**TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**BÁO CÁO BÀI TẬP LỚN**

**ĐỀ TÀI: XÂY DỰNG MÔ HÌNH LONG SHORT-TERM MEMORY DỰ ĐOÁN GIÁ CHỨNG KHOÁN**

|  |  |
| --- | --- |
| **Nhóm 03 – 64CS3** | |
| **Thành viên nhóm** | Dương Gia Khánh - 1526864  Vương Trung Thành - 186864  Ngô Đức Thịnh - 189464  Nguyễn Trần Lê Tuấn - 1553564 |
| **Giảng viên hướng dẫn** | Ths. Nguyễn Đình Quý |

**HÀ NỘI, 02/2022**

# LỜI NÓI ĐẦU

Ngày nay, cùng với sự phát triển vượt bậc của con người về mọi mặt thì trí tuệ nhân tạo ngày càng được ứng dụng nhiều trong tất cả mọi lĩnh vực đời sống.Nó không chỉ làm cuộc sống trở lên tốt mà còn xử lý được những công việc khó khan tốn nhiều thơi gian.Học máy là một lĩnh vực thuộc trí tuệ nhân tạo. Gần đây, học máy đang dẫn đầu xuthế và tạo nên những thay đổi vượt bậc trong trí tuệ nhân tạo nói chung và công nghệthông tin nói riêng.

Trong môn học “Học máy” kỳ này, nhóm 03-64CS3 thực hiệnmột project liên quan đến học máy giúp hiểu hơn một bài toán học máy cũngnhư tạo ra một giải pháp học máy có ứng dụng rộng rãi trong thực tế.Nhóm chúng em đã chọn chủ đề dự đoán giá đóng cửa chứng khoán. Đây là 1 chủ đề khá hay nhất là đang trong thời buổi dịch bênh vẫn đang ảnh hưởng nghiêm trọng đến nền kinh tế nhất là trongg 1 số thị trường như chứng khoán, tiền ảo, bất động sản, …. Do thời gian và khả năng có hạn nên không thể tránh khỏi sai sót. Vì vậy nhóm chúng em thầy có thể nhận xét và đưa ra ý kiến để chúng em hoàn thiện hơn bài báo cáo của nhóm mình. Cuối cùng, nhóm 03 xin chân thành cảm ơn thầy Nguyễn Đình Quý đã hướng dẫnchúng em trong suốt thời gian làm báo cáo này.

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc95311267)

[CHƯƠNG 1. KHẢO SÁT VẤN ĐỀ 4](#_Toc95311268)

[1.1 Đặt vấn đề 4](#_Toc95311269)

[1.2 Mô tả bài toán 4](#_Toc95311270)

[1.3 Định hướng công việc 4](#_Toc95311271)

[CHƯƠNG 2. THU THẬP DỮ LIỆU (CRAWL DATA) 5](#_Toc95311272)

[2.1 Phân tích, lựa chọn phương pháp crawl 5](#_Toc95311273)

[2.2 Lựa chọn công cụ crawl 6](#_Toc95311274)

[2.3 Tiến hành crawl dữ liệu 6](#_Toc95311275)

[2.4 Một số vấn đề gặp phải trong quá trình crawl và cách giải quyết 7](#_Toc95311276)

[2.5 Kết quả 8](#_Toc95311277)

[CHƯƠNG 3. PHÂN TÍCH DỮ LIỆU 10](#_Toc95311278)

[3.1 Tiền xử lý dữ liệu 10](#_Toc95311279)

[3.1.1 Tách dữ liệu trường “thay đổi” 10](#_Toc95311280)

[3.1.2 Thay đổi định dạng và gộp một số trường 10](#_Toc95311281)

[3.1.3 Xoá những trường không cần thiết 11](#_Toc95311282)

[3.1.4 Đảo lại dữ liệu theo thời gian 11](#_Toc95311283)

[3.1.5 Chuẩn hoá dữ liệu 11](#_Toc95311284)

[3.2 Phân tích các dữ liệu thu thập được 11](#_Toc95311285)

[3.2.1 Biểu đồ giá đóng cửa và giá tham chiếu 11](#_Toc95311286)

[3.2.2 Biểu đồ giá đóng cửa 12](#_Toc95311287)

[3.2.3 Biểu đồ khối lượng giao dịch trên sàn chứng khoán 13](#_Toc95311288)

[CHƯƠNG 4. XÂY DỰNG MÔ HÌNH HỌC MÁY 15](#_Toc95311289)

[4.1 Phương pháp đánh giá các mô hình 15](#_Toc95311290)

[4.1.1 Kịch bản đánh giá các mô hình 15](#_Toc95311291)

[4.1.2 Chọn độ đo 15](#_Toc95311292)

[4.2 Mô hình LSTM 16](#_Toc95311293)

[4.2.1 Lý do chọn LSTM 16](#_Toc95311294)

[4.2.2 Cơ sở lý thuyết 16](#_Toc95311295)

[4.2.3 Xây dựng mô hình 24](#_Toc95311296)

[4.2.4 Đánh giá mô hình 27](#_Toc95311297)

[CHƯƠNG 5. KẾT LUẬN 28](#_Toc95311298)

[TÀI LIỆU THAM KHẢO 29](#_Toc95311299)

# KHẢO SÁT VẤN ĐỀ

## Đặt vấn đề

Thị trường chứng khoán được biết đến là dễ biến động, luôn thay đổi khó lường và phi tuyến tính. Dự đoán giá cổ phiếu chính xác là vô cùng khó khăn vì phụ thuộc vào nhiều yếu tố vĩ mô và vi mô, chẳng hạn như yếu tố chính trị, bối cảnh kinh tế trong nước và kinh tế toàn cầu, các cú sốc bất ngờ (Covid-19, thiên tai, mất mùa,...), tình hình hoạt động tài chính của công ty,…

Tuy nhiên, tất cả những điều này cũng có nghĩa là có rất nhiều dữ liệu để tìm ra các pattern của chuỗi chứng khoán. Vì vậy, các chuyên gia phân tích tài chính, các nhà nghiên cứu và các nhà khoa học dữ liệu đang tiếp tục khám phá các kỹ thuật phân tích để phát hiện xu hướng và quy luật vận động của thị trường chứng khoán. Điều này đã làm nảy sinh khái niệm giao dịch theo thuật toán, sử dụng các chiến lược giao dịch tự động, được lập trình sẵn để thực hiện các lệnh.

Để phân tích được dữ liệu chứng khoán chúng ta có thể sử dụng cả phương pháp luận tài chính định lượng truyền thống và thuật toán máy học để dự đoán giá cổ phiếu. Một số phương pháp cụ thể trong việc phân tích dữ liệu chứng khoán chúng ta có thể lựa chọn chẳng hạn:

* Phân tích cổ phiếu theo phương pháp phân tích cơ bản và phương pháp phân tích kỹ thuật
* Dự đoán giá cổ phiếu bằng kỹ thuật Moving Average.
* Áp dụng các thuật toán RNN như LSTM, GRU.
* Sử dụng các lớp mô hình dự báo timeseries truyền thống như AR, MA, ARIMA, SARIMA, ARIMAX, GARCH…

Trong bài tập lớn môn học này chúng em sử dụng thuật toán LSTM để phân tích dữ liệu chứng khoán tại Việt Nam.

## Mô tả bài toán

Bài toán: Dự đoán giá chứng khoán thông qua dữ liệu từ quá khứ.

* Đầu vào: Các trường dữ liệu ảnh hưởng tới giá của một mã chứng khoán.
* Đầu ra: Giá đóng cửa của một mã chứng khoán trong một ngày cụ thể.
* Tập dữ liệu: tập dữ liệu được sử dụng là dữ liệu lịch sử giao dịch của các mã chững khoán được thu thập bằng cách crawl dữ liệu trên trang cafef.vn

## Định hướng công việc

Bước 1: Thu thập dữ liệu.

Bước 2: Phân tích dữ liệu.

Bước 3: Xây dựng mô hình học máy.

