HỌ TÊN SINH VIÊN: NGUYỄN VĂN NGHĨA

LỚP: D19HTTT1

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Đề tài: ÁP DỤNG MÔ HÌNH HỌC MÁY XÂY DỰNG ỨNG DỤNG DỰ ĐOÁN CHỨNG KHOÁN VIỆT NAM

Giảng viên hướng dẫn:	ThS. Đinh Xuân Trường
Sinh viên thực hiện:	Nguyễn Văn Nghĩa
Mã sinh viên:	B19DCCN469
Lớp:	D19HTTT1
Niên khóa:	2019-2024
Hệ đào tạo:	Đại học chính quy

Hà Nội 2024

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Đề tài: ÁP DỤNG MÔ HÌNH HỌC MÁY XÂY DỰNG ỨNG DỤNG DỰ ĐOÁN CHỨNG KHOÁN VIỆT NAM

Giảng viên hướng dẫn:	ThS. Đinh Xuân Trường
Sinh viên thực hiện:	Nguyễn Văn Nghĩa
Mã sinh viên:	B19DCCN469
Lớp:	D19HTTT1
Niên khóa:	2019-2024
Hệ đào tạo:	Đại học chính quy

Hà Nội 2024

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Điểm: **(Bằng chữ:)**

Hà Nội, ngày tháng năm 20

Giảng viên hướng dẫn

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Điểm: **(Bằng chữ:** **)**

Hà Nội, ngày tháng năm

Giảng viên phản biện

LỜI CẢM ƠN

Lời đầu tiên, em xin cảm ơn thầy Trường với những kiến thức vô cùng bổ ích và cho em những góp ý để em có thể hoàn thành được đồ án tốt nghiệp này. Việc được thực hiện đồ án giúp em củng cố thêm kiến thức, có cơ hội để tiếp cận với một lĩnh vực mới, biết thêm một hướng mới trong công nghệ và trên hết là thử thách bản thân mình.

Trong quá trình thực hiện đồ án tốt nghiệp, em nhận thấy mình đã cố gắng hết sức vì mặc dù gặp khá nhiều khó khăn trong việc tìm hiểu và thực hành. Do lượng kiến thức vẫn còn hạn hẹp nên trong bài báo cáo của em còn nhiều thiếu sót mong thầy có thể bổ sung để đồ án mà em thực hiện được hoàn thiện hơn.

Mặc dù đã cố gắng hoàn thành báo cáo trong phạm vi cho phép nhưng bài báo cáo chắc chắn sẽ không tránh được những sự thiếu sót. Em kính mong nhận được sự thông cảm của thầy cô và các bạn.

Em xin chân thành cảm ơn!

Hà Nội, ngày tháng 01 năm 2024

Sinh viên

Nguyễn Văn Nghĩa

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

STT	Từ viết tắt	Diễn giải
1	ANN	Artificial Neural Network
2	API	Application Programming Interface
3	HATEOAS	Hypermedia As The Engine Of Application State
4	HTML	HyperText Markup Language
5	HTTP	HyperText Transfer Protocol
6	IDE	Integrated Development Environment
7	LSTM	Long short-term memory
8	MACD	Moving Average Convergence Divergence
9	MAE	Mean Absolute Error
10	MAPE	Mean Absolute Percentage Error
11	noSQL	Not-only SQL or No-relation SQL
12	RMSE	Root Mean Square Error
13	RNN	Recurrent Neural Network
14	RSI	Relative Strength Index
15	SQL	Structured Query Language

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

TÓM TẮT NỘI DUNG ĐỒ ÁN

Những năm trở lại đây thị trường chứng khoán đã có nhiều biến động. Từ một thị trường vô cùng nóng với rất nhiều người đổ xô vào mua cổ phiếu rồi trở thành thị trường ảm đạm nhất. Đã có rất nhiều nhà đầu tư đổ tiền vào tuy nhiên vốn kiến thức của họ về những mã cổ phiếu, cách phân tích và cách đầu tư là chưa đủ. Cùng với sự phát triển của khoa học và công nghệ đã có rất nhiều phần mềm và công cụ hỗ trợ để giúp mọi người có nhiều góc nhìn về thị trường chứng khoán. Vì vậy, nhằm giúp cho các nhà đầu tư có thêm một cái nhìn đánh giá sơ bộ về biến động giá cả của thị trường chứng khoán Việt Nam và áp dụng những kiến thức và công nghệ đã tìm hiểu em đã quyết định nghiên cứu và thực hiện đề tài "Áp dụng mô hình học máy xây dựng ứng dụng dự đoán chứng khoán Việt Nam".

Đề tài sử dụng chủ yếu nguồn dữ liệu trong vài năm trở lại đây qua mô hình huấn luyện sử dụng thuật toán Bộ nhớ dài - ngắn hạn (LSTM) để đánh giá và dự đoán giá cổ phiếu trong 3 ngày tới. Phần mềm được xây dựng để sử dụng trên hệ điều hành Android. Sử dụng Python, Kotlin và một số thư viện, ứng dụng khác để lập trình nên phần mềm.

Kết quả sau cùng đạt được của đề tài này là hoàn thành một phần mềm hoàn chỉnh giúp cho người dùng có thể tham khảo thêm để quyết định đầu tư và mua bán cổ phiếu.

Nội dung của đồ án bao gồm những chương sau:

Chương 1 trình bày về giới thiệu đề tài, đặt vấn đề cũng như các giải pháp đang tồn tại trong cuộc sống. Bên cạnh đó ở trong chương sẽ nói về những tìm hiểu về bài toán dự đoán giá chứng khoán, những khó khăn đang tồn tại và những phương pháp đã có để giải quyết bài toán này. Trong chương 1 cũng trình bày cơ sở lý thuyết về chứng khoán, bộ nhớ dài ngắn hạn LSTM và các công nghệ được sử dụng trong đề tài

Chương 2 sẽ trình bày về việc lấy dữ liệu chứng khoán Việt Nam bằng Vnstock, sử dụng thuật toán học máy LSTM, xây dựng mô hình, huấn luyện mô hình và đánh giá mô hình để cho ra kết quả chính xác nhất

Chương 3 trình bày về việc phân tích thiết kế và triển khai hệ thống. Phần phân tích bao gồm phân tích chức năng, các Use Case và đặc tả của chúng. Phần thiết kế bao gồm sơ đồ lớp, cơ sở dữ liệu và tổng quan về hệ thống vật lý. Cuối cùng là trình bày sản phẩm bao gồm hướng dẫn cài đặt, demo phần mềm, các màn hình có trong phần mềm.

CHƯƠNG 1. TÌM HIỂU VỀ BÀI TOÁN DỰ ĐOÁN GIÁ CHỨNG KHOÁN VIỆT NAM

1.1 Đặt vấn đề

Hiện tại thị trường chứng khoán đã ảm đạm hơn so với vài năm về trước. Trong thời điểm Covid 19, tình hình kinh tế đi xuống, người dân không được ra khỏi nhà nên đó là thời điểm có nhiều người tìm đến chứng khoán như một hình thức tạo ra nguồn thu nhập. Tuy nhiên họ lại chạy theo những lời mời, giới thiệu, phân tích của người khác mà không có nhiều kiến thức về chứng khoán dẫn đến người được cũng nhiều nhưng chủ yếu là người thua. Vậy nên em đã nghiên cứu và phát triển nên phần mềm dự đoán giá chứng khoán Việt Nam nhằm giúp cho người dùng có một cách nhìn tóm tắt về thị trường, về biến động giá cả của các cổ phiếu. Giá cổ phiếu biến động theo thời gian nên qua sự tìm hiểu em đã chọn xây dựng phần mềm dự đoán dựa trên mô hình Bộ nhớ dài - ngắn hạn (LSTM) nhằm huấn luyện để dự đoán giá cổ phiếu ngày tiếp theo dựa vào giá trước đó.

Để có thể xây dựng được hệ thống cần có kiến thức cơ bản về chứng khoán như thế nào là giá mở cửa, giá đóng cửa, giá cao nhất, giá thấp nhất, khối lượng giao dịch,... Đối với việc xây dựng hệ thống thì cần có kiến thức về lập trình, về các ngôn ngữ với các chức năng như: Python dùng để xây dựng model, Kotlin để xây dựng phần mềm chạy ở nền tảng Android. Các thư viện dùng trong việc lấy data, tạo model, huấn luyện model và trực quan hoá dữ liệu như: vnstock, matplotlib, keras, tensorflow,...

1.2 Ngữ cảnh của bài toán và bài toán Time Series Forecasting

Bài toán xây dựng phần mềm dự đoán giá cổ phiếu được đặt trong bối cảnh phức tạp của thị trường tài chính và đầu tư. Đây là một dự án quan trọng nhằm cung cấp cho các nhà đầu tư một công cụ mạnh mẽ để đưa ra quyết định đầu tư thông minh. Dự án này bao gồm việc thu thập và xử lý dữ liệu tài chính, như giá cổ phiếu lịch sử, thông tin về công ty, chỉ số kinh tế, và các yếu tố thị trường khác.

Phần mềm sử dụng các mô hình dự đoán, có thể là mô hình học máy, học sâu hoặc thống kê, để dự đoán giá cổ phiếu trong tương lai. Điều này đòi hỏi sự lựa chọn, xây dựng và đào tạo mô hình phù hợp, cũng như việc liên tục cập nhật dữ liệu để duy trì tính chính xác của dự đoán.

Ngoài ra, phần mềm cần có giao diện thân thiện để người sử dụng có thể tương tác với hệ thống, nhập thông tin, và nhận kết quả dự đoán. Chính sách và quy định trong lĩnh vực tài chính cũng cần được tuân thủ, và phần mềm cần cung cấp thông tin liên quan đến rủi ro đầu tư.

Hiện tại cũng đã có rất nhiều phần mềm liên quan đến chứng khoán nhưng chỉ làm về phần phân tích, hiển thị số liệu hoặc là làm dự đoán giá các mã chứng khoán nhưng chủ yếu là mã chứng khoán nước ngoài.

Tất cả những yếu tố này kết hợp lại tạo ra một bài toán phức tạp yêu cầu hiểu biết sâu về tài chính, khoa học dữ liệu, và công nghệ phần mềm. Mục tiêu

của phần mềm này là hỗ trợ người sử dụng đưa ra quyết định đầu tư thông minh trong môi trường đầu tư chứng khoán Việt Nam đầy biến động và không chắc chắn.

Dữ liệu sử dụng và trực quan hoá dữ liệu trong hệ thống được có sự thay đổi theo ngày nên kiểu dữ liệu Time Series sẽ được sử dụng ở trong đề tài này. Kiểu dữ liệu Time Series giúp người dùng hiểu rõ hơn về xu hướng của các mức giá cổ phiếu khác nhau, việc phân tích chuỗi thời gian cũng thường được các nhà giao dịch chứng khoán sử dụng. Đặc biệt hữu ích là các biểu đồ chuỗi thời gian, giúp các nhà phân tích và nhà giao dịch chứng khoán hiểu được xu hướng và hướng đi của một giá cổ phiếu cụ thể [1].

Hình 1.1 Biểu đồ giá chứng khoán

Ở trên là ví dụ về dữ liệu giá đóng cửa cổ phiếu của Apple với dữ liệu thay đổi theo thời gian từ trái sang phải. Sự chênh lệch về giá và tăng giảm được thể hiện qua việc lên xuống của biểu đồ từ đó giúp cho nhà đầu tư có cái nhìn về giá cả của mã cổ phiếu đó, hỗ trợ người dùng ra quyết định trong việc đầu tư vào mã cổ phiếu.

1.3 Các giải pháp hiện tại và hạn chế

Hiện nay, trong lĩnh vực dự đoán giá chứng khoán đã có nhiều giải pháp vô cùng đa dạng. Chỉ báo kỹ thuật là một phương pháp phổ biến trong việc phân tích kỹ thuật của chứng khoán. Một số chỉ báo như RSI, MACD,... Chỉ số sức mạnh tương đối RSI dùng để đo lường mức độ thay đổi giá cổ phiếu so với biến động giá trong quá khứ bằng cách so sánh số ngày tăng điểm với số ngày giảm điểm. Chỉ báo MACD giúp cung cấp các biến động của thị trường, hỗ trợ nhà đầu tư chứng khoán xác định tín hiệu mua bán của thị trường. Bên cạnh các phương pháp về mặt kỹ thuật thì có các phương pháp sử dụng phân tích cơ bản tập trung vào sức khỏe tài chính của các công ty phát hành cổ phiếu. Nó xem xét các yếu tố như lợi nhuận, doanh thu, và cơ cấu vốn để đưa ra dự đoán về giá cổ phiếu. Ngoài ra sự biến động về giá cả của cổ phiếu còn được dự đoán qua tin tức về

các công ty phát hành mã cổ phiếu và các thông tin khác liên quan. Các giải pháp trên đều có thể sử dụng thông qua các phần mềm đã được lập trình sẵn, các biểu đồ và số liệu đã được trực quan hoá giúp cho người dùng rút ngắn thời gian phân tích. Các thuật toán để xử lý ngôn ngữ tự nhiên được lấy qua các bài báo cũng đã có giúp cho con người tăng năng suất cũng như hiệu quả công việc phân tích dữ liệu, thay vì phải phân tích thủ công thì đã có máy làm giúp phần đó.

Mặc dù các giải pháp để dự đoán chứng khoán đã xuất hiện nhiều tuy nhiên thị trường chứng khoán là một môi trường biến động, và không phải lúc nào cũng tuân theo các mô hình cổ điển. Điều này làm cho dự đoán trở nên khó khăn và không chắc chắn. Các tin tức và sự kiện toàn cầu có thể ảnh hưởng đáng kể đến thị trường chứng khoán, nhưng chúng thường không nằm trong phạm vi dự đoán của các mô hình truyền thống. Có những tin tức độ xác thực không cao nhưng có thể làm ảnh hưởng đến quyết định mua bán cổ phiếu của nhà đầu tư. Bên cạnh đó, sự biến động không chỉ xuất phát từ chính tài sản chứng khoán, mà còn từ các thị trường phái sinh và các sản phẩm tài chính phức tạp khác.

1.4 Những khó khăn của việc dự đoán giá cổ phiếu

Vấn đề thứ nhất cần nói tới đó là về dữ liệu chứng khoán. Để dự đoán giá chứng khoán của ngày hôm sau thì cần có giá của những ngày trước đó nên tập dữ liệu về thông tin của các mã chứng khoán là vô cùng quan trọng. Dữ liệu chính xác thì việc dự đoán giá mới có thể chính xác được. Hiện nay có rất nhiều trang web cung cấp thông tin về các mã chứng khoán tuy nhiên nguồn dữ liệu nếu chỉ xử lý và lấy bằng tay thì rất mất thời gian. Bên cạnh đó độ chính xác của dữ liệu cũng ảnh hưởng đến kết quả của hệ thống. Vậy nên cần có kĩ thuật xử lý để có thể lấy được dữ liệu một cách đầy đủ và chính xác nhằm giảm rủi ro cho việc đầu tư sau này.

