ổn định cao nhất, mô hình LSTM kết hợp CNN có độ ổn định thấp. Ngoài ra trong quá trình huấn luyện tôi nhận thấy mô hình LSTM kết hợp CNN dễ bị overfitting và cho kết quả kém trên tập kiểm thử. Mô hình Transformer đề xuất cho kết quả tốt nhất nhưng độ ổn định không cao bằng LSTM.



## CHƯƠNG 5. KẾT LUẬN

### 5.1 Kết luận

Trong bài báo này, chúng tôi đề xuất áp dụng mô hình Transformer để dự đoán giá đóng cửa của cổ phiếu trong đó cơ chế self-attention có thể giúp nắm bắt sự phụ thuộc dài hạn của chuỗi thời gian tài chính. Hơn nữa, khi áp dụng các cải tiến được đề xuất là sử dụng Time2Vec để mô hình hóa đặc trưng thời gian, trực quan hóa tham số, mô hình dựa trên Transformer của chúng tôi đạt được hiệu suất đáng kể so với một số mô hình hiện đại trên tập dữ liệu thế giới thực.

## TÀI LIỆU THAM KHẢO

- [1] C. Cortes **and** V. Vapnik, “Support-vector networks,” *Machine Learning*, **jourvol** 20, **number** 3, **pages** 273–297, 1995.
- [2] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, **jourvol** 46, **number** 3, **pages** 175–185, 1992.
- [3] L. Breiman, “Random forests,” *Machine Learning*, **jourvol** 45, **number** 1, **pages** 5–32, 2001.
- [4] R. P. Schumaker **and** H. Chen, “Textual analysis of stock market prediction using breaking financial news: The AZFin text system,” *ACM Transactions on Information Systems (TOIS)*, 2009.
- [5] Y. Xu **and** S. B. Cohen, “Stock movement prediction from tweets and historical prices,” **in** *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* 2018, 1970–1979.
- [6] D. Chen, Y. Zou, K. Harimoto, R. Bao, X. Ren **and** X. Sun, *Incorporating fine-grained events in stock movement prediction*, 2019. **url**: <https://arxiv.org/abs/1910.05078>.
- [7] D. E. Rumelhart, G. E. Hinton **and** R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, **jourvol** 323, **number** 6088, **pages** 533–536, 1986.
- [8] S. Hochreiter **and** J. Schmidhuber, “Long short-term memory,” *Neural computation*, 1735–1780, 1997.
- [9] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong **and** W. chun Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” **in** *Advances in neural information processing systems* 2015, **pages** 802–810.
- [10] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang **and** G. Cottrell, *A dual-stage attention-based recurrent neural network for time series prediction*, 2017. **url**: <https://arxiv.org/abs/1704.02971>.
- [11] S. Li, X. Jin, Y. Xuan **and others**, *Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting*, 2019. **url**: <https://arxiv.org/abs/1907.00235>.
- [12] A. Vaswani, N. Shazeer, N. Parmar **and others**, “Attention is all you need,” **in** *Advances in neural information processing systems* 2017.
- [13] J. Liu, H. Lin, X. Liu **and others**, “Transformer-based capsule network for stock movement prediction,” 2019.

- [14] Q. Ding, S. Wu, H. Sun, J. Guo **and** J. Guo, “Hierarchical multi-scale gaussian transformer for stock movement prediction,” *in Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20 C. Bessiere, editor*, International Joint Conferences on Artificial Intelligence Organization, **july** 2020, **pages** 4640–4646.
- [15] C. Tao, S. Gao, M. Shang, W. Wu, D. Zhao **and** R. Yan, “Get the point of my utterance! learning towards effective responses with multi-head attention mechanism,” *in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* International Joint Conferences on Artificial Intelligence Organization, 2018, **pages** 4418–4424.
- [16] J. Li, Z. Tu, B. Yang, M. R. Lyu **and** T. Zhang, *Multi-head attention with disagreement regularization*, 2018. **url**: <https://arxiv.org/abs/1810.10183>.
- [17] M. Lee, J. Lee, H. J. Jang, B. Kim, W. Chang **and** K. Hwang, *Orthogonality constrained multi-head attention for keyword spotting*, 2019. **url**: <https://arxiv.org/abs/1910.04500>.
- [18] S. M. Kazemi, R. Goel, S. Eghbali **and others**, *Time2vec: Learning a vector representation of time*, 2019. **url**: <https://arxiv.org/abs/1907.05321>.
- [19] J. Bollen, H. Mao **and** X. Zeng, “Twitter mood predicts the stock market,” *Journal of Computational Science*, **jourvol** 2, **number** 1, **pages** 1–8, 2011, ISSN: 1877-7503.
- [20] H. Sul, A. R. Dennis **and** L. I. Yuan, “Trading on twitter: The financial information content of emotion in social media,” *in 2014 47th Hawaii International Conference on System Sciences* 2014, **pages** 806–815.
- [21] S. Pagolu, K. Challa, G. Panda **and** B. Majhi, “Sentiment analysis of twitter data for predicting stock market movements,” 2016.
- [22] W. Bao, J. Yue **and** Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory,” *PLoS ONE*, **jourvol** 12, 2017.
- [23] H. M, G. E.A., V. K. Menon **and** S. K.P., “Nse stock market prediction using deep-learning models,” *Procedia Computer Science*, **jourvol** 132, **pages** 1351–1362, 2018.
- [24] J. Moody **and** M. Saffell, “Learning to trade via direct reinforcement,” *IEEE Transactions on Neural Networks*, **jourvol** 12, **number** 4, **pages** 875–889, 2001.

- [25] M. Dempster **and** V. Leemans, “An automated fx trading system using adaptive reinforcement learning,” *Expert Systems with Applications*, **journal** 30, **number** 3, **pages** 543–552, 2006, Intelligent Information Systems for Financial Engineering, ISSN: 0957-4174.
- [26] A. Moghar **and** M. Hamiche, “Stock market prediction using lstm recurrent neural network,” *Procedia Computer Science*, **journal** 170, **pages** 1168–1173, 2020.
- [27] W. Lu, J. Li, Y. Li, A. Sun **and** J. Wang, “A cnn-lstm-based model to forecast stock prices,” *Complexity*, **journal** 2020, 2020.