# THU THẬP DỮ LIỆU (CRAWL DATA)

Dữ liệu của các mã chứng khoán được thể hiện dưới dạng bảng như hình dưới đây(link: https://s.cafef.vn/Lich-su-giao-dich-VNA-1.chn) và để có thể lấy được chúng chúng ta có thể dùng API để lấy dữ liệu dưới dạng đơn giản hoặc trích xuất HTML để thu được các trường dữ liệu của bảng.

Graphical user interface, application, table, Excel

Description automatically generated

Hình . Dữ liệu của mã VNA được lấy từ cafef.vn

## Phân tích, lựa chọn phương pháp crawl

Để crawl dữ liệu trên chúng ta có thể sử dụng một trong 2 cách là dùng API lấy trực tiếp dữ liệu của 1 mã chứng khoán hoặc thông qua việc trích xuất HTML lấy dữ liệu dạng bảng của mã chứng khoán.Việc crawl thông qua API sẽ được ưu tiên hơn vì dữ liệu thu được tức thời và có thể dùng phân tích được luôn.

Do không tìm được API public của trang này và nhận thấy lượng dữ liệu lấy cũng không quá lớn nên chúng em quyết định crawl thông qua việc trích xuất HTML.

**Để crawl xong một mã chứng khoán ta cần lấy được tất cả các bảng dữ liệu của mã chứng khoán đó và để chuyển sang bảng tiếp theo chúng ta cần thực thi javascript của trang.**

## Lựa chọn công cụ crawl

Qua tìm hiểu nhóm thấy có các công cụ crawl website phổ biến như sau:

* Requests + BeautifulSoup: Sử dụng requests để lấy dữ liệu html sau đó dùng BeautifulSoup để trích xuất dữ liệu HTML ,cách này rất đơn giản và hiệu quả nhưng vấn đề gặp phải là chúng ta không thể khó có thể dùng request để dữ liệu các bảng tiếp theo của mã chứng khoán đó.
* Scrapy: Hoạt động tương tự như BeautifulSoup được tối ưu để có thể crawl được dữ liệu nhanh chóng và ít tiêu tốn tài nguyên hơn nhưng cũng gặp phải vấn đề về việc lấy dữ liệu của các trang tiếp theo.Ở đây ta có thể kết hợp thêm 1 công cụ khác là Splash để thực thi LUA script chuyển qua các bảng dữ liệu tiếp theo.
* Selenium: Khác với 2 công cụ trên selenium được tạo ra với mục đích không phải để crawl website mà để kiểm thử tự động website. Tuy nhiên, với các chức năng có sẵn của selenium chứng ta cũng có thể đồng thời trích xuất dữ liệu HTML cũng như chuyển sang các bảng dữ liệu khác như một người dùng. Do hoạt động bằng cách giả lập người dùng thật nên có thể crawl được các trang web có sử dụng các cơ chế ngăn chặn crawl.Tuy nhiên, selenium có nhược điểm là tốc độ crawl không cao cũng như tiêu tốn rất nhiều tài nguyên của máy tính.

**Do lượng dữ liệu crawl cũng không lớn cũng như chưa có kinh nghiệm gì trong việc crawl dữ liệu nên nhóm quyết định sẽ crawl dữ liệu bằng selenium với đặc điểm là tiêu thụ nhiều tài nguyên tốc độ crawl thấp nhưng do là công cụ kiểm thử nên có thể hiển thị trực quan khi dữ liệu đang được crawl.**

## Tiến hành crawl dữ liệu

Sau khi xác định được trang web để crawl, selenium có thể tìm các thành phần của trang web đó và có thể tương tác hoặc lấy dữ liệu của thành phần đó. Để lấy dữ liệu lịch sử của một mã chứng khoán ta sẽ bắt đầu với một đường dẫn có cấu trúc như sau:



Dữ liệu được crawl được lấy thông qua XPATH: các hàng dữ liệu đều có id có cấu.

trúc phần đầu giống nhau nên ta dễ dàng lấy được toàn bộ chúng sau đó lấy dữ liệu của từng thẻ <td> bên trong.

Graphical user interface, text

Description automatically generated

Sau khi lấy dữ liệu xong một trang ta chuyển sang các trang kế tiếp thông qua element chuyển trang:



## Một số vấn đề gặp phải trong quá trình crawl và cách giải quyết

* Trang web không load được hoặc load rất chậm tại một số giai đoạn vào những ngày chứng khoán giao dịch nên việc triển khai crawl chỉ được thực hiện vào những ngày cuối tuần.
* Các thành phần tương tác chuyển trang không hoạt động. Có thể là cơ chế chống crawl của trang này vì trong quá trình thử tương tác với trang như người dùng bình thường thì càng chuyển về các trang đằng sau thì khi click sang trang khác sẽ phải thực hiện rất nhiều lần mới hoàn thành được.Để giải quyết vấn đề này ta thêm cơ chế kiểm tra trang hiện tại và trang kế tiếp nếu tương tác chuyển sang không thành công hay không hợp lệ thì thực hiện lại sao cho đúng.Tuy nhiên nếu vẫn xảy ra ngoại lệ thì sẽ dừng lại để xử lý bằng tay thông qua giao diện.
* Một số dữ liệu chứng khoán có cấu trúc bảng dữ liệu khác nên phải sửa lại code phần này.
* Thỉnh thoảng trang có nhảy ra quảng cáo ngay giữa màn hình gây gián đoạn crawl. Hiện tại thì nhóm xử lý bằng tay thông qua việc giám sát lúc crawl.

## Kết quả

Mặc dù trong quá trình crawl vẫn phải xử lý bằng tay và cần giám sát nhưng việc thu thập dữ liệu cũng đã được hoàn thành.Dữ liệu được lưu vào file csv ,đầy đủ các trường dữ liệu như trên web và sau khi kiểm tra thì không thấy có dữ liệu NULL nên có thể triển khai các bước tiếp theo luôn.

**Graphical user interface, application, table, Excel

Description automatically generated**

Hình . Kết quả sau khi crawl dữ liệu

Ảnh có chứa bàn

Mô tả được tạo tự động

Hình . Chú thích của các cột trong bảng

Link tập dữ liệu nhóm đã crawl được:

<https://drive.google.com/file/d/1ExPJY32pmT2km6yAmVwhra-6zhfNYfFw/view?usp=sharing>

# PHÂN TÍCH DỮ LIỆU

## Tiền xử lý dữ liệu

Dữ liệu đầu vào:

Ảnh có chứa bàn

Mô tả được tạo tự động

### Tách dữ liệu trường “thay đổi”

Tách dữ liệu ở cột **Thay đổi** thành 2 cột mới do có 2 giá trị cùng 1 lúc là **thay đổi theo giá trị** và **thay đổi theo phần trăm** trong cột **Thay đổi**

Ảnh có chứa văn bản

Mô tả được tạo tự động

### Thay đổi định dạng và gộp một số trường

Chuyển các cột KL1, GT1, KL2, GT2 về kiểu float và gộp thành 2 cột là KL và GT với