Vấn đề thứ hai của việc dự đoán giá chứng khoán đó là sự biến động kinh tế của thị trường. Thị trường chứng khoán có thể thay đổi nhanh chóng và khó dự đoán được. Điều này khiến cho việc xử lý thông tin phải chạy theo thời gian thực và sát với thực tế. Lượng thông tin đó không ngừng thay đổi liên tục nên cần có các phương pháp đánh giá về sự thay đổi nhằm cập nhật thông tin dự đoán hoặc cần có giải pháp để giải quyết vấn đề đó.

Vấn đề tiếp theo là vấn đề quản lý rủi ro của việc đầu tư. Mặc dù việc đầu tư là có thể sinh lời và có thể gây ra thua lỗ, tuy nhiên nhằm thống kê được mức rủi ro là bao nhiêu cũng như quản lý việc đầu tư của người dùng thì cần có chức năng nhằm hạn chế rủi ro nhiều nhất có thể. Xây dựng lên một chức năng có thể đánh giá được tỉ suất lợi nhuận là bao nhiêu, khả năng sinh lời là bao nhiêu cũng như quản lý nguồn đầu tư để người dũng biết được bản thân đã đầu tư bao nhiêu, lời hay lỗ ra sao.

1.5 Mục tiêu và định hướng giải pháp

Mục tiêu của đề tài này là giúp cho người dùng và nhà đầu tư có thêm một góc nhìn khác về giá cả của các mã chứng khoán ở trên thị trường nhằm tối ưu hoá lợi nhuận, giảm thiểu rủi ro cho các nhà đầu tư. Sau khi tìm hiểu về các giải pháp hiện tại cho việc dự đoán chứng khoán, xác định hướng giải quyết cho những hạn chế của các giải pháp đó. Để đối phó với tác động của thông tin toàn cầu và tin tức, việc tích hợp các thuật toán xử lý ngôn ngữ tự nhiên để phân tích và hiểu các tin tức và bài báo trở thành một giải pháp quan trọng. Rủi ro về việc đầu tư là điều có thể xảy ra, vì vậy việc áp dụng chiến lược quản lý rủi ro một cách thông minh. Một giải pháp quan trọng đó là kết hợp giữa cách phân tích kỹ thuật, phân tích truyền thống và mô hình học máy. Kết hợp các phân tích cơ bản với dự đoán dựa trên mô hình học máy có thể giúp cải thiện sự chính xác của dự đoán. Phân tích cơ bản giúp hiểu sâu về sức khỏe tài chính của công ty, trong khi mô hình học máy có thể xử lý thông tin lịch sử và thời gian thực.

1.6 Đóng góp của đề tài

Đề tài này có những đóng góp chính như sau:

- Đề tài đưa ra các khảo sát về thị trường chứng khoán hiện tại của Việt Nam cũng như các công nghệ đang được sử dụng trong việc dự đoán giá chứng khoán
- Đề tài này đóng góp vào việc cải thiện khả năng dự đoán giá chứng khoán ở thị trường Việt Nam bằng việc áp dụng mô hình học máy. Bằng việc sử dụng mô hình LSTM, đề tài cung cấp một phương pháp tiên tiến để phân tích và dự đoán biến động giá cổ phiếu
- Một phần quan trọng của đóng góp là khả năng kết hợp dữ liệu lịch sử với phân tích cơ bản. Bằng cách kết hợp thông tin về sức khỏe tài chính của các công ty với khả năng dự đoán từ mô hình học máy, đề tài này cung cấp một cách tiếp cận toàn diện để đưa ra quyết định đầu tư
- Đề tài không chỉ tập trung vào việc nghiên cứu mà còn xây dựng một ứng dụng thực tế để áp dụng kiến thức và công cụ thu được. Điều này có thể hỗ trợ các nhà đầu tư và người giao dịch trong quyết định giao dịch hàng ngày và quản lý rủi ro
- Đề tài này đóng góp vào việc phát triển nghiên cứu về chứng khoán và công nghệ tại Việt Nam. Nó thể hiện sự phát triển trong việc áp dụng mô hình học máy trong lĩnh vực tài chính và thúc đẩy sự hiểu biết về thị trường chứng khoán trong nước

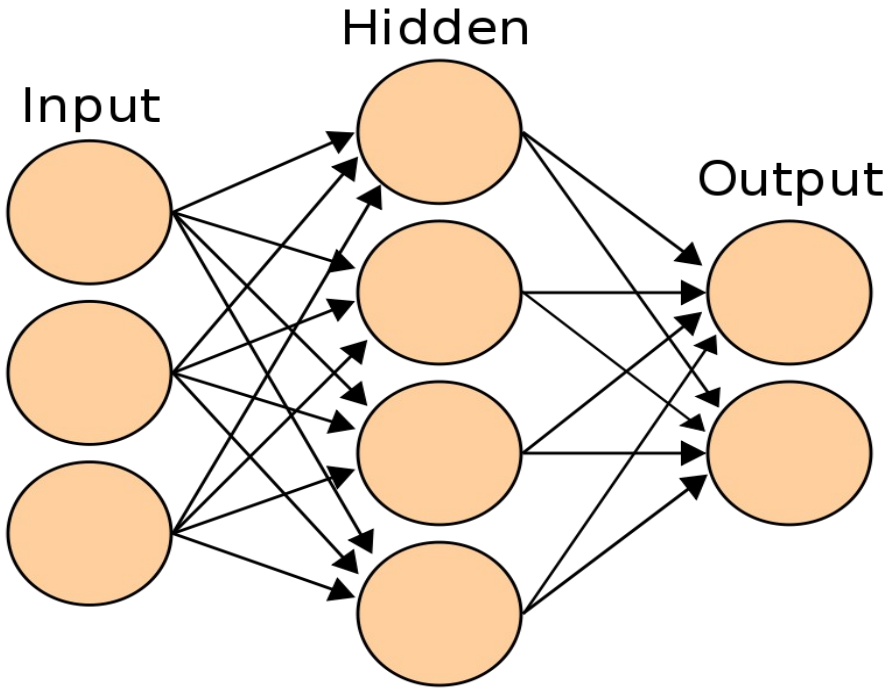
Hình 1.2: Sơ đồ tổng quan lấy dữ liệu và xây dựng mô hình

1.7 Cơ sở mạng nơ-ron nhân tạo

1.7.1 Kiến trúc mạng nơ-ron nhân tạo

Mạng nơ ron nhân tạo (Artificial Neural Network – ANN) là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các nơ-ron được gắn kết để xử lý thông tin. ANN hoạt động giống như bộ não của con người, được học bởi kinh nghiệm (thông qua việc huấn luyện), lưu giữ các tri thức sau đó sử dụng các tri thức đó trong việc dự đoán các dữ liệu chưa biết [1].

Một mạng nơ-ron bao gồm nhiều nút nối với nhau, mô phỏng mạng nơ-ron thần kinh của não người. Mạng nơ-ron nhân tạo bao gồm ba thành phần cơ bản: mô hình của nơ-ron, cấu trúc và sự liên kết giữa các nơ-ron. Trong một số trường hợp, mạng nơ-ron nhân tạo là một hệ thống thích ứng, tự thay đổi cấu trúc của mình dựa trên các thông tin bên ngoài hay bên trong chạy qua mạng trong quá trình học.



Hình 1.3 : Mạng nơ ron nhân tạo[8]

Kiến trúc chung của một ANN gồm 3 thành phần đó là lớp đầu vào, lớp ẩn và lớp đầu ra.

Dữ liệu từ thế giới bên ngoài sẽ đi vào mạng nơ-ron nhân tạo qua lớp đầu vào. Các nút đầu vào xử lý, phân tích, phân loại dữ liệu sau đó chuyển sang lớp tiếp theo.

Lớp ẩn gồm các nơ-ron, nhận dữb liệbu đầu vào từ các nơ-ron ở lớp trước đó và chuyển đổi các dữ liệu này cho các lớp xử lý tiếp theo. Trong một mạng ANN có thể có nhiều lớp ẩn.

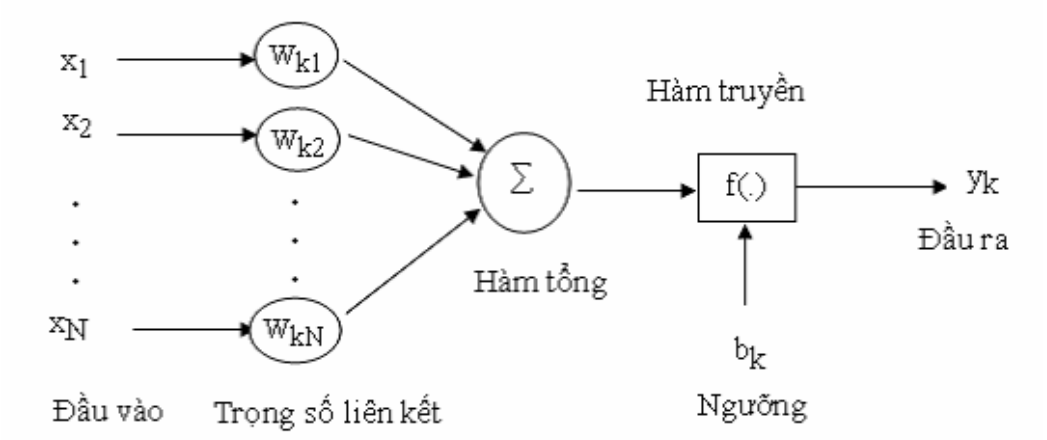
Lớp đầu ra cho kết quả cuối cùng sau khi mạng nơ-ron xử lý dữ liệu. Lớp này có thể có một hoặc nhiều nút nhằm cho ra kết quả của một hoặc nhiều thuộc tính sau khi xử lý.

Lợi thế lớn nhất của các mạng ANN là khả năng được sử dụng như một cơ chế xấp xỉ hàm tùy ý mà "học" được từ các dữ liệu quan sát. Tuy nhiên, sử dụng chúng không đơn giản như vậy cần hiểu biết một số các đặc tính và kinh nghiệm khi thiết kế một mạng nơ-ron ANN:

- Chọn mô hình: Điều này phụ thuộc vào cách trình bày dữ liệu và các ứng dụng. Mô hình quá phức tạp có xu hướng dẫn đền những thách thức trong quá trình học.
- Cấu trúc và sự liên kết giữa các nơ-ron
- Thuật toán học: Có hai vấn đề cần học đối với mỗi mạng ANN, đó là học tham số của mô hình (parameter learning) và học cấu trúc (structure learning). Học tham số là thay đổi trọng số của các liên kết giữa các nơron trong một mạng, còn học cấu trúc là việc điều chỉnh cấu trúc mạng bằng việc thay đổi số lớp ẩn, số nơ-ron mỗi lớp và cách liên kết giữa chúng. Hai vấn đề này có thể được thực hiện đồng thời hoặc tách biệt.

Nếu các mô hình, hàm chi phí và thuật toán học được lựa chọn một cách thích hợp, thì mạng ANN sẽ cho kết quả có thể vô cùng mạnh mẽ và hiệu quả.

1.7.2 Hoạt động của mạng nơ-ron nhân tạo



Hình 1.4 : Sơ đồ hoạt động mạng nơ ron nhân tạo

Đầu vào: Mỗi thông tin đầu vào tương ứng với 1 đặc trưng của dữ liệu. Ví dụ như trong ứng dụng của ngân hàng xem xét có chấp nhận cho khách hàng vay tiền hay không thì mỗi thông tin đầu vào là một thuộc tính của khách hàng như thu nhập, nghề nghiệp, tuổi, số con,…

Đầu ra: Kết quả của một mạng ANN là một giải pháp cho một vấn đề, ví dụ như với bài toán xem xét chấp nhận cho khách hàng vay tiền hay không thì đầu ra là có hoặc không.

Trọng số liên kết : Đây là thành phần rất quan trọng của một ANN, nó thể hiện mức độ quan trọng, độ mạnh của dữ liệu đầu vào đối với quá trình xử lý

thông tin chuyển đổi dữ liệu từ lớp này sang lớp khác. Quá trình học của ANN thực ra là quá trình điều chỉnh các trọng số liên kết của các dữ liệu đầu vào để có được kết quả mong muốn.

Hàm tổng: Hàm tổng dùng để tính tổng trọng số của tất cả các giá trị đầu vào được đưa vào mỗi nơ-ron. Hàm tổng của một nơ-ron đối với nhiều giá trị đầu vào tính theo công thức sau:

$$Y = \sum_{i=1}^{n} XiWi$$

Trong đó : Xi là giá trị của đầu vào thứ i

Wi là giá trị trọng số thứ i

Hàm truyền: Hàm truyền của một nơ-ron cho biết khả năng kích hoạt của nơ-ron đó còn gọi là kích hoạt bên trong. Các nơ-ron này có thể sinh ra một output hoặc không trong mạng ANN, nói cách khác rằng có thể đầu ra của 1 nơ-ron có thể được chuyển đến layer tiếp trong mạng nơ-ron theo hoặc không. Mối quan hệ giữa hàm tổng và kết quả output được thể hiện bằng hàm truyền.

Việc lựa chọn hàm truyền có tác động lớn đến kết quả đầu ra của mạng ANN. Hàm truyền phi tuyến được sử dụng phổ biến trong mạng ANN là hoặc sigmoid hoặc tanh.

$$f(z) = \frac{1}{1 + \exp(-z)}$$

$$f(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Trong đó, hàm tanh là phiên bản thay đổi tỉ lệ của sigmoid , tức là khoảng giá trị đầu ra của hàm truyền thuộc khoảng [-1, 1] thay vì [0,1] nên chúng còn gọi là hàm chuẩn hóa.

Kết quả xử lý tại các nơ-ron đôi khi rất lớn, vì vậy hàm truyền được sử dụng để xử lý output này trước khi chuyển đến layer tiếp theo. Đôi khi thay vì sử dụng hàm truyền thì sẽ sử dụng giá trị ngưỡng để kiểm soát các đầu ra của các nơ-ron tại một lớp nào đó trước khi chuyển các đầu ra này đến các lớp tiếp theo. Nếu đầu ra của một nơ-ron nào đó nhỏ hơn ngưỡng thì nó sẽ không được chuyển đến lớp tiếp theo.

Mạng nơ-ron dự đoán dựa trên lan truyền thẳng là các phép nhân ma trận cùng với hàm kích hoạt để thu được kết quả đầu ra. Nếu đầu vào x là vector 2 chiều thì có thể tính kết quả dự đoán $\hat{y}$ bằng công thức sau:

$$z_1 = xW_1 + b_1$$
$$a_1 = \tanh(z_1)$$
$$z_2 = a_1W_2 + b_2$$
$$a_2 = \hat{y} = softmax(z_2)$$

Trong đó, zi là đầu vào của lớp thứ i, ai là đầu ra của lớp thứ i sau khi áp dụng hàm kích hoạt. $W1$, $b1$, $W2$, $b2$ là các thông số cần tìm của mô hình mạng nơ-ron.

Huấn luyện để tìm các thông số cho mô hình tương đương với việc tìm các thông số $W1$, $b1$, $W2$, $b2$, sao cho độ lỗi của mô hình đạt được là thấp nhất. Đối với hàm trung bình mũ thì dùng hàm cross-entropy.

Nếu có N dòng dữ liệu huấn luyện, và C nhóm phân lớp (trường hợp này là hai lớp nam, nữ), khi đó hàm mất mát giữa giá trị dự đoán $\hat{y}$ và y được tính như sau:

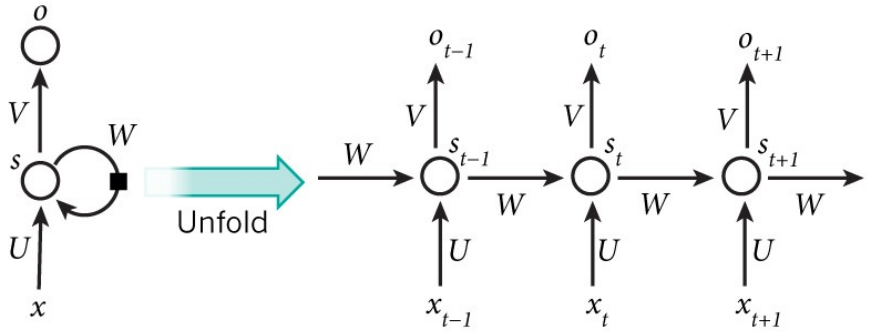
$$L(y, \hat{y}) = -\frac{1}{N}\sum_{n\in N}\sum_{i\in C} y_{n,i} log\hat{y}_{n,i}$$

Ý nghĩa công thức trên nghĩa là: lấy tổng trên toàn bộ tập huấn luyện và cộng dồn vào hàm mất mát nếu kết quả phân lớp sai. Độ dị biệt giữa hai giá trị $\hat{y}$ và y càng lớn thì độ lỗi càng cao. Mục tiêu là tối thiểu hóa hàm lỗi này. Có thể sử dụng phương pháp suy giảm độ dốc để tối tiểu hóa hàm lỗi. Có hai loại thuật toán suy giảm độ dốc, một loại với fixed learning rate được gọi là suy giảm độ dốc hàng loạt, loại còn lại có learning rate thay đổi theo quá trình huấn luyện được gọi là SGD (stochastic gradient descent) hay mini-batch gradient descent.

1.8 Mạng nơ-ron hồi quy Recurrent Neural Network (RNN) và ứng dụng

1.8.1 Mạng nơ-ron hồi quy Recurrent Neural Network (RNN)

Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau, ckhông liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì sẽ cần biết các từ trước đó xuất hiện lần lượt thế nào RNN được gọi là hồi quy bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó, RNN có khả năng nhớ các thông tin được tính toán trước đó [1]. Trên lý thuyết, RNN có thể sử dụng được thông tin của một văn bản rất dài, tuy nhiên thực tế thì nó chỉ có thể nhớ được một vài bước trước đó mà thôi. Về cơ bản một mạng RNN có dạng như sau:



Hình 1.5: Mạng RNN[6]

Mô hình trên mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là vẽ ra một mạng nơ-ron chuỗi tuần tự. Ví dụ có một câu gồm 5 chữ "*Đồ án tốt nghiệp PTIT*", thì mạng nơ-ron được triển khai sẽ gồm 5 tầng nơ-ron tương ứng với mỗi chữ một tầng. Lúc đó việc tính toán bên trong RNN được thực hiện như sau:

- x_t là đầu vào tại bước ***t***. Ví dụ, x_1 là một vec-tơ one-hot tương ứng với từ thứ 2 của câu (*án*). $o_t = x_{t+1}$

- s_t là trạng thái ẩn tại bước *t*. Nó chính là ***bộ nhớ*** của mạng. s_t được tính toán dựa trên cả các trạng thái ẩn phía trước và đầu vào tại bước đó: $s_t = f(Ux_t + Ws_{t-1})$ Hàm f thường là một hàm phi tuyến tính như **tang hyperbolic (tanh)** hay **ReLu**. Để làm phép toán cho phần tử ẩn đầu tiên cần khởi tạo thêm s_{-1}, thường giá trị khởi tạo được gắn bằng 0.

- o_t là đầu ra tại bước ***t***. Ví dụ, muốn dự đoán từ tiếp theo có thể xuất hiện trong câu thì o_t chính là một vec-tơ xác suất các từ trong danh sách từ vựng :

$$o_t = soft\max(Vs_t)$$

1.8.2 Các ứng dụng của Recurrent Neural Network (RNN)

- **Mô hình hoá ngôn ngữ sinh văn bản**

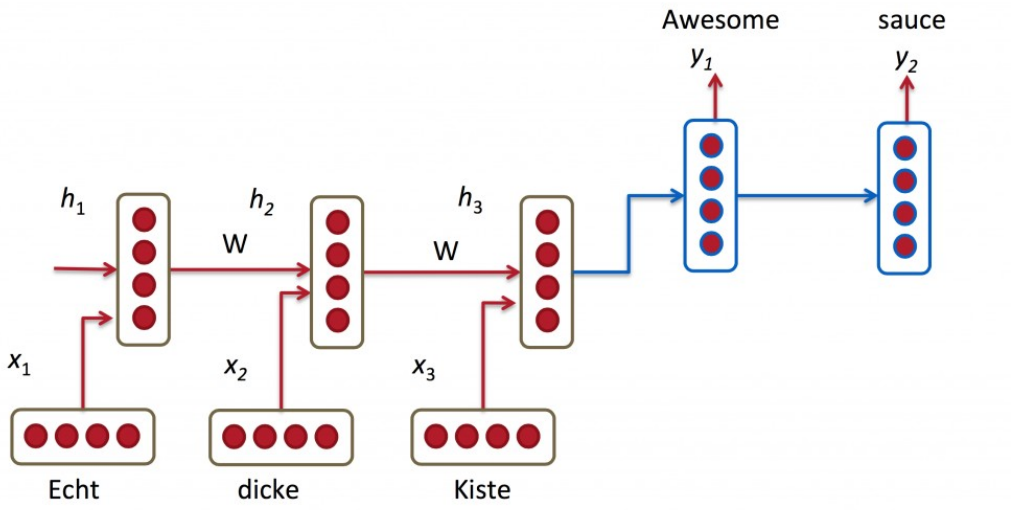
Mô hình ngôn ngữ có thể hỗ trợ trong việc dự đoán tỉ lệ xuất hiện của một từ cụ thể sau chuỗi các từ đã xuất hiện trước đó. Với khả năng ước lượng độ tương tự giữa các câu, mô hình này còn được ứng dụng rất nhiều trong dịch máy. Khả năng dự đoán từ tiếp theo là khả năng giúp cho mô hình có thể tự sinh từ, cho phép máy tính tự động tạo ra văn bản mới từ tập mẫu và xác suất đầu ra của mỗi từ. Tùy thuộc vào mô hình ngôn ngữ mà có thể tạo ra nhiều văn bản khác nhau. Trong cấu trúc mô hình ngôn ngữ, đầu vào thường là một chuỗi các từ (được biểu diễn bằng vectơ one-hot), trong khi đầu ra là chuỗi các từ được dự đoán. Trong quá trình huấn luyện mạng,

thường sử dụng $o_t = x_{t+1}$ với mục tiêu là đầu ra tại bước thời gian t là từ tiếp theo của câu.

Từ bài toán này có thể mở rộng thành bài toán phát sinh văn bản (generating text/generative model). Mô hình này cho phép phát sinh ra văn bản mới dựa vào tập dữ liệu huấn luyện. Ví dụ, khi huấn luyện mô hình này bằng các văn bản truyện Kiều thì có thể phát sinh được các đoạn văn tựa truyện Kiều. Tuỳ theo loại dữ liệu huấn luyện sẽ có nhiều loại ứng dụng khác nhau.

- **Dịch máy**

Dịch máy (Machine Translation) tương tự như mô hình hóa ngôn ngữ ở điểm là đầu vào là một chuỗi các từ trong ngôn ngữ nguồn (ngôn ngữ cần dịch - ví dụ là tiếng Việt). Còn đầu ra sẽ là một chuỗi các từ trong ngôn ngữ đích (ngôn ngữ dịch - ví dụ là tiếng Anh). Điểm khác nhau ở đây là đầu ra chỉ xử lý sau khi đã xem xét toàn bộ chuỗi đầu vào. Vì từ dịch đầu tiên của câu dịch cần phải có đầy đủ thông tin từ đầu vào cần dịch mới có thể suy luận được.



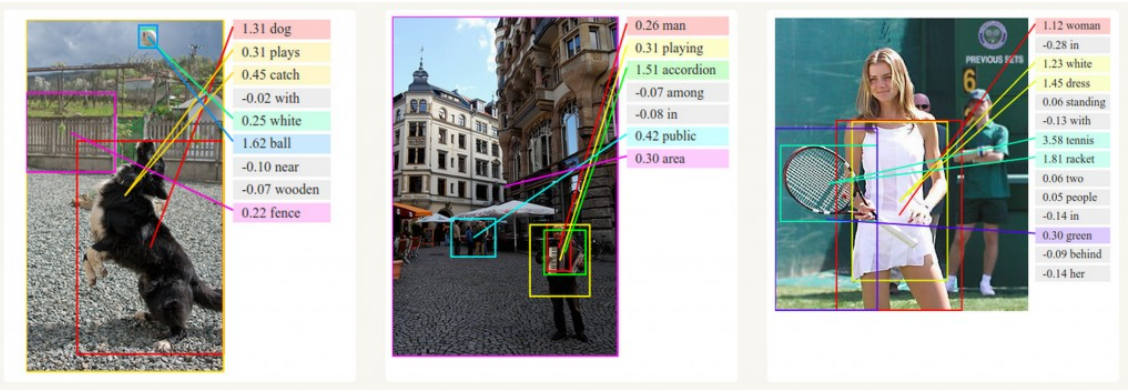
Hình 1.6: Ứng dụng RNN trong Dịch máy[9]

- **Nhận dạng giọng nói**

Đưa vào một chuỗi các tín hiệu âm thanh có thể dự đoán được chuỗi các đoạn ngữ âm đi kèm với xác suất của chúng.

- **Mô tả hình ảnh**

RNN kết hợp với Convolution Neural Networks có thể phát sinh ra được các đoạn mô tả cho ảnh. Mô hình này hoạt động bằng cách tạo ra những câu mô tả từ các đặc trưng rút trích được trong bức ảnh.



Hình 1.7: Ứng dụng RNN trong mô tả hình ảnh[10]

1.8.3 Huấn luyện mạng

Huấn luyện mạng nơ-ron hồi quy (RNN) tương tự như việc huấn luyện mạng nơ-ron truyền thống, nhưng có một số điều chỉnh đặc biệt. Việc huấn luyện này sử dụng thuật toán lan truyền ngược (backpropagation), nhưng có một sự tinh chỉnh nhất định. Điểm quan trọng là tại mỗi đầu ra không chỉ phụ thuộc vào kết quả tính toán của bước hiện tại, mà còn phụ thuộc vào kết quả tính toán của các bước trước đó.

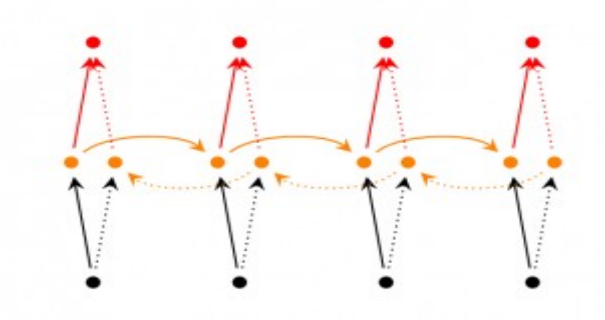
Chẳng hạn, để tính giá trị tại thời điểm t = 4, cần thực hiện lan truyền ngược qua 3 bước thời gian trước đó và cộng dồn các gradient này lại với nhau. Kỹ thuật này được gọi là Lan truyền ngược theo thời gian (Backpropagation Through Time). Một điểm hạn chế của phương pháp này là việc lớp ẩn không giữ được trạng thái nhớ lâu dài và để giải quyết nó, mô hình Long Short-Term Memory (LSTM) đã được phát triển.

Việc giữ được thông tin quan trọng giúp LSTM trở thành một lựa chọn hiệu quả hơn khi xử lý dữ liệu chuỗi có độ dài lớn và giúp cải thiện khả năng học của mô hình trong các nhiệm vụ liên quan đến ngôn ngữ tự nhiên hoặc chuỗi thời gian.

1.8.4 Mở rộng mạng RNN

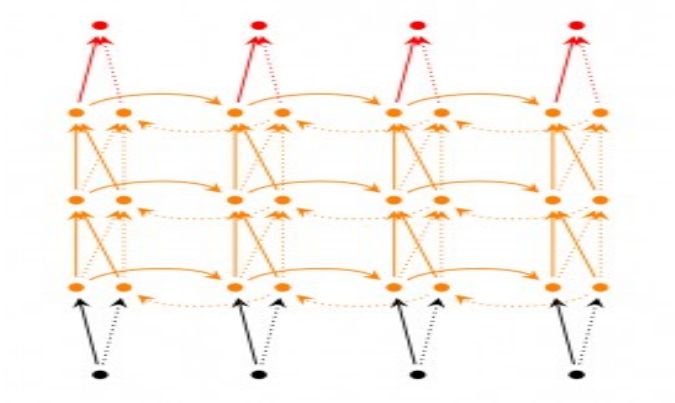
Trong những năm gần đây, các nhà nghiên cứu đã đưa ra nhiều biến thể RNN ngày càng tinh vi để vượt qua những hạn chế của mô hình gốc:

Bidirectional RNN (RNN hai chiều): Xây dựng dựa trên ý tưởng rằng đầu ra tại thời điểm ***t*** không chỉ phụ thuộc vào các thành phần trước đó mà còn phụ thuộc vào các thành phần trong tương lai. Ví dụ, khi dự đoán một từ bị thiếu trong chuỗi, cần quan sát cả từ bên trái và bên phải của từ đó. Mô hình này kết hợp hai RNNs được xếp chồng lên nhau, trong đó trạng thái ẩn được tính toán dựa trên cả hai phía, từ bên trái và bên phải của mạng.



Hình 1.8: Mạng RNN 2 chiều

Deep (Bidirectional) RNN: Tương tự như Bidirectional RNN, nhưng mô hình này bao gồm nhiều tầng Bidirectional RNN tại mỗi thời điểm. Điều này mang lại khả năng thực hiện tính toán phức tạp hơn, tuy nhiên, đòi hỏi tập huấn luyện lớn hơn.