* KL: KHỐI LƯỢNG GIAO DỊCH
* GT: GIÁ TRỊ GIAO DỊCH

Ảnh có chứa văn bản

Mô tả được tạo tự động

### Xoá những trường không cần thiết

### Đảo lại dữ liệu theo thời gian



### Chuẩn hoá dữ liệu

Ảnh có chứa văn bản

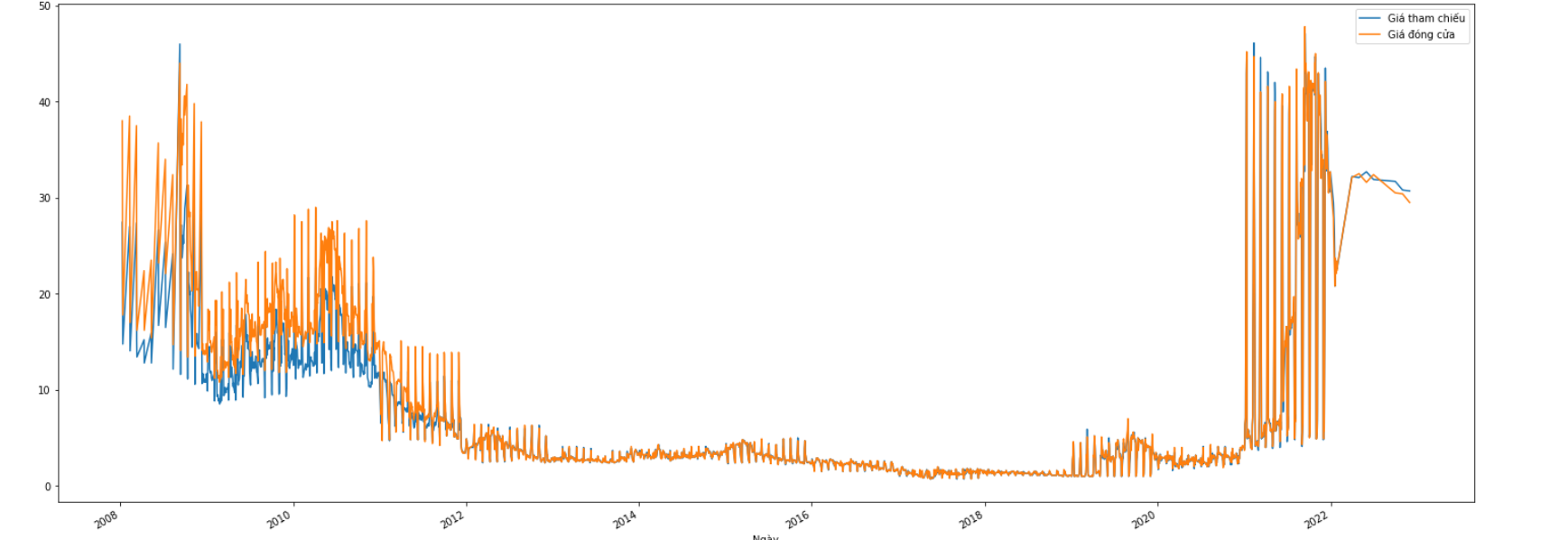
Mô tả được tạo tự động

## Phân tích các dữ liệu thu thập được

### Biểu đồ giá đóng cửa và giá tham chiếu

Giá đóng cửa là mức giá cửa cổ phiếu đó khi kết thúc phiên giao dịch của này hôm đó. Giá đóng cửa gồm giá mua và giá bán chứng khoán được xác định theo phương thức đấu giá. Vì cùng một thời điểm có nhiều người mua, nhiều người bán nên phải thực hiện việc đối chiếu giá mua hoặc giá bán. Trong một ngày giao dịch chứng khoán sẽ có mức giá cuối cùng được xác định cho việc mua, bán từng loại chứng khoán. Giá đóng cửa của ngày hôm nay sẽ là giá tham chiếu của ngày hôm sau làm cơ sở để tính giá trần và giá sàn của ngày đó.

Sự chênh lệnh giữa giá đóng cửa và giá tham chiếu thể hiện sự thay đổi của cổ phiếu trong ngày. Các bảng số liệu sau đây về giá đóng cửa và giá tham chiếu sẽ cho thấy cái nhìn khách quan về vấn đề này:



Hình . Biểu đồ giữa giá tham chiếu và giá đóng cửa của cổ phiếu VNA

Để biết cổ phiếu tăng hay giảm ta sẽ lấy:

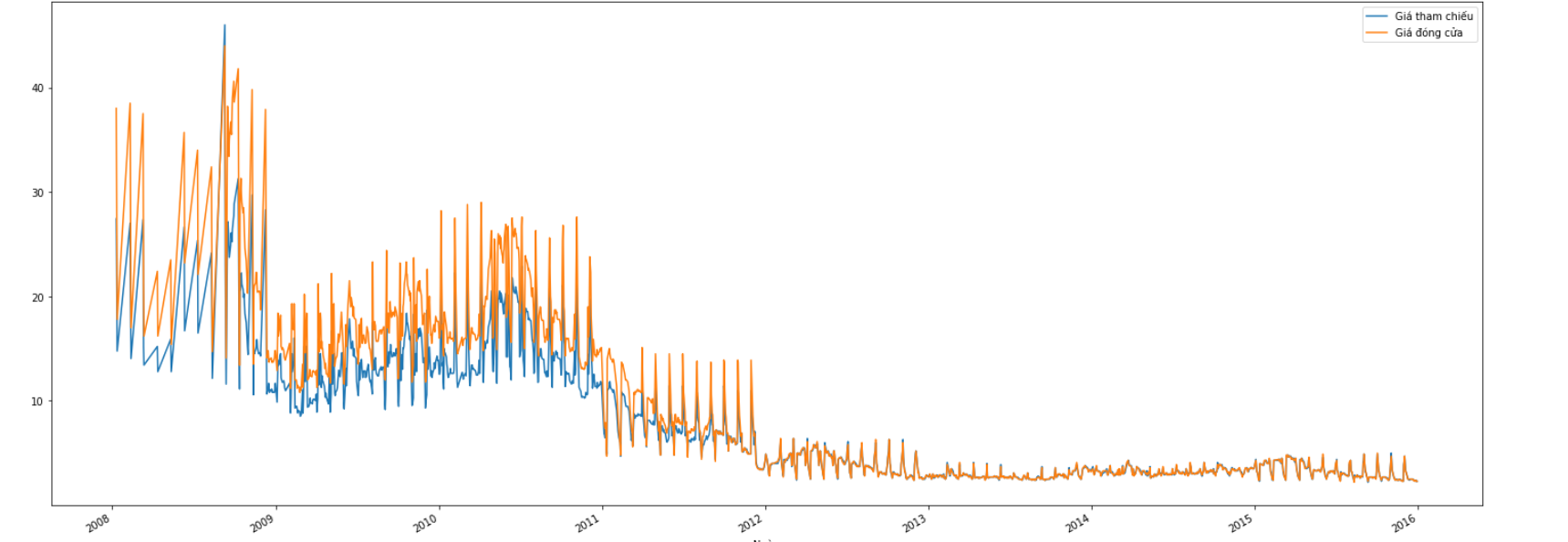
**Thay đổi = Giá đóng cửa – Giá tham chiếu**

Nếu thay đổi > 0 thì cổ phiếu tăng

Nếu thay đổi = 0 thì cổ phiếu không thay đổi

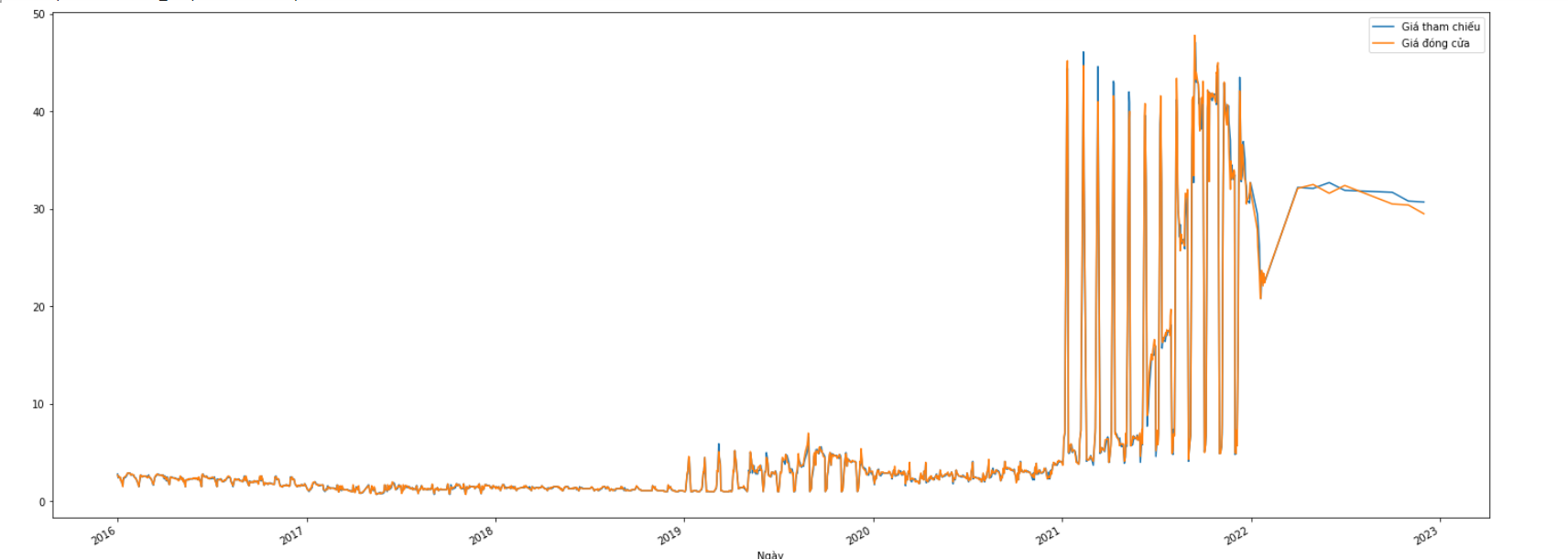
Nếu thay đổi < 0 thì cổ phiếu giảm

Để thấy rõ hơn sự thay đổi nhóm em sẽ chia biểu đồ làm 2 phần từ năm 2008 đến 2016 và 2022. Dưới đây là biểu đồ giá tham chiếu và đóng cửa trong khoảng thời gian đầu tiên từ 2008-2016