Hình 1.9: Deep RNN

Long Short-Term Memory networks (LSTM): Mô hình này giữ cấu trúc tương tự như RNNs nhưng có cách tính toán khác cho các trạng thái ẩn. Memory trong LSTM được biểu diễn bằng cells (hạt nhân). Nó có khả năng nhận và xử lý thông tin đầu vào bao gồm cả trạng thái ẩn và giá trị, quyết định thông tin nào cần được lưu giữ và thông tin nào cần bị loại bỏ. Nhờ vào cơ chế này, mô hình này có khả năng lưu trữ thông tin lâu dài, giải quyết vấn đề vanishing/exploding gradient của RNN. Đề tài này sẽ sử dụng mô hình LSTM và sẽ được mô tả chi tiết hơn ở mục 2.5

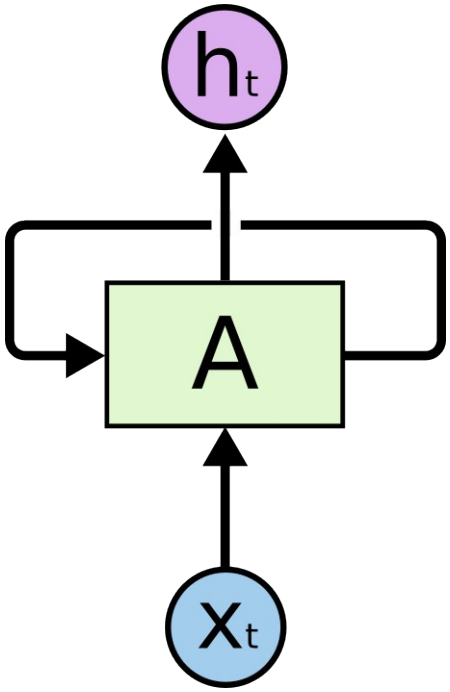
1.9 Bộ nhớ dài – ngắn hạn Long short-term memory (LSTM)

1.9.1 Ưu nhược điểm của RNN và lý do chọn LSTM

LSTM là một mạng cải tiến của RNN nhằm giải quyết vấn đề nhớ các bước dài của RNN.

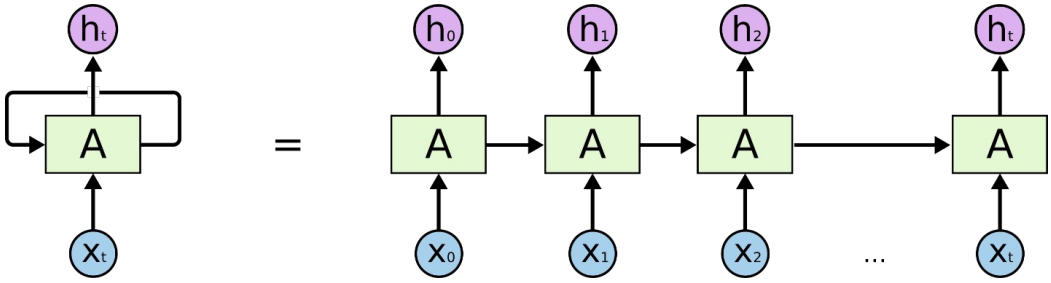
Mạng nơ-ron truyền thống không thể lưu lại dữ liệu đã thực hiện trước đó, đây có thể coi là một khuyết điểm chính của mạng nơ-ron truyền thống. Ví dụ muốn phân loại các bối cảnh xảy ra trong một bộ phim thì cần phải nhớ các tình huống xảy ra trước đó thì mới hiểu được tình huống hiện tại, cái này không thể làm ở mạng nơ-ron truyền thống.

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



Hình 1.10: Mạng RNN

LSTM là một dạng đặc biệt của mạng nơ-ron hồi quy, với nhiều bài toán thì nó tốt hơn mạng tái phát thuần. Việc này tương tự như sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời, nhưng RNN có làm được điều này hay không thì còn tùy thuộc vào hoàn cảnh. Ở một số trường hợp chỉ cần xem lại những thông tin trước đó thôi là có thể biết được tình huống hiện tại và đưa ra dự đoán tiếp theo.



Hình 1.11: Sơ đồ hoạt động mạng RNN

Ở trong một số tình huống buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận. Ví dụ, cần dự đoán chữ cuối cùng trong đoạn: "Tôi sinh ra và lớn lên ở Việt Nam… Tôi có thể sử dụng thành thạo tiếng Việt". Rõ ràng là các thông tin gần ("Tôi có thể sử dụng thành thạo tiếng") chỉ cho 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 cần phải có thêm ngữ cảnh "Tôi sinh ra và lớn lên ở Việt Nam" nữa mới có thể suy luận được. Rõ ràng là khoảng cách thông tin lúc này có thể đã khá xa rồi. Và với khoảng cách càng lớn thì RNN càng không nhớ được nhiều thông tin nữa.

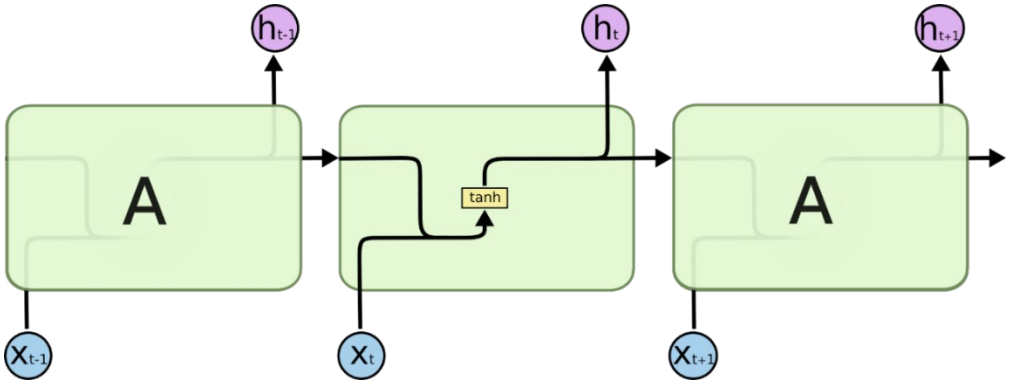
Về mặt lý thuyết, rõ ràng là RNN có khả năng xử lý các phụ thuộc xa. Có thể xem xét và cài đặt các tham số sao cho khéo là có thể giải quyết được vấn đề này. Tuy nhiên, trong thực tế RNN có vẻ không thể học được các tham số đó. Vấn đề này đã được khám phá bởi Hochreiter và Bengio, et al. (1994), trong các bài báo của mình, họ đã tìm được những lý do căn bản để giải thích tại sao RNN không thể học được. Vậy nên mô hình LSTM đã được sinh ra để giải quyết vấn đề đó.

1.9.2 Mô hình, ý tưởng và bên trong của mô hình LSTM [7]

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), 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 [3].

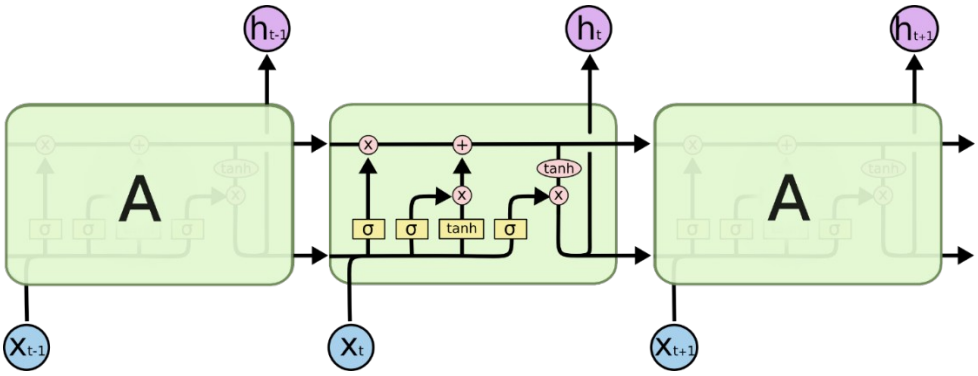
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin xa là đặc tính của nó chứ không cần huấn luyện để làm điều đó.

Mọi mạng tái phát đều có dạng là một chuỗi các module lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các module này có cấu trúc rất đơn giản, thường là một tầng tanh.

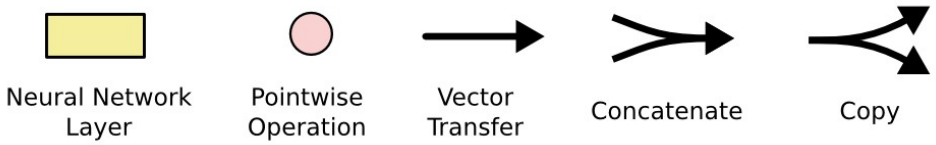


Hình 1.12: Sơ đồ hoạt động mạng LSTM

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



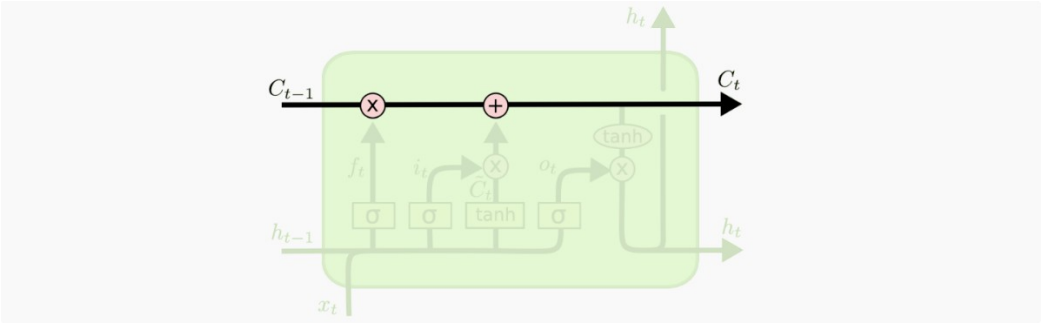
Hình 1.13: Kiến trúc bên trong mạng LSTM



Hình 1.14: Các kí hiệu trong hình vẽ

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.

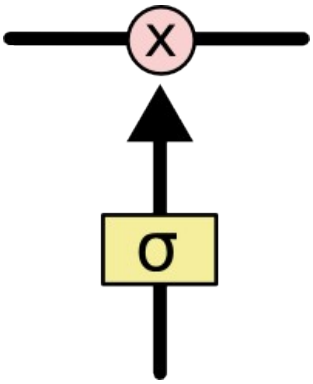
Phần quan trọng nhất của mô hình LSTM là trạng thái tế bào - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ. Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các các nút mạng và chỉ tương tác tuyến tính với nhau một chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.



Hình 1.15: Trạng thái tế bào

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

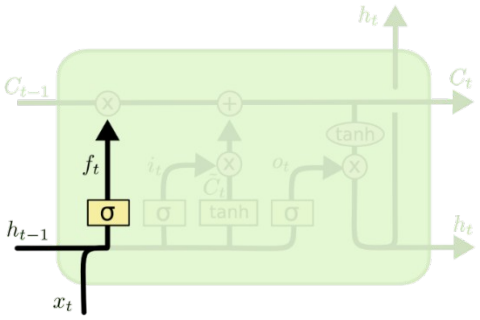
Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân



Hình 1.16: Tầng sigmoid

Tầng sigmoid sẽ cho đầu ra là một số trong khoảng [0, 1], mô tả có bao nhiêu thông tin có thể được thông qua. Với đầu ra của tầng này là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi kết quả 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.

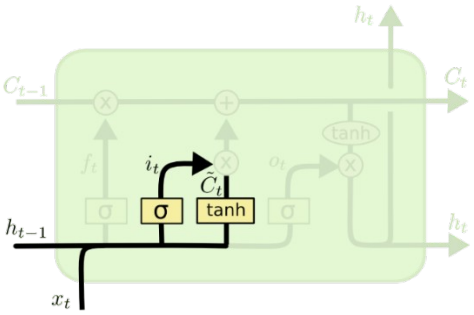
Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là "tầng cổng quên" (forget gate layer). Nó sẽ lấy đầu vào là h_{t-1} và x_t rồi đưa ra kết quả là một số trong khoảng [0, 1] cho mỗi số trong trạng thái tế bào C_{t-1}. 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.



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

Hình 1.17: Mô tả hàm sigmoid

Bước tiếp theo là quyết định xem thông tin mới nào 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 sẽ cập nhật. Tiếp theo là một tầng tanh tạo ra một véc-tơ cho giá trị mới $\tilde{C}_t$ nhằm thêm vào cho trạng thái. Trong bước tiếp theo, sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhật cho trạng thái.

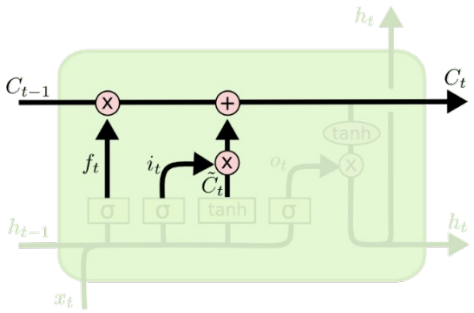


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hình 1.18: Mô tả hàm sigmoid kết hợp với tanh

Giờ là lúc cập nhật trạng thái tế bào cũ C_{t-1} thành trạng thái mới C_t . Ở các bước trước đó đã quyết định những việc cần làm, nên giờ chỉ cần thực hiện là xong.

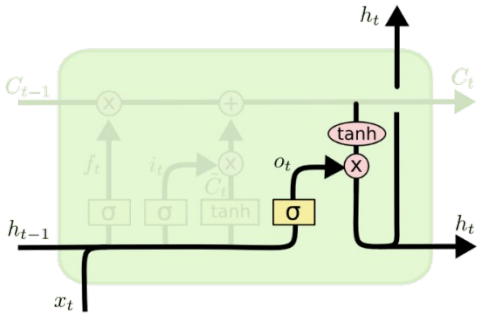
Nhân trạng thái cũ với f_t để bỏ đi những thông tin quyết định quên lúc trước. Sau đó cộng thêm $i_t * \tilde{C}_t$. Trạng thái mới thu được này phụ thuộc vào việc quyết định cập nhập mỗi giá trị trạng thái ra sao.



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

Hình 1.19: Kết hợp để cho ra đầu ra

Cuối cùng, cần quyết định xem 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, chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào muốn xuất ra. Sau đó, 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 mong muốn.



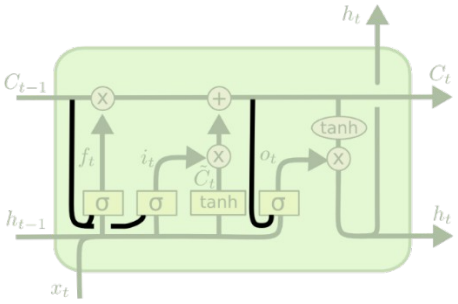
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Hình 1.20: Kết hợp các tầng để cho ra đầu ra là input tiếp theo của dữ liệu tiếp theo

1.9.3 Các biến thể của mô hình LSTM[7]

Những thứ vừa mô tả ở trên là một LSTM khá bình thường. Nhưng không phải tất cả các LTSM đều giống như vậy. Thực tế, các bài báo về LTSM đều sử dụng một phiên bản hơi khác so với mô hình LTSM chuẩn. Sự khác nhau không lớn, nhưng chúng giúp giải quyết phần nào đó trong cấu trúc của LTSM.

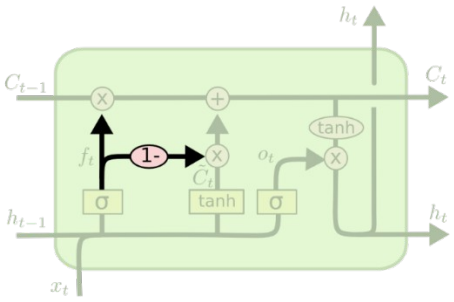
Một dạng LTSM phổ biến được giới thiệu bởi Gers & Schmidhuber (2000) được thêm các đường kết nối "peephole connections", làm cho các tầng cổng nhận được giá trị đầu vào là trạng thái tế bào.