Hình . Giá tham chiếu và giá đóng của của VNA từ 2008 đến 2016

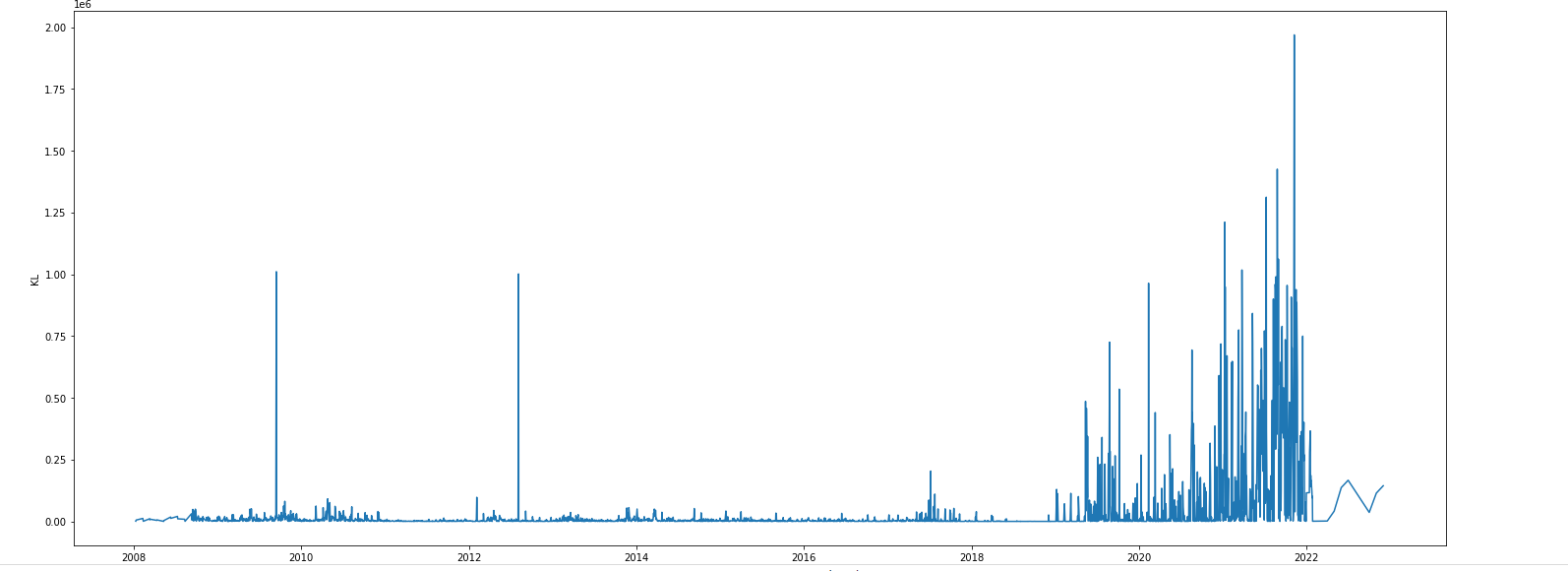
Duới đây là biểu đồ giá tham chiếu và đóng cửa trong khoảng thời gian còn lại từ 2016-2002



Hình . Giá tham chiếu và giá đóng của của VNA từ 2016 đến 2022

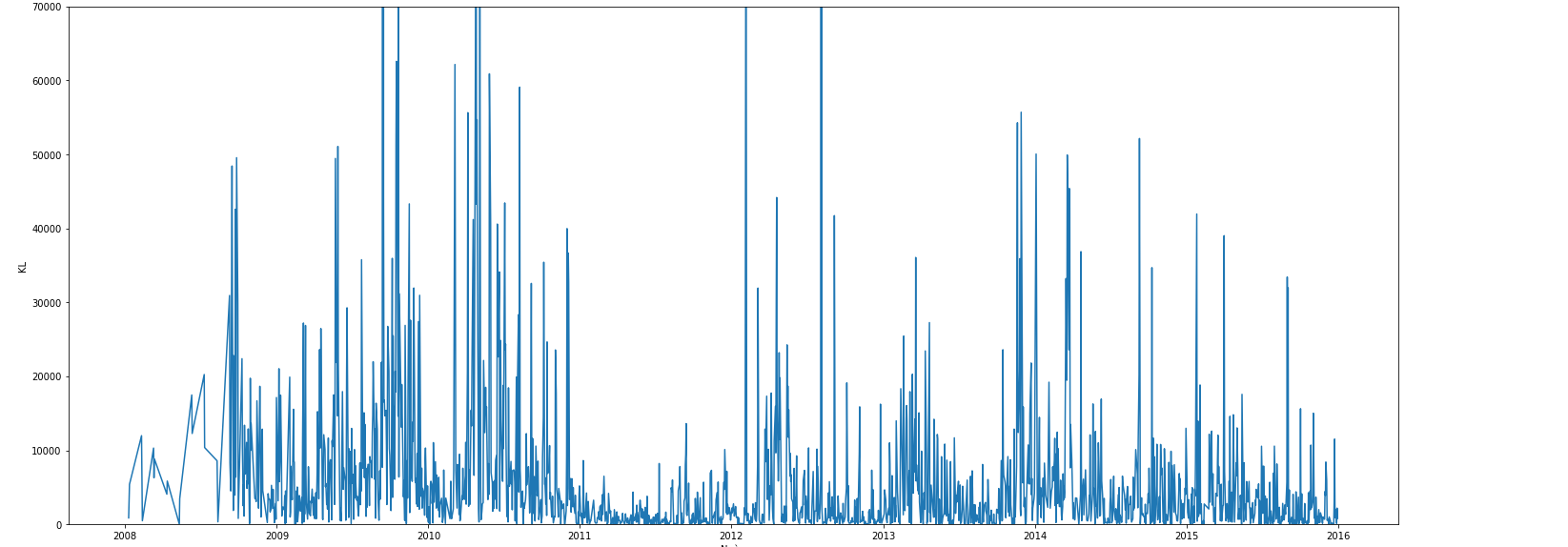
### Biểu đồ khối lượng giao dịch trên sàn chứng khoán

Khối lượng giao dịch của 1 cổ phiếu trên sàn chứng khoán là tổng số cổ phiếu thực sự được giao dịch (mua và bán) trong ngày giao dịch hoặc khoảng thời gian đã định, là một trong những yếu tố giúp cho nhiều nhà đầu tư xác định được xu hướng giá của cổ phiếu trái phiếu. Thông qua khối lượng giao dịch sau một phiên giao dịch ta có thể thấy được nhu cầu của người đầu tư hiện nay như thế nào, đánh giá tiềm năng của cổ phiếu đó. Chúng tôi thống kê khối lượng giao dịch qua từng năm như sau.