$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

Hình 1.21: Biến thể đầu ra của dữ liệu trước được thêm vào mọi cổng

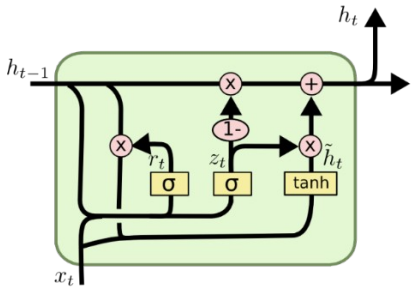
Một biến thể khác là nối 2 cổng loại trừ và đầu vào với nhau. Thay vì phân tách các quyết định thông tin loại trừ và thông tin mới thêm vào, sẽ quyết định chúng cùng với nhau luôn. Chỉ bỏ đi thông tin khi mà thay thế nó bằng thông tin mới đưa vào. Chỉ đưa thông tin mới vào khi bỏ thông tin cũ nào đó đi.



$$C_t = f_t * C_{t-1} + (\mathbf{1} - \boldsymbol{f_t}) * \tilde{C}_t$$

Hình 1.22: Mô hình nối 2 cổng loại trừ đầu vào với nhau

Một biến thể khá thú vị khác của LSTM là Gated Recurrent Unit, hay GRU được giới thiệu bởi Cho, et al. (2014). Nó kết hợp các cổng loại trừ và đầu vào thành một cổng "cổng cập nhật" (update gate). Kết quả là mô hình sẽ đơn giản hơn mô hình LSTM chuẩn và ngày càng trở nên phổ biến.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Hình 1.23: Mô hình GRU

1.10 Các công nghệ sử dụng

1.10.1 Ngôn ngữ lập trình Python

Python, một ngôn ngữ lập trình đa mục đích, đã trở thành một ngôn ngữ phổ biến và ảnh hưởng mạnh mẽ trong cộng đồng lập trình toàn cầu. Ngôn ngữ này được sáng tạo bởi Guido van Rossum và xuất hiện lần đầu vào năm 1991, với mục tiêu tạo ra một ngôn ngữ đơn giản và dễ đọc.

Lịch sử hình thành của Python bắt đầu với phiên bản đầu tiên vào thập kỷ 1990 và từ đó, nó đã trải qua nhiều phiên bản cải tiến. Quá trình phát triển của Python đặc trưng bởi sự linh hoạt và sự mở rộng, với việc thêm vào các tính năng mới và cải thiện hiệu suất của ngôn ngữ.

Python nổi bật với cú pháp rõ ràng, giúp làm giảm độ phức tạp của mã nguồn và tạo điều kiện thuận lợi cho người mới học lập trình. Sự đa nhiệm và hỗ trợ đa nền tảng của Python đã làm cho nó trở thành một công cụ linh hoạt cho phát triển ứng dụng trên nhiều hệ điều hành và môi trường.

Python còn được biết đến với hệ sinh thái phong phú của mình, bao gồm hàng ngàn thư viện và frameworks, giúp lập trình viên tiết kiệm thời gian và công sức khi xây dựng ứng dụng. Sự hỗ trợ cho lập trình hướng đối tượng cũng là một điểm mạnh, giúp tái sử dụng mã nguồn và tạo ra mã nguồn có tổ chức.

Tuy nhiên, như mọi ngôn ngữ lập trình khác, Python cũng có nhược điểm của mình. Một trong những điểm yếu là hiệu suất không cao như các ngôn ngữ biên dịch như C++ hoặc Java, đặc biệt trong các ứng dụng đòi hỏi xử lý số liệu lớn. Mặt khác, sự phổ biến của Python cũng đôi khi dẫn đến việc có quá nhiều thư viện và frameworks, khiến cho việc lựa chọn trở nên khó khăn.

1.10.2 Ngôn ngữ lập trình Kotlin

Kotlin là một ngôn ngữ lập trình đa nền tảng, đã nhanh chóng trở thành một lựa chọn phổ biến trong cộng đồng phần mềm di động và phát triển ứng dụng. Hình thành từ sự cần thiết của một ngôn ngữ thay thế cho Java trên nền tảng Android, Kotlin ra đời bởi JetBrains, một công ty phần mềm có trụ sở tại Nga. Phiên bản chính thức đầu tiên của Kotlin được công bố vào tháng 2 năm 2016.

Lịch sử hình thành của Kotlin liên quan chặt chẽ đến những thách thức và hạn chế của Java trong việc phát triển ứng dụng di động trên Android. Kotlin được thiết kế với mục tiêu giải quyết những vấn đề này, mang lại tính năng hiện đại, mã nguồn ngắn gọn và khả năng tương thích hoàn hảo với mã nguồn Java sẵn có.

Quá trình phát triển của Kotlin nhanh chóng và tích cực, với sự hỗ trợ của cộng đồng lập trình viên ngày càng mở rộng. Google chính thức công nhận Kotlin là ngôn ngữ chính thức trên Android vào năm 2017, mở đầu cho sự lan rộng của Kotlin trong cả phát triển ứng dụng di động và máy chủ.

Ưu điểm lớn nhất của Kotlin là sự tương thích với Java, cho phép dễ dàng tích hợp và chuyển đổi mã nguồn giữa hai ngôn ngữ. Kotlin hỗ trợ các tính năng

hiện đại như lambda expressions, extension functions, nullable types, và smart casts, giúp tăng cường sức mạnh của ngôn ngữ.

Tuy nhiên, có một số nhược điểm cần xem xét, bao gồm sự mới mẻ của ngôn ngữ này so với Java, có thể gây khó khăn trong việc tìm nguồn tư vấn và tài liệu. Mặc dù Kotlin có sự hỗ trợ tốt trên Android, nhưng nó vẫn chưa phổ biến rộng rãi trong mọi lĩnh vực phát triển ứng dụng.

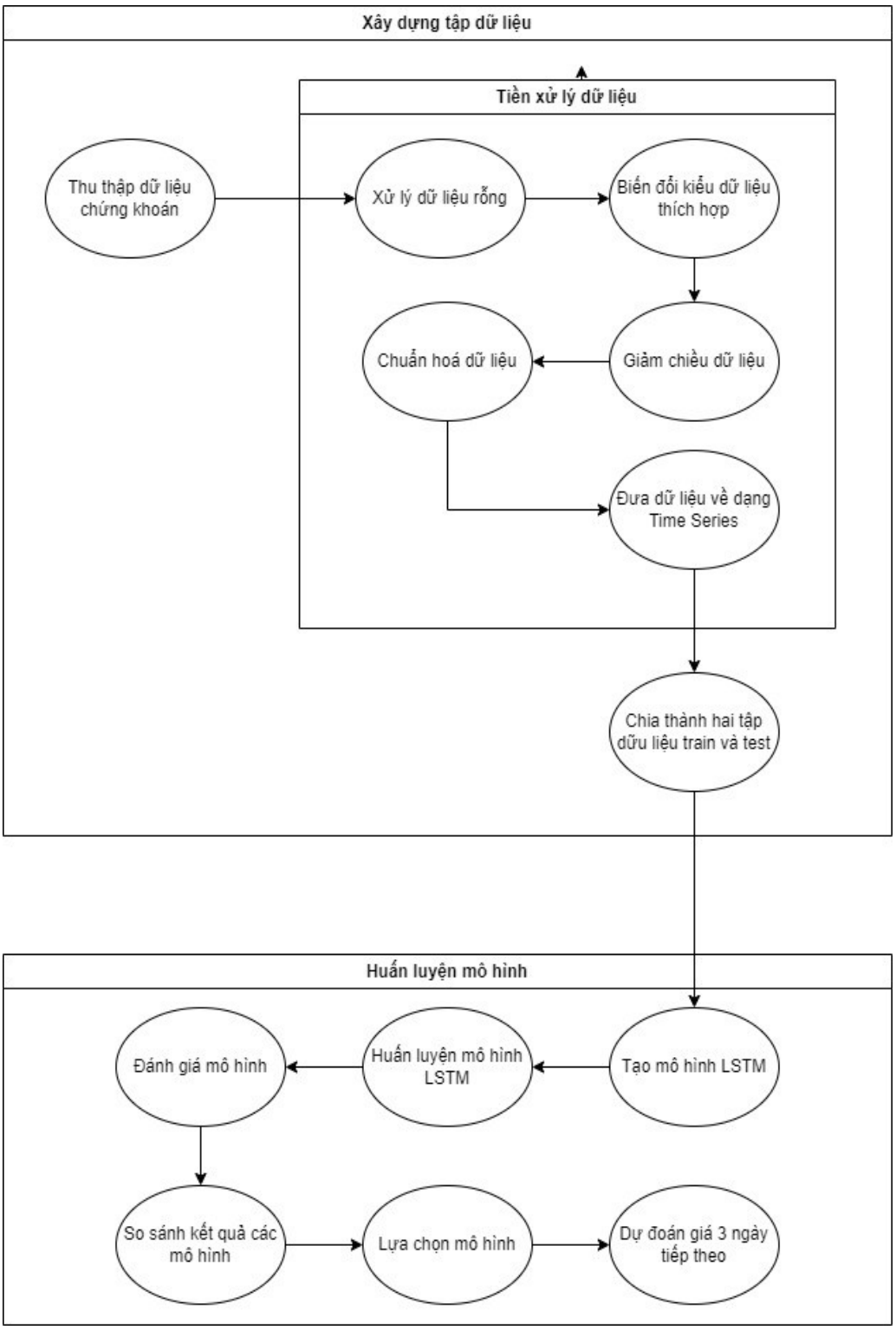
Tóm lại, Kotlin không chỉ là một ngôn ngữ lập trình mới mẻ mà còn là một sự tiến bộ trong việc phát triển ứng dụng di động và đa nền tảng. Với sự hỗ trợ của JetBrains và cộng đồng người dùng ngày càng tích cực, Kotlin đang trở thành một lựa chọn hấp dẫn cho các nhà phát triển mong muốn sự hiệu quả và linh hoạt trong việc xây dựng ứng dụng chất lượng cao.

1.11 Kết luận chương

Trong chương 1 đã trình bày về những tìm hiểu về bài toán dự đoán giá chứng khoán, những cơ sở lý thuyết để thực hiện bài toán. Dựa vào những thông tin đã được tìm hiểu ở chương này, chương tiếp theo của đồ án sẽ thực hiện việc lấy dữ liệu và xây dựng mô hình dự đoán cho hệ thống.

CHƯƠNG 2. LẤY DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH

Trong chương này sẽ trình bày về quá trình lấy dữ liệu và xây dựng mô hình dự đoán giá chứng khoán sử dụng mô hình LSTM, bao gồm quá trình thu thập dữ liệu, xử lý dữ liệu, xây dựng mô hình, huấn luyện và chọn ra mô hình và cuối cùng là dự đoán giá tương lai.



Hình 2.24 Sơ đồ hệ thống lấy dữ liệu, xử lý dữ liệu và huấn luyện mô hình

2.1 Thu thập dữ liệu

Để có thể huấn luyện, xây dựng và đánh giá mô hình cùng với so sánh dữ liệu dự đoán với dữ liệu thực tế thì cần bộ dữ liệu đủ lớn để có thể làm điều đó.

2.1.1 Crawl dữ liệu là gì và dữ liệu chứng khoán được crawl như thế nào

Crawl là một thuật ngữ mô tả quá trình thu thập dữ liệu trên Website của các con bot công cụ tìm kiếm. Hành động này được ví như là bò trườn vì trong quá trình thu thập dữ liệu của mình, các con bot sẽ lần lượt truy cập vào từng liên kết trên trang mà nó bắt gặp, và tiếp tục thu thập dữ liệu ở các liên kết mới đó.

Đề tài này sẽ sử dụng bộ dữ liệu về giá các mã chứng khoán và các mã phái sinh của Việt Nam với thời gian được lấy tối đa là 10 năm. Bộ dữ liệu được lấy bằng cách crawl data từ trang web cophieu68.vn.

Các bước để lấy giá cổ phiếu :

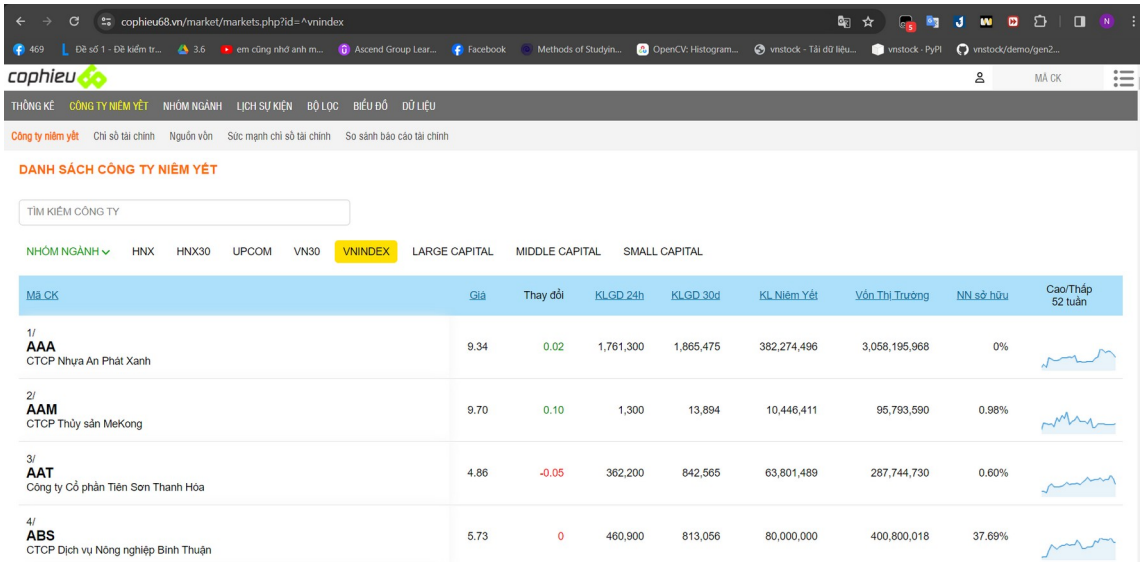
- Lấy danh sách các mã chứng khoán. Ở trong phạm vi đề tài này sẽ lấy các mã chứng khoán của sàn VNINDEX
- Lấy thông tin của từng mã chứng khoán
- Lấy giá chứng khoán của từng mã
- Lưu giá chứng khoán vào cơ sở dữ liệu

Phương pháp crawl chung của việc lấy danh sách các mã chứng khoán và giá của từng mã là sẽ sử dụng request để lấy source code HTML của trang đó. Sau đó xử lý source code ở dạng HTML để lấy ra thẻ <table> do danh sách và giá các mã chứng khoán ở trang web đó đều để trong thẻ <table>. Sau khi xử lý source code xong sẽ lấy ra các trường và lưu dữ liệu lại.