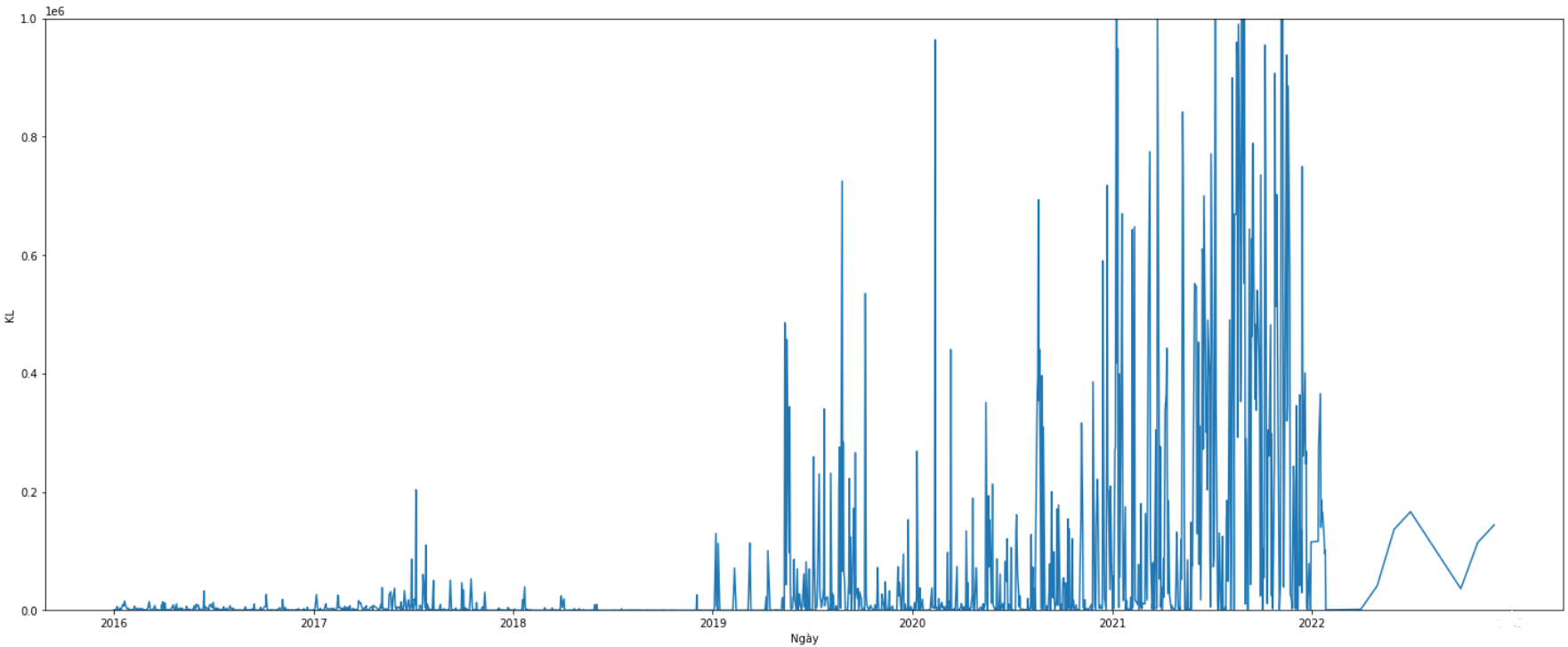


Hình . Biểu đồ khối lượng giao dịch của VNA từ 2008 đến 2022

Biểu đồ sau đây sẽ thống kê khối lượng giao dịch số cổ phiếu từ 2008-2015



Hình . Biểu đồ khối lượng giao dịch của VNA từ 2008 đến 2016



Hình . Biểu đồ khối lượng giao dịch của VNA từ 2016 đến 2022

# XÂY DỰNG MÔ HÌNH HỌC MÁY

## Phương pháp đánh giá các mô hình

### Kịch bản đánh giá các mô hình

Nhóm chúng em sẽ lấy mã chứng khoán VNA để lấy làm tập dữ liệu và áp dụng các mô hình học máy. Để có thế thực hiện các mô hình thì nhóm chúng em đã chia bộ dữ liệu ra thành 3 phần theo TimeSeriesSplit gồm: training set, validation set, test set với tỉ lệ lần lượt là: 7,5:1,5:1,chia nhỏ dữ liệu để thực hiện xác thực chéo. Phần lớn dữ liệu liệu là traing set là để huấn luyện mô hình, càng nhiều dữ liệu tốt được “feed” vào mô hình thì dự đoán càng chính xác. Test set là phần dữ liệu dùng để đánh giá việc học của mô hình trong đó test set là dữ liệu mới hoàn toàn đối với mô hình để có thể đóng vai trò là dữ liệu thực tế trong bài toán cần giải quyết. Validation set là 1 tập dữ liệu cùng phân phối với test set nhằm cải thiện kết quả đánh giá trên test set vì trong quá trình huấn luyện có thể xảy ra tình trạng overfitting nên tập validation set có tác dụng như là cảnh báo sớm các vấn đề xấu của mô hình.

### Chọn độ đo

Để có thể đánh giá được chất lượng mô hình nhóm chúng em sẽ đánh giá dựa theo MSE (Mean Squared Erorr) và RMSE (Root Mean Squared Erorr) để có thể kiểm tra độ tin cậy của mô hình. Để có thể hiểu dễ hơn về RMSE thì ta sẽ đi qua khái niệm MSE trước. MSE đánh giá chất lượng của một ước lượng hay một yếu tố dự báo nào đó. Công thức:

Trong đó: là giá trị ước lượng, dự báo

Y là giá trị quan sát được

n là số lần dự báo ước lượng

Sau khi đã hiểu qua về MSE thì RMSE được coi như là một biện pháp thường sử dụng trong những khác biệt của giá trị được dự đoán bởi mô hình hay là ước lượng các giá trị ta quan sát được. RMSE là thước đo mức độ lan truyền của những phần lỗi dự đoán. Nói cách khác, nó cho biết mức độ tập trung của dữ liệu xung quanh dòng phù hợp nhất. RMSE là thước đo mức độ hiệu quả của mô hình. Nó thực hiện điều này bằng cách đo sự khác biệt giữa các giá trị dự đoán và giá trị thực tế. RMSE càng nhỏ tức là sai số càng bé thì mức độ ước lượng cho thấy độ tin cậy của mô hình có thể đạt cao nhất. RMSE là thước đo độ chính xác, để so sánh các lỗi dự báo của các mô hình khác nhau cho một tập dữ liệu cụ thể chứ không phải giữa các bộ dữ liệu, vì nó phụ thuộc vào quy mô.

Công thức:

Trong đó: là giá trị ước lượng, dự báo

Y là giá trị quan sát được

n là số lần dự báo ước lượng

## Mô hình LSTM

### Lý do chọn LSTM

Neural Network có 2 mô hình lớn là Convolutional Neural Network (CNN) cho bài toán có input là ảnh và Recurrent neural network (RNN) cho bài toán dữ liệu dạng chuỗi (sequence). Về cơ bản nếu thấy sequence data hay time-series data mà muốn áp dụng Neural Network sẽ nghĩ ngay đến RNN. Tuy nhiên mạng RNN gặp phải một số hạn chế đó là phải thực hiện tuần tự, đạo hàm bị triệt tiêu (Vanishing gradient), bùng nổ đạo hàm (Exploding gradient). LSTM là một mạng cải tiến của RNN nhằm giải quyết phần nào các vấn đề mà mạng RNN gặp phải. Rất nhiều các bài toán học máy sử dụng LSTM đem lại kết quả rất đáng chú ý so với việc sử dụng các phương pháp khác. Dữ liệu chứng khoán chúng ta thu thập được là một dạng dữ liệu kiểu time-series vì vậy khá phù hợp khi sử dụng LSTM huấn luyện dữ liệu mà nhóm đã thu thập được.