- Lấy danh sách các mã chứng khoán

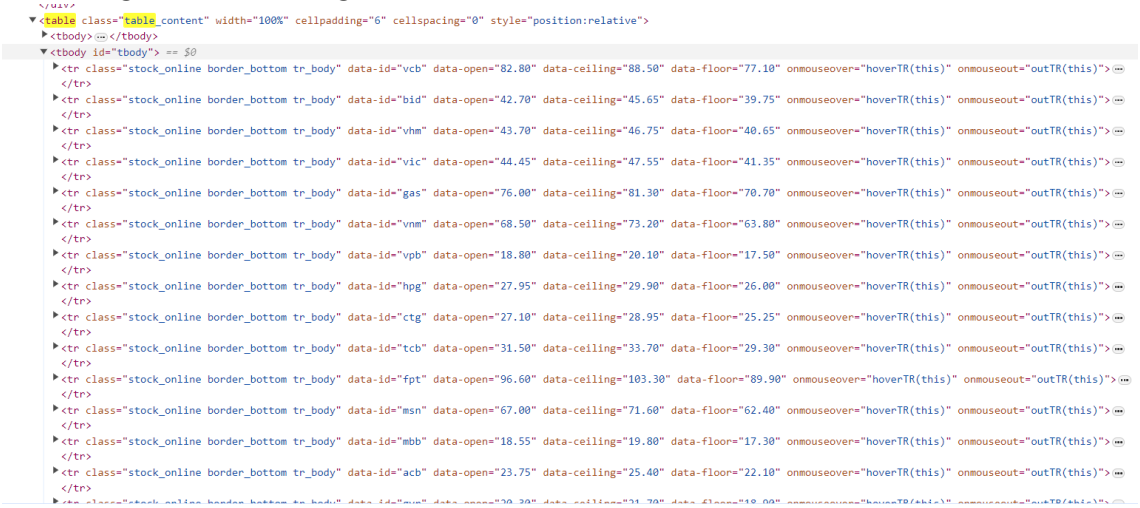
Danh sách các mã chứng khoán của sàn VNINDEX được lấy từ trang https://www.cophieu68.vn/market/markets.php?id=^vnindex với :

id=^vnindex : Hiển thị các mã cổ phiếu của sàn VNINDEX



Hình 2.25: Dữ liệu trên trang web cophieu68.vn

Đối với danh sách các mã chứng khoán sẽ chỉ lấy trường "MÃ CK" tương ứng với trường "data-id" trong source code.



Hình 2.26 Danh sách các mã chứng khoán

Chi tiết mỗi mã sẽ lấy tên của mã chứng khoán đó. Ví dụ như trong hình là "CTCP Nhựa An Phát Xanh"

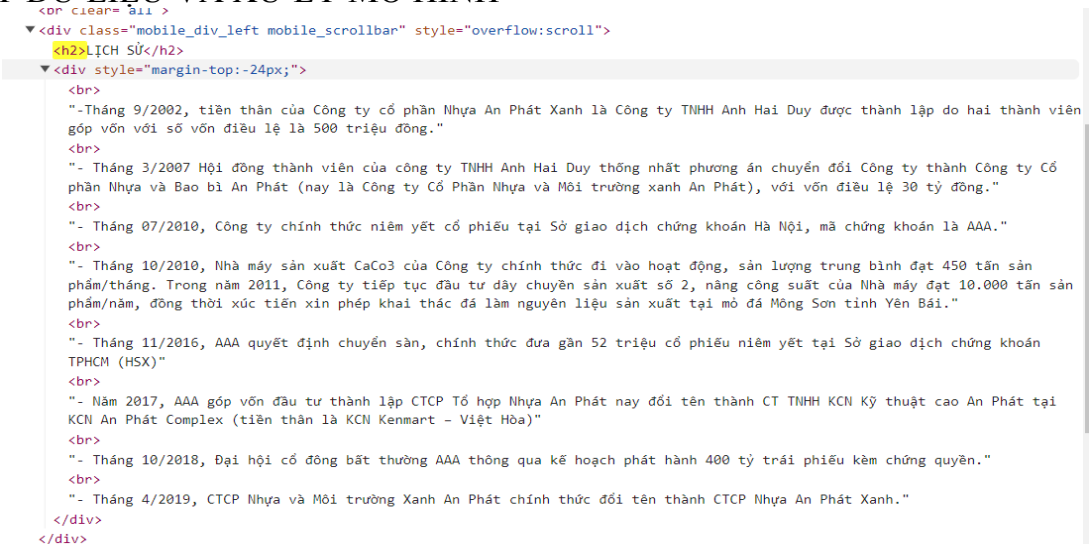
Hình 2.27 Thông tin mã chứng khoán

- Lấy thông tin của từng mã chứng khoán

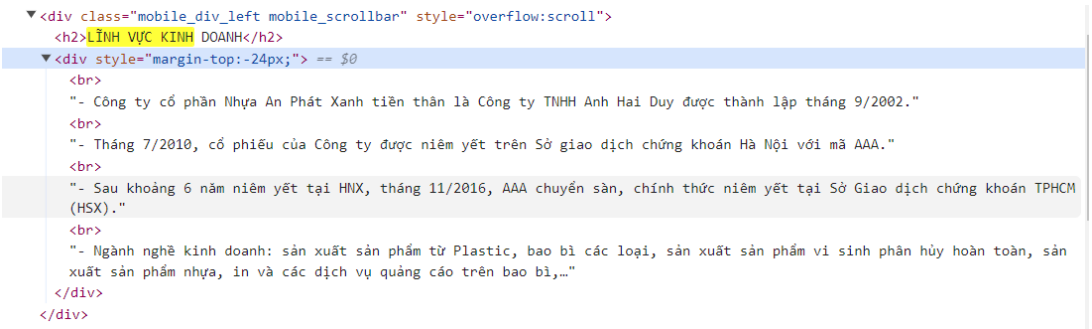
  Thông tin của từng mã chứng khoán sẽ được lấy tại trang có cấu trúc : https://www.cophieu68.vn/quote/profile.php?id=aaa

  Với id=aaa là thông tin của mã cổ phiếu có mã là AAA

  Thông tin về lịch sử cũng như lĩnh vực kinh doanh được lấy trong thẻ <h2> với nội dung là "LỊCH SỬ" và thẻ <h2> với nội dung là "LĨNH VỰC KINH DOANH".



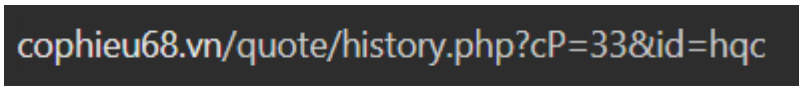
Hình 2.28: Thông tin về lịch sử mã chứng khoán



Hình 2.29: Thông tin về lịch sử kinh doanh của mã chứng khoán

- Lấy giá chứng khoán của từng mã

Do mỗi trang web hiển thị giá của mỗi mã cổ phiếu có một cấu trúc chung như sau :



Hình 2.30: Đường dẫn mẫu crawl

Ý nghĩa của đường dẫn :

id=hqc : là trang web giá của cổ phiếu có mã là hqc

P = 33 : là trang thứ 33 của danh sách giá cổ phiếu của mã ở trên

Dữ liệu sau khi crawl của mỗi mã ở dạng HTML sẽ nằm trong thẻ <table> nên sẽ lấy nó ra ở dạng bảng và bao gồm các trường:

- Date : Ngày
- Open : Giá mở cửa
- High : Giá cao nhất
- Low : Giá thấp nhất
- Close : Giá đóng cửa
- Volume : Khối lượng giao dịch

2.1.2 Xử lý dữ liệu

Xử lý dữ liệu là quá trình thao tác và biến đổi dữ liệu từ các nguồn khác nhau thành dạng có ý nghĩa và sẵn sàng cho việc phân tích, đánh giá, hoặc ứng dụng trong các ứng dụng khác nhau. Quá trình này bao gồm một loạt các bước nhằm đảm bảo dữ liệu lành mạnh, chính xác và có thể sử dụng hiệu quả.

Quy trình xử lý dữ liệu đặt ra mục tiêu tối ưu hóa chất lượng của dữ liệu để đảm bảo rằng nó phản ánh đúng thực tế và đáp ứng được các yêu cầu cụ thể của dự án hoặc ứng dụng. Thông qua việc loại bỏ nhiễu, điều chỉnh định dạng, và tạo ra một tập dữ liệu đồng nhất, quá trình xử lý dữ liệu giúp tạo ra một nền tảng cơ bản cho các phân tích sau này, từ phân tích thống kê đến việc xây dựng và huấn luyện mô hình máy học. Điều này giúp nâng cao khả năng hiểu biết và đưa ra quyết định chính xác trong môi trường phức tạp của dữ liệu hiện đại.

Với lượng dữ liệu lớn theo thời gian của các mã chứng khoán như trong đề tài, cần xử lý chúng để tối ưu hiệu năng cũng như bộ nhớ của hệ thống từ đó giúp tăng năng suất công việc và giảm chi phí cho việc đầu tư vào hệ thống. Một số hoạt động xử lý dữ liệu và cách xử lý dữ liệu trong đề tài là :

- Xử lý dữ liệu rỗng : Các bản ghi dữ liệu với các trường có giá trị rỗng hoặc không được định nghĩa sẽ được loại bỏ khỏi tập dữ liệu
- Biến đổi dữ liệu về dạng thích hợp: Đối với tập dữ liệu thay đổi theo thời gian như trong đề tài thì chỉ mục của dữ liệu sẽ không phải là số thứ tự của bản ghi mà sẽ là ngày tháng của bản ghi đó. Kiểu dữ liệu cũng sẽ được chuẩn hoá, các trường dữ liệu về thông tin giá các mã chứng khoán sẽ được đưa về kiểu số thực còn số lượng giao dịch sẽ được đưa về kiểu số nguyên.
- Giảm chiều dữ liệu: Trong nguồn data được trả về sau khi được xử lý từ HTML request có rất nhiều trường dữ liệu nên việc chuẩn hoá giúp lấy ra những trường cần thiết như: Ngày mở cửa, giá mở cửa, giá cao nhất trong ngày, giá thấp nhất trong ngày, giá đóng cửa, khối lượng giao dịch.
- Chuẩn hoá dữ liệu: Dữ liệu trước khi được đưa vào trong mô hình huấn luyện cần được chuẩn hoá nhằm biến đổi các giá trị của mỗi trường về cùng một tỷ lệ, giúp mô hình học máy hội tụ nhanh hơn và làm tăng hiệu suất của mô hình. Trong đề tài sẽ sử dụng phương pháp chuẩn hoá dữ liệu Min-Max Scaling đưa dữ liệu về khoảng từ [0-1] theo công thức:

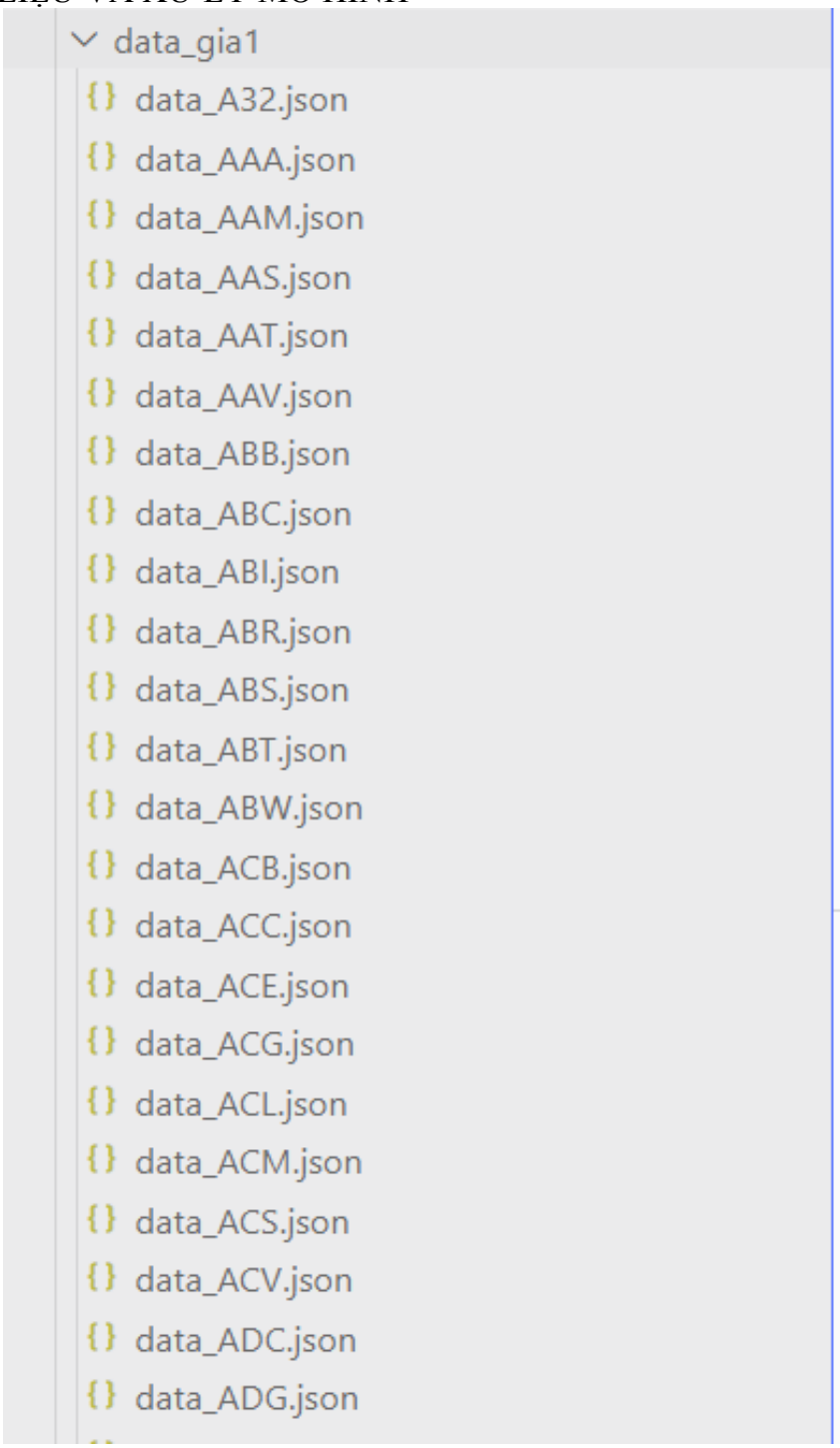
$$X_{normalized} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Hình 2.31: Công thức Min-Max Scaling

Trong đó, X là giá trị ban đầu của biến số.

2.1.3 Lưu trữ dữ liệu

Dữ liệu sau khi xử lý sẽ được lưu trữ dưới dạng file JSON trong thư mục :



Hình 2.32 Dữ liệu thông tin giá từng mã chứng khoán

Dữ liệu thông tin của từng mã chứng khoán sẽ được lưu trữ ở ở file json theo cấu trúc tên ”daa_<mã chứng khoán>.json”. Cách lưu trữ như vậy sẽ giúp cho việc đọc file cũng như lưu trữ file để sau này cập nhật giá sẽ thuận tiện hơn. Các app cần sử dụng file này sẽ có cấu trúc chung để đọc và biết thông tin về giá đang đọc và ghi là của mã chứng khoán nào.