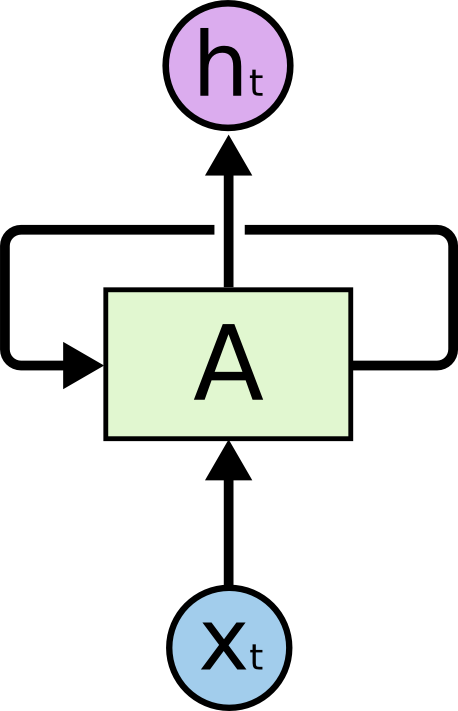
### Cơ sở lý thuyết

#### Mạng hồi quy RNN

Con người không bắt đầu suy nghĩ của họ từ đầu tại tất cả các thời điểm. Cũng như báo cáo này, để hiểu mỗi chữ ở đây dựa vào việc ta đã hiểu các chữ trước đó chứ không phải là đọc tới đâu ném hết đi tới đó, rồi lại bắt đầu suy nghĩ lại từ đầu tới chữ đang đọc. Tức là tư duy đã có một bộ nhớ để lưu lại những gì diễn ra trước đó.

Tuy nhiên các mô hình mạng nơ-ron truyền thống thì không thể làm được việc lưu trữ như vậy, đó có thể coi là một khuyết điểm chính của mạng nơ-ron truyền thống. Ví dụ, ta muốn phân loại các bối cảnh xảy ra ở tất cả các thời điểm trong một bộ phim, thì đúng là không rõ làm thế nào để có thể hiểu được một tình huống trong phim mà lại phụ thuộc vào các tình huống trước đó nếu sử dụng các mạng nơ-ron truyền thống.

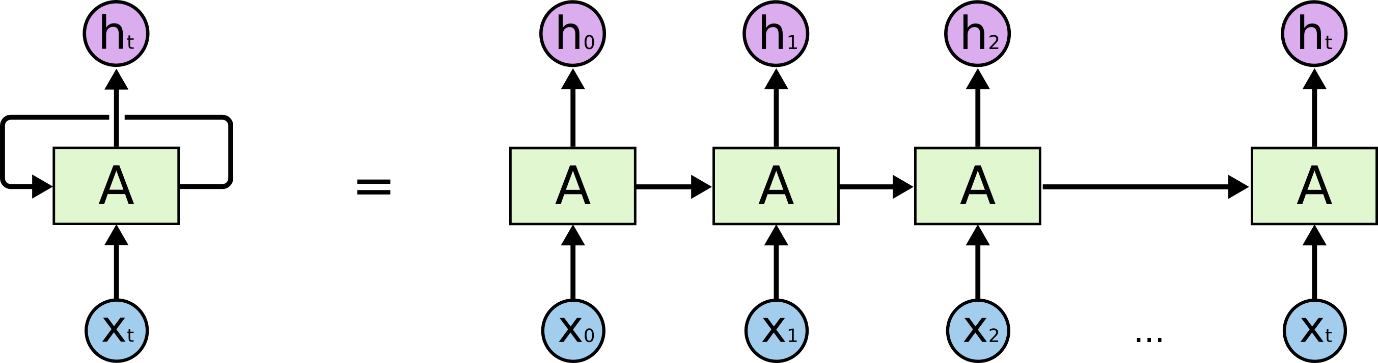
Mạng nơ-ron hồi quy (Recurrent Neural Network) sinh ra để giải quyết vấn đề đó. Mạng này chứa các vòng lặp bên trong cho phép thông tin có thể lưu lại được.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/RNN-rolled.png)

Hình . Recurrent Neural Networks có vòng lặp

Hình vẽ trên mô tả một đoạn của mạng nơ-ron hồi quy A với đầu vào là ​ và đầu ra là ​. Một vòng lặp cho phép thông tin có thể được truyền từ bước này qua bước khác của mạng nơ-ron.

Các vòng lặp này khiến cho mạng nơ-ron hồi quy trông có vẻ khó hiểu. Tuy nhiên, nếu ta để ý một chút thì nó không khác mấy so với các mạng nơ-ron thuần. Một mạng nơ-ron hồi quy có thể được coi là nhiều bản sao chép của cùng một mạng, trong đó mỗi đầu ra của mạng này là đầu vào của một mạng sao chép khác. Ta có hình mô tả như sau:

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/RNN-unrolled.png)

Hình . An unrolled recurrent neural network

Chuỗi lặp lại các mạng này chính là phân giải của mạng nơ-ron hồi quy, các vòng lặp khiến chúng tạo thành một chuỗi danh sách các mạng sao chép nhau. Ta có thể thấy Recurrent Neural Network giống với một mạng nơ-ron đơn thuần. Các nút của mạng vẫn nhận đầu vào và có đầu ra hệt như mạng nơ-ron thuần.

Việc ứng dụng RNN đã đưa ra được nhiều kết quả không thể tin nổi trong nhiều lĩnh vực: nhận dạng giọng nói, mô hình hóa ngôn ngữ, dịch máy, mô tả ảnh,… Danh sách vẫn còn đang được mở rộng tiếp. Anh Andrej Karpathy đã đề cập tới một số kết quả mà RNN mang lại.

Đằng sau sự thành công này chính là sự đóng góp của LSTM. LSTM là một dạng đặc biệt của mạng nơ-ron hồi quy, với nhiều bài toán thì nó tốt hơn mạng hồi quy thuần. Hầu hết các kết quả thú vị thu được từ mạng RNN là được sử dụng với LSTM.

#### Vấn đề phụ thuộc xa

Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại. Việc này tương tự như ta sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời. Nếu mà RNN có thể làm được việc đó thì chúng sẽ cực kì hữu dụng, tuy nhiên để có thể làm được còn tùy vào một số vấn đề*.*

Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại. Ví dụ, ta có câu: “các đám mây trên bầu trời” thì ta chỉ cần đọc tới “các đám mây trên bầu” là đủ biết được chữ tiếp theo là “trời” rồi. Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ, nên RNN hoàn toàn có thể học được.

[Graphical user interface, application

Description automatically generated](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/RNN-shorttermdepdencies.png)

Hình . Vấn đề phụ thuộc xa

Nhưng trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận. Ví dụ, dự đoán chữ cuối cùng trong đoạn: “I grew up in France… I speak fluent French.”. Rõ ràng là các thông tin gần (“I speak fluent”) chỉ có phép ta biết được đằng sau nó sẽ là tên của một ngôn ngữ nào đó, còn không thể nào biết được đó là tiếng gì. Muốn biết là tiếng gì, thì ta cần phải có thêm ngữ cảnh “I grew up in France” nữa mới có thể suy luận được. Rõ ràng là khoảng cách thông tin lúc này có thể đã khá xa rồi.

Thật không may là với khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa.

[Ảnh có chứa văn bản, đồng hồ, thiết bị đo

Mô tả được tạo tự động](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/RNN-longtermdependencies.png)

Hình . Vấn đề phụ thuộc xa

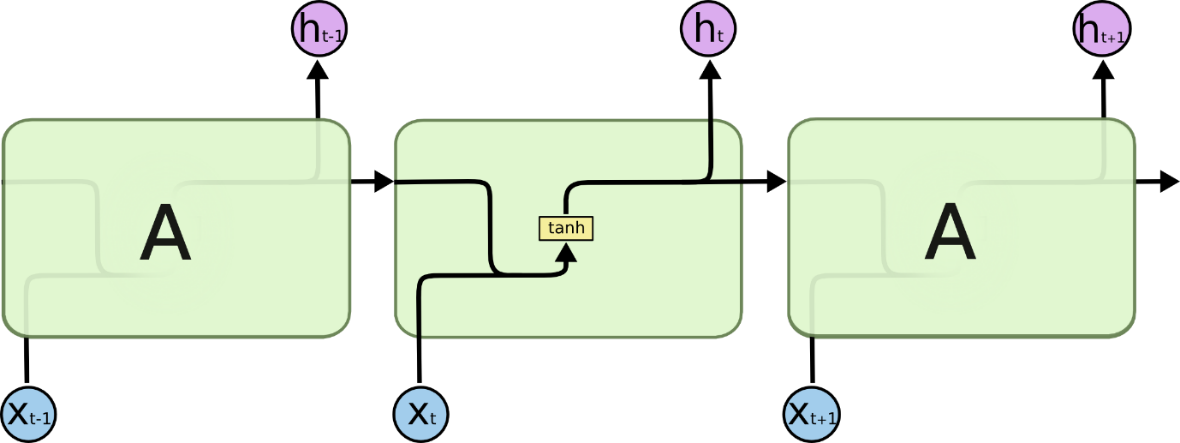
Về mặt lý thuyết, rõ ràng là RNN có khả năng xử lý các phụ thuộc xa (long-term dependencies). Chúng ta có thể xem xét và cài đặt các tham số sao cho khéo là có thể giải quyết được vấn đề này. Tuy nhiên, đáng tiếc trong thực tế RNN có vẻ không thể học được các tham số đó. Vấn đề này đã được khám phá khá sâu bởi [**Hochreiter (1991)**](http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf) **và**[**Bengio (1994)**](http://www-dsi.ing.unifi.it/~paolo/ps/tnn-94-gradient.pdf), trong các bài báo của mình, họ đã tìm được nhưng lý do căn bản để giải thích tại sao RNN không thể học được. Tuy nhiên, LSTM không vấp phải vấn đề đó.