[{"Date":"2012-03-20","Open":4230.0,"High":4230.0,"Low":4138.0,"Close":4184.0,"Volume":302500,"ticker":"AAA","Difference":-46.0},
{"Date":"2012-03-21","Open":4207.0,"High":4461.0,"Low":4207.0,"Close":4392.0,"Volume":826800,"ticker":"AAA","Difference":185.0},
{"Date":"2012-03-22","Open":4392.0,"High":4693.0,"Low":4392.0,"Close":4623.0,"Volume":672400,"ticker":"AAA","Difference":231.0},
{"Date":"2012-03-23","Open":4762.0,"High":4947.0,"Low":4762.0,"Close":4924.0,"Volume":473300,"ticker":"AAA","Difference":162.0},
{"Date":"2012-03-26","Open":5224.0,"High":5247.0,"Low":5039.0,"Close":5247.0,"Volume":394300,"ticker":"AAA","Difference":23.0},
{"Date":"2012-03-27","Open":5247.0,"High":5247.0,"Low":4901.0,"Close":4901.0,"Volume":551000,"ticker":"AAA","Difference":-346.0},
{"Date":"2012-03-28","Open":4693.0,"High":4970.0,"Low":4577.0,"Close":4854.0,"Volume":551900,"ticker":"AAA","Difference":161.0},
{"Date":"2012-03-29","Open":4970.0,"High":4970.0,"Low":4531.0,"Close":4531.0,"Volume":556300,"ticker":"AAA","Difference":-439.0},
{"Date":"2012-03-30","Open":4508.0,"High":4623.0,"Low":4300.0,"Close":4461.0,"Volume":234900,"ticker":"AAA","Difference":-47.0},
{"Date":"2012-04-03","Open":4461.0,"High":4762.0,"Low":4461.0,"Close":4762.0,"Volume":134000,"ticker":"AAA","Difference":301.0},
{"Date":"2012-04-04","Open":4831.0,"High":4901.0,"Low":4508.0,"Close":4716.0,"Volume":197600,"ticker":"AAA","Difference":-115.0},
{"Date":"2012-04-05","Open":4623.0,"High":4854.0,"Low":4508.0,"Close":4739.0,"Volume":199800,"ticker":"AAA","Difference":116.0},
{"Date":"2012-04-06","Open":4808.0,"High":4924.0,"Low":4739.0,"Close":4785.0,"Volume":221600,"ticker":"AAA","Difference":-23.0},
{"Date":"2012-04-09","Open":4808.0,"High":5039.0,"Low":4646.0,"Close":4970.0,"Volume":309100,"ticker":"AAA","Difference":162.0},
{"Date":"2012-04-10","Open":4947.0,"High":4970.0,"Low":4739.0,"Close":4785.0,"Volume":333500,"ticker":"AAA","Difference":-162.0},
{"Date":"2012-04-11","Open":5016.0,"High":5062.0,"Low":4854.0,"Close":4993.0,"Volume":312900,"ticker":"AAA","Difference":-23.0},
{"Date":"2012-04-12","Open":4924.0,"High":5155.0,"Low":4901.0,"Close":4970.0,"Volume":272600,"ticker":"AAA","Difference":46.0},
{"Date":"2012-04-13","Open":4924.0,"High":4970.0,"Low":4808.0,"Close":4877.0,"Volume":356200,"ticker":"AAA","Difference":-47.0},
{"Date":"2012-04-16","Open":4901.0,"High":5155.0,"Low":4831.0,"Close":5039.0,"Volume":293100,"ticker":"AAA","Difference":138.0},
{"Date":"2012-04-17","Open":5132.0,"High":5155.0,"Low":4970.0,"Close":5039.0,"Volume":425600,"ticker":"AAA","Difference":-93.0},
{"Date":"2012-04-18","Open":5039.0,"High":5109.0,"Low":4924.0,"Close":4970.0,"Volume":325700,"ticker":"AAA","Difference":-69.0},
{"Date":"2012-04-19","Open":4924.0,"High":4947.0,"Low":4669.0,"Close":4739.0,"Volume":310900,"ticker":"AAA","Difference":-185.0},
{"Date":"2012-04-20","Open":4785.0,"High":4854.0,"Low":4669.0,"Close":4762.0,"Volume":118300,"ticker":"AAA","Difference":-23.0},
{"Date":"2012-04-23","Open":4831.0,"High":4901.0,"Low":4739.0,"Close":4762.0,"Volume":56700,"ticker":"AAA","Difference":-69.0},
{"Date":"2012-04-24","Open":4831.0,"High":4854.0,"Low":4669.0,"Close":4831.0,"Volume":144500,"ticker":"AAA","Difference":0.0},
{"Date":"2012-04-25","Open":4947.0,"High":5086.0,"Low":4924.0,"Close":4993.0,"Volume":292500,"ticker":"AAA","Difference":46.0},

Hình 2.33 Thông tin giá của một mã chứng khoán

,ticker,thong tin
747,A32,"<DIV style=""FONT-FAMILY: Arial; FONT-SIZE: 10pt;"">Công ty Cổ phần 32 (A32) có tiền thân là Xí nghiệp X32 - Bộ Quốc Phò
497,AAA," Công ty Cổ phần Nhựa An Phát Xanh (AAA) có tiền thân là Công ty TNHH Anh Hai Duy được thành lập vào năm 2002. Công ty h
584,AAM," Công ty Cổ Phần Thủy Sản Mê Kông (AAM) có tiền thân là Xí nghiệp Rau quả đông lạnh xuất khẩu Hậu Giang được thành lập v
70,AAS,"Công ty Cổ phần Chứng khoán SmartInvest (AAS) có tiền thân là Công ty Cổ phần Chứng khoán Gia Anh được thành lập vào năm
744,AAT,"Công ty Cổ phần Tập đoàn Tiên Sơn Thanh Hóa (AAT), tiền thân là Công ty Tiên Sơn Thanh Hoá - TNHH được thành lập năm 199
101,AAV,"Công ty Cổ phần AAV Group (AAV) được thành lập vào năm 2010. Công ty hoạt động trong lĩnh vực đầu tư xây dựng và kinh do
372,ABB,"Ngân hàng Thương mại Cổ phần An Bình (ABB) có tiền thân là Ngân hàng Thương Mại Cổ phần Nông thôn An Bình được thành lập
31,ABC,"Được thành lập vào năm 2006, Công ty Cổ phần truyền thông VMG (ABC), có trụ sở tại Quận Đống Đa, Hà Nội, hoạt động trong
217,ABI,"Công ty Cổ phần Bảo hiểm Ngân hàng Nông Nghiệp (ABIC) thành lập ngày 18/10/2006 và đi vào hoạt động ngày 08/08/2007 với
1481,ABR,"Công ty Cổ phần Đầu tư Nhãn Hiệu Việt (ABR) có tiền thân là Công ty Cổ phần Chế biến Gỗ Kiến An được thành lập vào năm
897,ABS,"Công ty Cổ phần Dịch vụ Nông nghiệp Bình Thuận (ABS) có tiền thân là Công ty Vật tư Nông nghiệp Thuận Hải được thành lập
585,ABT," Công ty Cổ phần Xuất nhập khẩu Thủy sản Bến Tre (ABT) tiền thân là Xí Nghiệp Đông Lạnh thành lập năm 1977. Năm 2004 chu
55,ABW,"Công ty Cổ phần Chứng khoán An Bình (ABW) chính thức hoạt động từ ngày 5/11/2006 với vốn điều lệ ban đầu 50 tỷ đồng. Các
8,ACB," Ngân hàng Thương mại Cổ phần Á Châu (ACB) được thành lập năm 1993. Ngân hàng hoạt động trong lĩnh vực huy động, kinh doan
1024,ACC,"Công ty Cổ phần Đầu tư và Xây dựng Bình Dương ACC (ACC) có tiền thân là Xí nghiệp bê tông cốt thép và bê tông nhựa nóng
1025,ACE,"Công ty Cổ phần Bê Tông Ly Tâm An Giang (ACE) tiền thân là Xí nghiệp Bê Tông Ly Tâm An Giang trực thuộc Công ty Xây lắp
386,ACG,"Công ty Cổ phần Gỗ An Cường (ACG), tiền thân là Công ty TNHH Thương mại An Cường, được thành lập năm 1994. ACG là nhà sả
586,ACL," Công ty Cổ phần Xuất nhập khẩu Thủy sản Cửu Long An Giang (ACL) được thành lập năm 2003. Năm 2007, công ty chuyển đổi s
451,ACM,"Công ty Cổ phần Tập đoàn Khoáng Sản Á Cường (ACM) tiền thân là Công ty TNHH Tam Cường, được thành lập năm 1966. Lĩnh vực
1026,ACS,"Công ty Cổ phần Xây lắp Thương mại 2 (ACS) có tiền thân là Công ty Xây lắp Miền Nam, được thành lập vào năm 1976. Công
1370,ACV," Tổng Công ty Cảng Hàng không Việt Nam (ACV) được thành lập vào năm 2012 trên cơ sở hợp nhất Tổng Công ty Cảng Hàng khô
804,ADC," Công ty Cổ phần Mĩ thuật và Truyền thông (ADC), được thành lập vào năm 2007. Công ty hoạt động trong lĩnh vực thiết kế,
809,ADG,"Công ty Cổ phần Clever Group (ADG) có tiền thân là Công ty Cổ phần Quảng Cáo Thông minh được thành lập vào năm 2008. Côn
1027,ADP," Công ty Cổ phần Sơn Á Đông (ADP) được thành lập năm 1970 là một trong hai nhà sản xuất lớn nhất và chi phối thị trường
755,ADS,"Công ty Cổ phần Damsan (ADS) có tiền thân là Công ty Cổ phần Dệt sợi Damsan được thành lập vào năm 2006. Công ty hoạt đô

Hình 2.34 Dữ liệu thông tin các mã chứng khoán

Ở trên là thông tin chi tiết về giá của một mã chứng khoán và các thông tin về mã chứng khoán để bổ sung vào phần thông tin chi tiết chứng khoán.

2.2 Mô hình LSTM được xây dựng

Trước khi xây dựng mô hình LSTM cần xác định được mỗi cặp key value của một phần tử trong Time Series làm input của mô hình [5]. Trong đề tài này mỗi một phần tử sẽ có một giá trị là giá đóng cửa của ngày hôm đó, còn số ngày lấy ra để huấn luyện để dự đoán giá cho ngày hôm đó thì là giá của 50 ngày trước đó.

```
def trainmodel(listxy,ticker) :
    x_train,y_train = listxy[0], listxy[1]
    #xay mo hinh

    model = Sequential()
    model.add(LSTM(units = 128,input_shape = (x_train.shape[1],1),return_sequences=True))
    model.add(LSTM(units = 64))
    model.add(Dropout(0.5))
    model.add(Dense(1))
    model.compile(loss ='mean_absolute_error',optimizer = 'adam')
```

Hình 2.35 Xây dựng mô hình

Xây dựng mô hình LSTM gồm nhiều lớp bao gồm 1 lớp đầu vào, 2 lớp ẩn và một lớp đầu ra. Mỗi lớp với khối của đầu vào được khởi tạo với kích thước là (50,1) tương ứng với giá của 50 ngày trước sẽ cho ra giá của ngày hiện tại. Lớp ẩn thứ nhất sẽ có kích thước khối đầu vào là kích thước của khối đầu vào. Kết quả của lớp thứ nhất sẽ được sử dụng tiếp cho lớp thứ 2. Mô hình cần phải bỏ qua một số unit để tránh cho việc model sẽ học tủ, làm giảm hiệu suất của mô hình với lớp quên và chỉ giữ lại 50%. Lớp Dense để cho ra 1 kết quả sau khi mạng nơ ron kết nối với nhau. Hàm mất mát được sử dụng để đo lường sự chênh lệch giữa giá trị dự đoán và giá trị thực tế. Trong trường hợp này, đây là mean absolute error. Thuật toán tối ưu hoá được sử dụng trong mô hình này là Adam.

2.3 Chia tập dữ liệu, huấn luyện mô hình và đánh giá mô hình

Tập dữ liệu đầu vào được chia làm 2 phần train và test, dữ liệu train được lấy từ một nửa của dữ liệu đầu vào và dữ liệu test là một nửa còn lại. Do đề tài làm về nhiều mã chứng khoán nên mỗi một mã chứng khoán sẽ là một mô hình khác nhau. Nên các mô hình sau khi huấn luyện sẽ được lưu lại vào một folder riêng để có thể sử dụng dự đoán cho lần sau nếu còn sử dụng được.

2.4 Đánh giá mô hình dự đoán

Sau khi xây dựng mô hình xong thì dữ liệu sẽ được cho vào train và sẽ lấy ra model có độ chính xác cao nhất. Model sẽ được đánh giá theo Mean Squared Error. Mean Squared Error được gọi là giá trị sai số bình phương trung bình hoặc là lỗi bình phương trung bình. Vấn đề khi nói về sai số trung bình của một mô hình thống kê nhất định là rất khó xác định mức độ lỗi là do mô hình và mức độ là do ngẫu nhiên. Lỗi bình phương trung bình (MSE) cung cấp một thống kê cho phép các nhà nghiên cứu đưa ra tuyên bố như vậy. MSE chỉ đơn giản đề cập đến giá trị trung bình của chênh lệch bình phương giữa tham số dự đoán và tham số quan sát được. Công thức tính :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Trong đó : Yi là giá trị gốc

Ŷi là giá trị ước lượng

Công thức tính sai số MSE sau đó sẽ được lấy căn bậc 2 để ra giá trị Root Mean Square Error(RMSE). Ngoài MSE, RMSE thì để đánh giá độ sai số so với giá trị thực có thể sử dụng chỉ số Mean Absolute Error (MAE) và Mean Absolute Percentage Error(MAPE). Hai công thức có cách tính như sau:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i |$$

Trong đó : Yi là giá trị gốc

Ŷi là giá trị ước lượng

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

Trong đó : Yi là giá trị gốc

Ŷi là giá trị ước lượng

Dưới đây là bảng đánh giá mô hình dự đoán ở trên tương ứng với một số mã chứng khoán:

Mã chứng khoán	RMSE	MAE	MAPE(%)
AAS	792.11	609.19	5.94
QBS	174.24	117.753	3.48
AAT	343.0	253.86	4.27
PLC	1205.78	803.03	3.19
FPT	2054.21	1402.20	2.78
DQC	898.27	532.50	2.36
FIT	354.34	246.20	3.424
HAG	338.30	255.31	4.03
DIG	7440.74	2896.19	7.04

Bảng 2.1 Đánh giá mô hình dự đoán

Trong bảng đánh giá mô hình dự đoán được thống kê ở trên có các chỉ số RMSE, MAE, MAPE với MAPE là tỉ lệ sai số cao nhất là 7.04% và thấp nhất là 2.36%. Đây là một tỉ lệ tương đối tốt của một mô hình dự đoán với xác suất dự đoán sai nhỏ hơn 10% và chủ yếu nằm trong khoảng từ 2-5%.

Nhằm đánh giá rõ hơn giữa giá dự đoán và giá thực tế thì dữ liệu đã được trực quan hoá theo các biểu đồ dưới đây:

Hình 2.36 So sánh giá dự đoán và giá thực tế AAS

Hình 2.37 So sánh giá dự đoán và giá thực tế QBS

Hình 2.38 So sánh giá dự đoán và giá thực tế AAT

Ba hình ảnh thể hiện giá dự đoán và giá thực tế của ba mã chứng khoán AAS, QBS và AAT. Qua biểu đồ của chúng có thể thấy giá dự đoán tương đối sát với giá thực tế. Giá dự đoán luôn có dấu hiệu tăng giảm đi trước một chút so với giá thực tế để giúp người dùng có thể biết thời điểm thích hợp để đầu tư.