#### Mạng LSTM

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks) thường được gọi là LSTM - là một biến thể của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi [**Hochreiter & Schmidhuber (1997)**](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf) và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

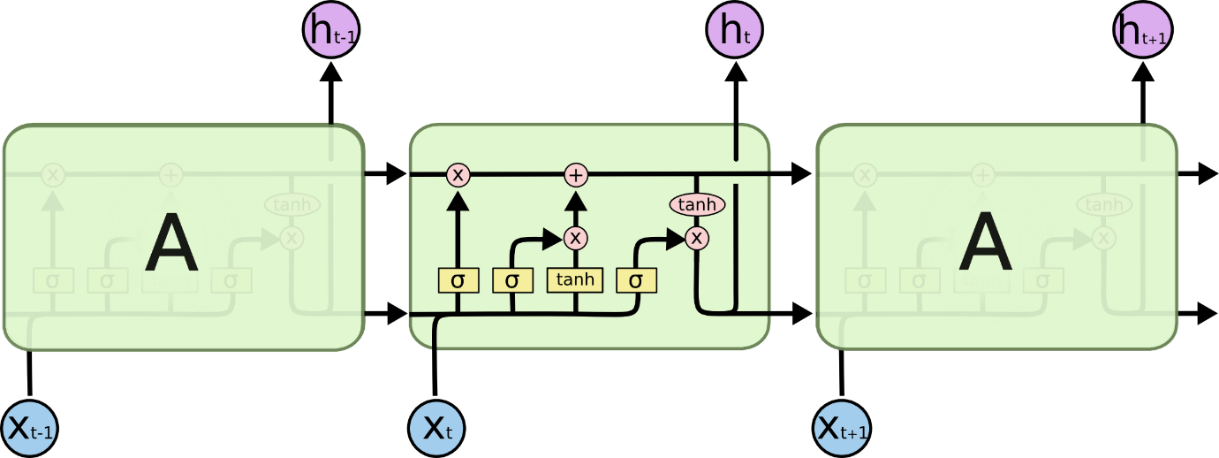
LSTM được thiết kế để tránh vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-SimpleRNN.png)

Hình . Mô-đun trong RNN tiêu chuẩn chứa một layer

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-chain.png)

Hình . Mô-đun trong LSTM chứa 4 layers

Một số ký hiệu trong LSTM:

[Diagram

Description automatically generated](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM2-notation.png)

Ở sơ đồ trên, mỗi một đường mang một véc-tơ từ đầu ra của một nút tới đầu vào của một nút khác. Các hình trong màu hồng biểu diễn các phép toán như phép cộng véc-tơ chẳng hạn, còn các ô màu vàng được sử dụng để học trong các từng mạng nơ-ron. Các đường hợp nhau kí hiệu việc kết hợp, còn các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và chuyển tới các nơi khác nhau.

*a, Ý tưởng cốt lõi của LSTM*

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ. Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.

Ảnh có chứa văn bản, đồng hồ

Mô tả được tạo tự động

Hình . Cell state (C)

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.



Hình . Gates

Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0, 1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

*b, Bên trong LSTM*

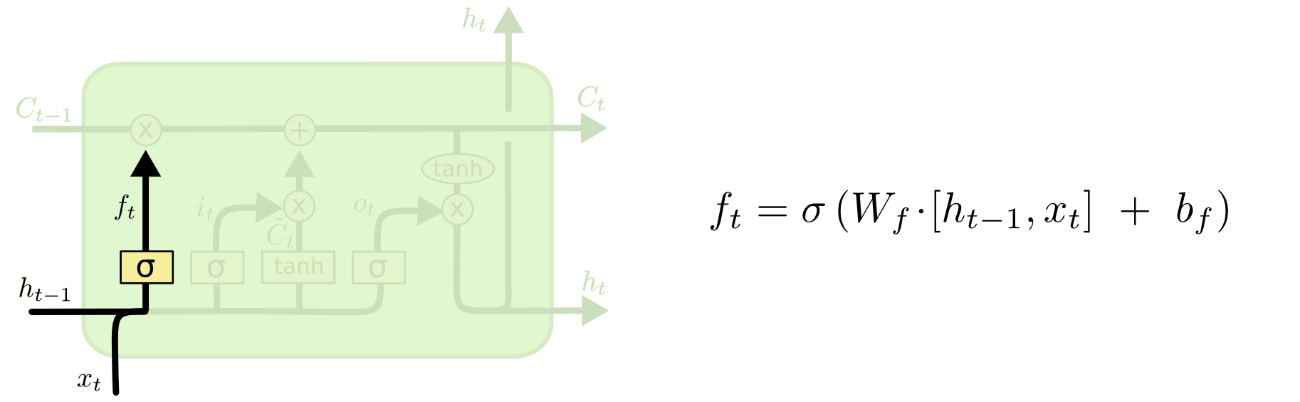
Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là "tầng cổng quên" (forget gate layer). Nó sẽ lấy đầu vào là và rồi đưa ra kết quả là một số trong khoảng [0, 1] cho mỗi số trong trạng thái tế bào . Đầu ra là 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.

LSTM gồm 4 thành phần chính: **cell state**, **input gate**, **output gate** và **forget gate**. Các cell có nhiệm vụ nhớ các giá trị trong khoảng thời gian phụ thuộc và giá trị của cổng **forget gate**. Ba cổng có nhiệm vụ điều chỉnh luồng thông tin vào ra khỏi cell.

Mỗi cell trong LSTM network có khả năng xử lý dữ liệu một cách tuần tự. LSTM rất phù hợp cho bài toán phân loại với dữ liệu theo chuỗi thời gian. LSTMs được phát triển để giải quyết vấn đề đạo hàm bị triệt tiêu (Vanishing gradient)

Với ví dụ mô hình ngôn ngữ dự đoán từ tiếp theo dựa trên tất cả các từ trước đó, với những bài toán như vậy, thì trạng thái tế bào có thể sẽ mang thông tin về giới tính của một nhân vật nào đó giúp ta sử dụng được đại từ nhân xưng chuẩn xác. Tuy nhiên, khi đề cập tới một người khác thì ta sẽ không muốn nhớ tới giới tính của nhân vật nữa, vì nó không còn tác dụng gì với chủ thể mới này. Đầu ra nằm trong khoảng [0, 1] sẽ cho biết sẽ loại đi những gì.

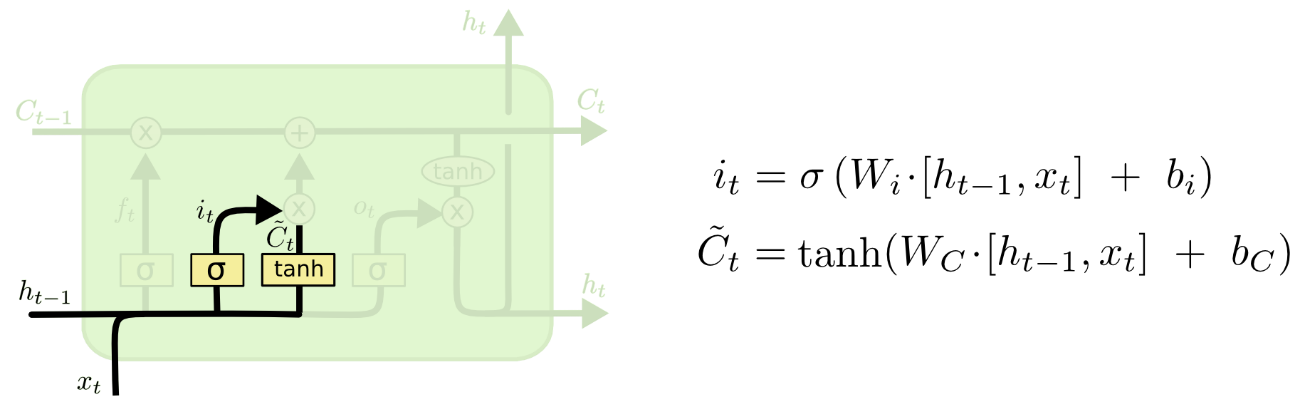
Công thức:



Hình . Forget gate

Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là “tầng cổng vào" (input gate layer) để quyết định giá trị nào ta sẽ cập nhập. Tiếp theo là một tầng tanh tạo ra một véc-tơ cho giá trị mới nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái. Chẳng hạn với ví dụ mô hình ngôn ngữ của ta, ta sẽ muốn thêm giới tính của nhân vật mới này vào trạng thái tế bào và thay thế giới tính của nhân vật trước đó.

Công thức:

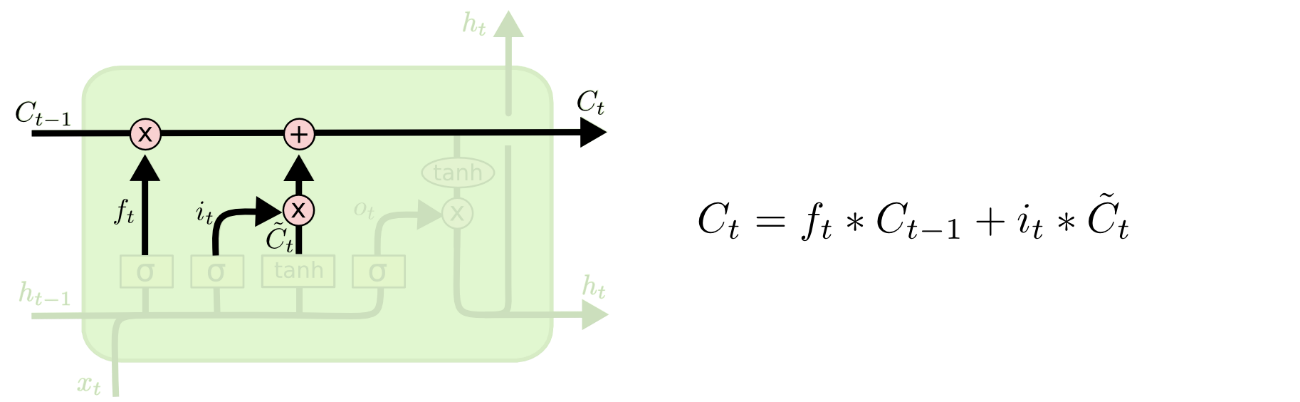


Hình . Input gate

Giờ là lúc cập nhập trạng thái tế bào cũ thành trạng thái mới . Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện là xong. Ta sẽ nhân trạng thái cũ với để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm \* . Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao.

Với bài toàn mô hình ngôn ngữ, chính là việc ta bỏ đi thông tin về giới tính của nhân vật cũ, và thêm thông tin về giới tính của nhân vật mới như ta đã quyết định ở các bước trước đó.

Công thức:



Hình . Update Cell state

Cuối cùng, ta cần quyết định xem ta muốn đầu ra là gì. Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm tanh để co giá trị nó về khoảng [-1, 1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.

Với ví dụ về mô hình ngôn ngữ, chỉ cần xem chủ thế mà ta có thể đưa ra thông tin về một trạng từ đi sau đó. Ví dụ, nếu đầu ra của chủ thể là số ít hoặc số nhiều thì ta có thể biết được dạng của trạng từ đi theo sau nó phải như thế nào.

Công thức:

Ảnh có chứa văn bản, đồng hồ

Mô tả được tạo tự động

Hình . Output gate

### Xây dựng mô hình

#### Mô hình LSTM với các tham số mặc định

**Chart, histogram

Description automatically generated**

Kết quả:

*Graphical user interface, text, application

Description automatically generated*

#### Mô hình LSTM với các tham số đã điều chỉnh

**Chart, histogram

Description automatically generated**

Kết quả:

**Graphical user interface, text, application

Description automatically generated**

**Nhìn vào kết quả, ta thấy mô hình LSTM cải tiến có kết quả tốt hơn nhiều trên tập Test so với mô hình LSTM cơ bản.**

Các tham số đã điều chỉnh:

**Units**: số layer ẩn đặc trưng cho độ phức tạp của mạng

Table

Description automatically generated

Chọn units = 32 Vì lượng dữ liệu train không lớn nên không lựa chọn units cao để làm phức tạp mạng hơn.

**Batch\_size**: số lượng mẫu dữ liệu trong một batch

Table

Description automatically generated

Chọn batch\_size = 64. Vì dữ liệu không lớn nên muốn lượt duyệt danh sách lâu hơn, thời gian học nhiều hơn.

### Đánh giá mô hình

#### Kết quả

**Text

Description automatically generated**

#### Biểu đồ trên tập Test

**Chart

Description automatically generated**

#### Nhận xét

Sau khi huấn luyện mô hình từ tập train và tập validation, chúng tôi thử nghiệm dự đoán trên tập test – đây được coi là tập dữ liệu thực tế như chúng tôi đã phân chia ở mục 4.1.1.

Chúng tôi dự đoán giá giao dịch trong 350 ngày tiếp theo và ta có thể thấy ở mỗi ngày bất kì thì kết quả đường dự đoán thường sẽ cao hơn so với giao dịch thực tế. Điểm đáng chú ý là mô hình đã có thể gần như theo kịp được so với giá trị thực tế, mặc dù vẫn còn chút sai số không thể tranh khỏi như ở mục 4.2.4.6

# KẾT LUẬN

Qua việc ứng dụng mô hình LSTM để giải quyết các bài toán dự đoán giá chứng khoán mô hình học máy đã cho những kết quả dự đoán tốt với độ chính xác cao so với việc dự đoán thủ công thông thường.

Qua đề tải này nhóm em đã phần nào hiểu hơn về các mô hình trong học máy và thấy được ứng dụng của học máy trong cuộc sống đem lại lợi ích nhiều đến như thế nào giúp con người tiết kiệm được rất nhiều thời gian và tiền bạc. Nhóm 03 chúng em xin cảm ơn thầy đã hướngdẫn tận tình trong các bài giảng trên lớp để giúp nhóm chúng em có thể hiểu hơn về các mô hình học máy và có thể hoàn thành project này.

**BẢNG PHÂN CÔNG**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Người thực hiện** |
| **01** | Xác định yêu cầu bài toán | Ngô Đức Thịnh |
| **02** | Lấy dữ liệu | Dương Gia Khánh |
| **02** | Tiền xử lý dữ liệu, Xử lý dữ liệu | Vương Trung Thành |
| **03** | Lý thuyết LSTM | Vương Trung Thành |
| **04** | Xây dựng mô hình | Nguyễn Trần Lê Tuấn |
| **05** | Đánh giá mô hình | Nguyễn Trần Lê Tuấn |
| **06** | Cải thiện tối ưu mô hình | Ngô Đức Thịnh |
| **07** | Hoàn thiện báo cáo | Dương Gia Khánh |

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Slides môn Học máy của giảng viên Ths. Nguyễn Đình Quý |
| [2] | [Understanding LSTM Networks -- colah's blog](https://colah.github.io/posts/2015-08-Understanding-LSTMs/) |
| [3] | Deep learning cơ bản - Nguyễn Thanh Tuấn |

[4] [CS 230 - Recurrent Neural Networks Cheatsheet (stanford.edu)](https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#architecture)