Hình 2.39 So sánh giá dự đoán và giá thực tế PLC

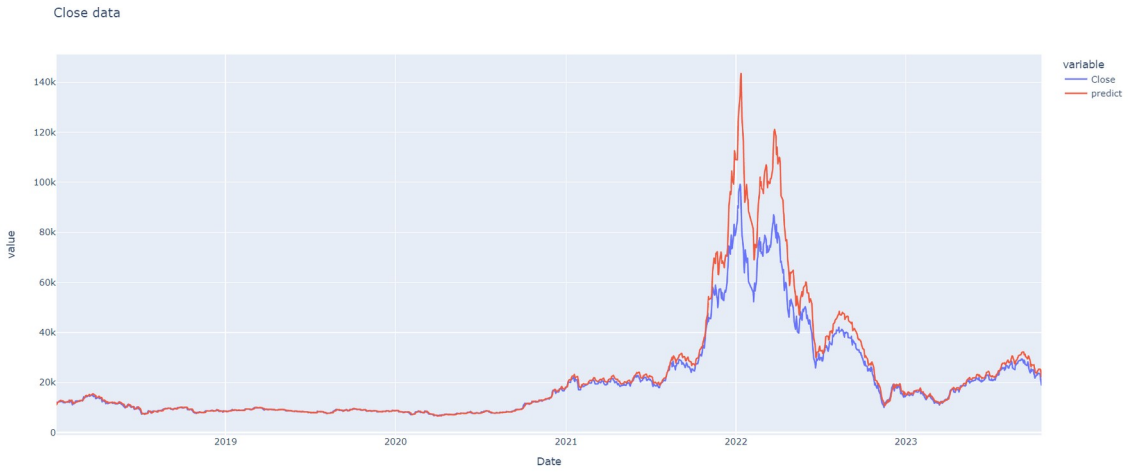
Hình 2.40 So sánh giá dự đoán và giá thực tế FPT

Hình 2.41 So sánh giá dự đoán và giá thực tế DQC

Tương tự như biểu đồ của ba mã đầu tiên thì biểu đồ giá dự đoán và giá đóng cửa thực tế của ba mã chứng khoán PLC, FPT và DQC cũng sát với nhau. Đoạn giá về sau của FPT mặc dù về xu hướng tăng giảm tương đối chính xác nhưng hai đường có vẻ cách xa nhau.

Hình 2.42 So sánh giá dự đoán và giá thực tế FIT

Hình 2.43 So sánh giá dự đoán và giá thực tế HAG



Hình 2.44 So sánh giá dự đoán và giá thực tế DIG

Cuối cùng là biểu đồ của ba mã chứng khoán FIT,HAG và DIG. Hai biểu đồ này cũng thể hiện việc giá thực tế và giá dự đoán khá là khớp với nhau, xu hướng tăng giảm cũng tương đối chính xác.

Về việc giá tăng giảm giống nhau nhưng về mức tăng giảm thì có chênh lệch là do giá dự đoán chỉ mang tính chất tham khảo và được huấn luyện dựa vào giá đóng cửa của các ngày trước đó. Tuy nhiên giá cả còn phụ thuộc vào các sự kiện có trong năm của công ty niêm yết. Có thể là các báo cáo kết quả hoạt động, thay đổi cổ đông, thay đổi cổ phần,... Những yếu tố đó cũng làm ảnh hưởng đến giá cả đột ngột.

Tuy nhiên việc tăng giảm giá của các mã chứng khoán đã có sự khớp giữa dự đoán và giá thực tế cũng đã đủ để giúp người dùng có thể đầu tư ngắn hạn và nắm bắt được nên mua vào và bán ra khi nào là có lợi nhất cho người dùng.

2.5 Kết luận chương

Trong chương này đã tìm hiểu về việc lấy dữ liệu, xây dựng mô hình, huấn luyện mô hình cũng như đánh giá mô hình để tối ưu việc lựa chọn mô hình cho hệ thống nhằm tăng khả năng dự đoán kết quả cho chính xác.

Sau khi có dữ liệu và xử lý dữ liệu xong thì chương tiếp theo sẽ trình bày về việc phân tích và thiết kế hệ thống sau đó cài đặt hệ thống.

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG

Trong chương này của đồ án sẽ trình bày về phân tích thiết kế cho hệ thống, cho từng module có mặt trong hệ thống và cài đặt hệ thống.

3.1 Phân tích hệ thống

3.1.1 Tên hệ thống

Hệ thống dự đoán chứng khoán Việt Nam

3.1.2 Mục tiêu của hệ thống

Hệ thống giúp cho người dùng theo dõi được thông tin của các mã chứng khoán bao gồm : Thông tin của công ty niêm yết, thông tin giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa, khối lượng giao dịch, mức tăng giảm giá đóng cửa giữa hai ngày liên tiếp. Bên cạnh đó sẽ trực quan hoá giá của các mã chứng khoán dưới dạng biểu đồ và hiển thị giá dự đoán trong tương lai ở trong biểu đồ. Từ các thông tin đó giúp người dùng nắm bắt được các thông tin của các mã chứng khoán nhằm hỗ trợ đầu tư chứng khoán ở hiện tại và tương lai.

3.1.3 Các tác nhân của hệ thống

Hệ thống bao gồm các tác nhân chính sau :

- Người dùng : Người muốn theo dõi thông tin và đầu tư chứng khoán

3.1.4 Yêu cầu của hệ thống

Hệ thống có khả năng lấy dữ liệu bao gồm cả dữ liệu dự đoán được cập nhật hàng ngày theo thời gian thực sau đó hiển thị lên phần mềm. Hệ thống có thể cho phép người dùng chọn mã cổ phiếu muốn xem và tìm kiếm mã trong ô tìm kiếm. Đối với biểu đồ trực quan hoá, người dùng có thể trỏ vào từng vị trí để xem thông tin chi tiết của một mã chứng khoán theo ngày được trỏ tới.

3.1.5 Phạm vi của hệ thống

Người dùng có thể :

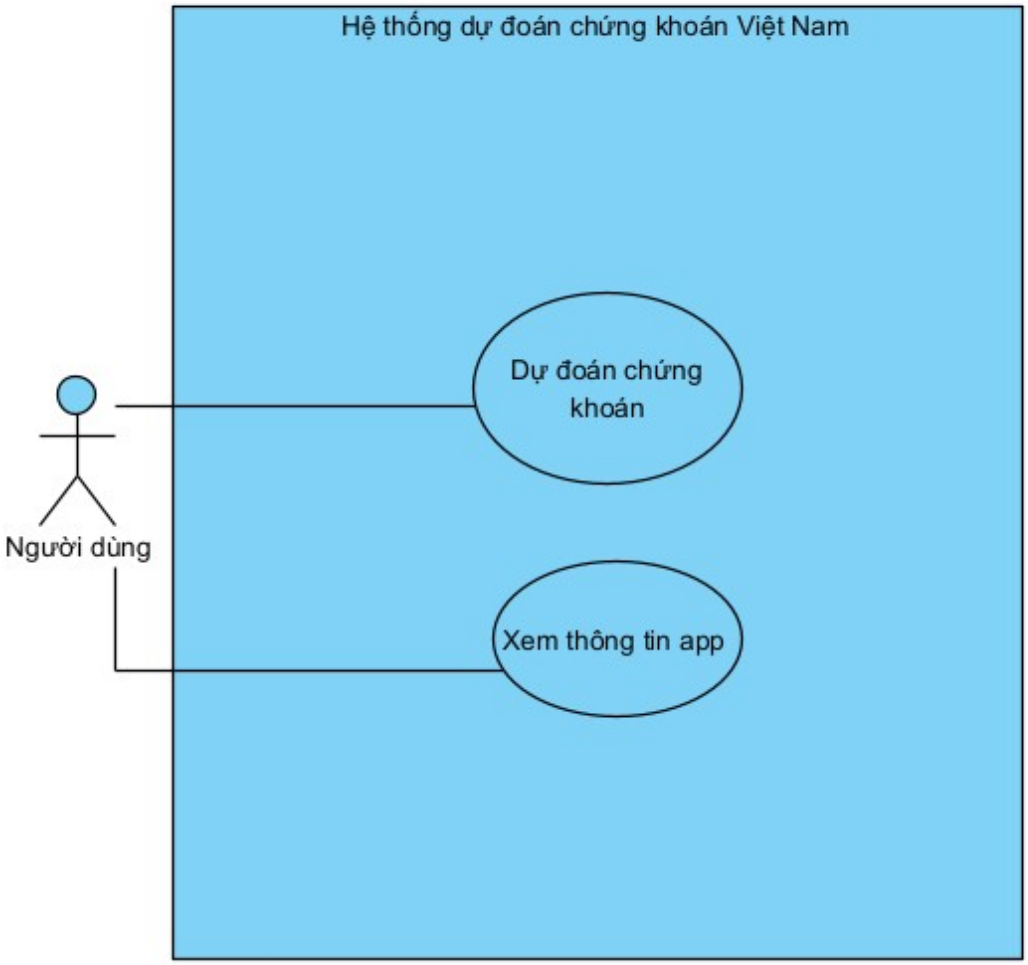
- Xem thông tin chi tiết của app
- Xem thông tin chi tiết của từng mã chứng khoán
- Tìm kiếm, chọn mã chứng khoán để xem thông tin
- Trỏ vào các vị trí trong biểu đồ trực quan hoá thông tin của mã chứng khoán để xem thông tin chi tiết của từng ngày

3.2 Thiết kế hệ thống

Phần thiết kế hệ thống sẽ bao gồm thiết kế về các Usecase tổng quan cùng với các biểu đồ lớp thực thể.

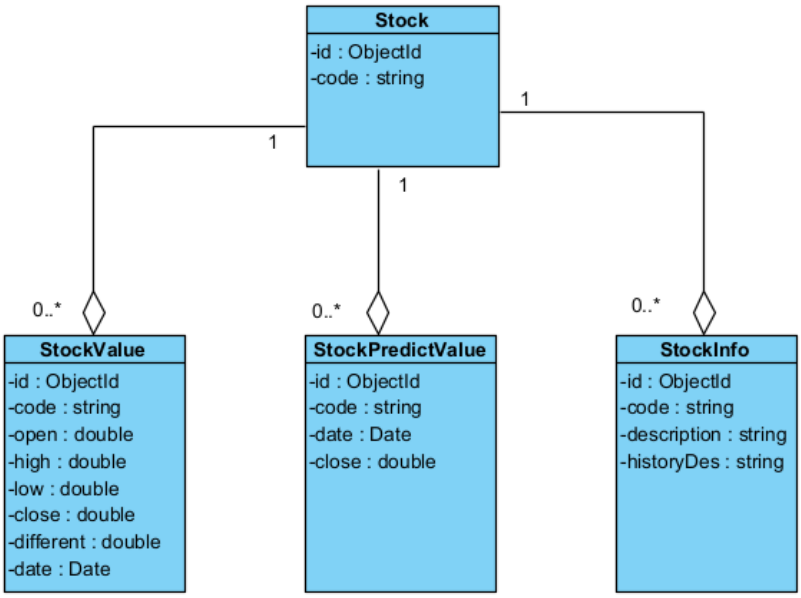
Trong phần này sẽ trình bày tổng quát nhất về những yêu cầu hệ thống, những chức năng mà hệ thống cần có.

3.2.1 Usecase tổng quan



Hình 3.45: Usecase tổng quan hệ thống

3.2.2 Biểu đồ lớp thực thể



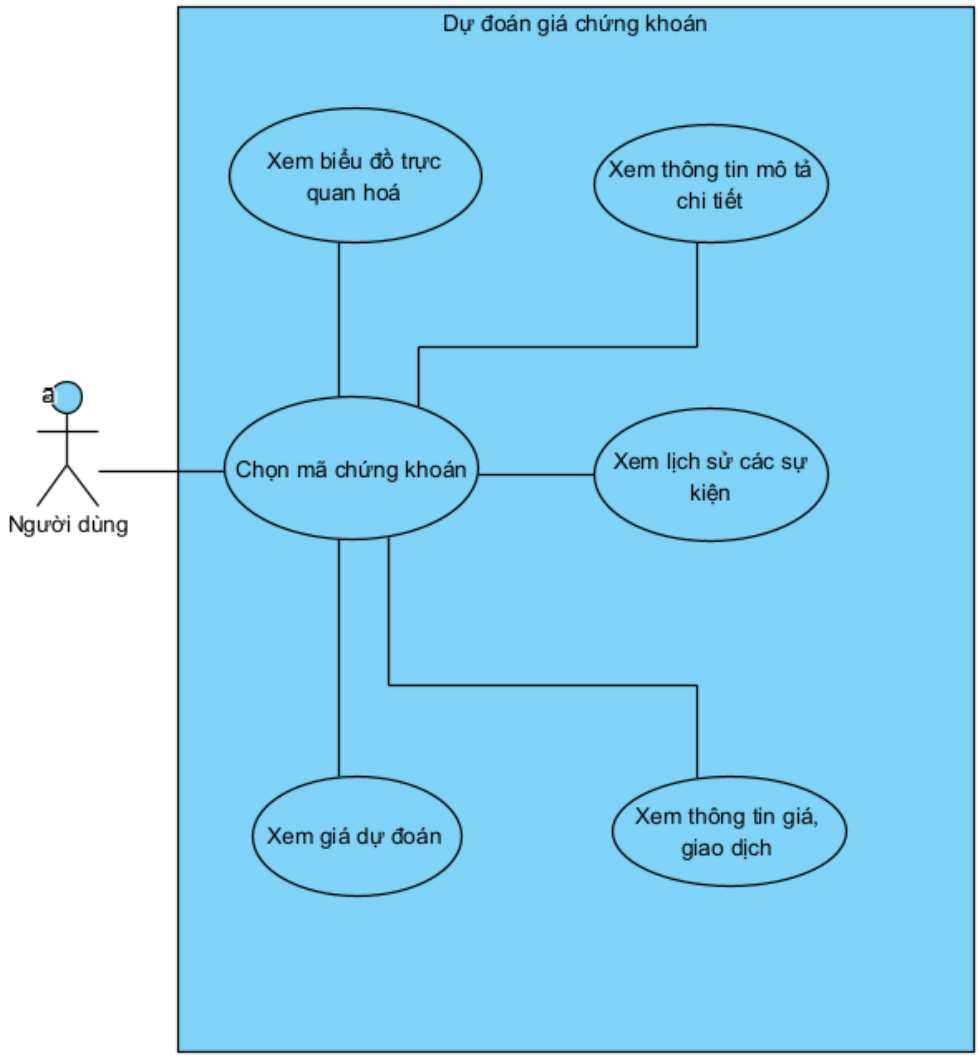
Hình 3.46: Biểu đồ lớp thực thể

3.3 Phân rã các module

3.3.1 Module dự đoán giá chứng khoán

Trong phần này sẽ trình bày về usecase chi tiết, kịch bản và biểu đồ tuần tự của module dự đoán giá chứng khoán.

a) Usecase chi tiết



Hình 3.47: Usecase module dự đoán giá chứng khoán

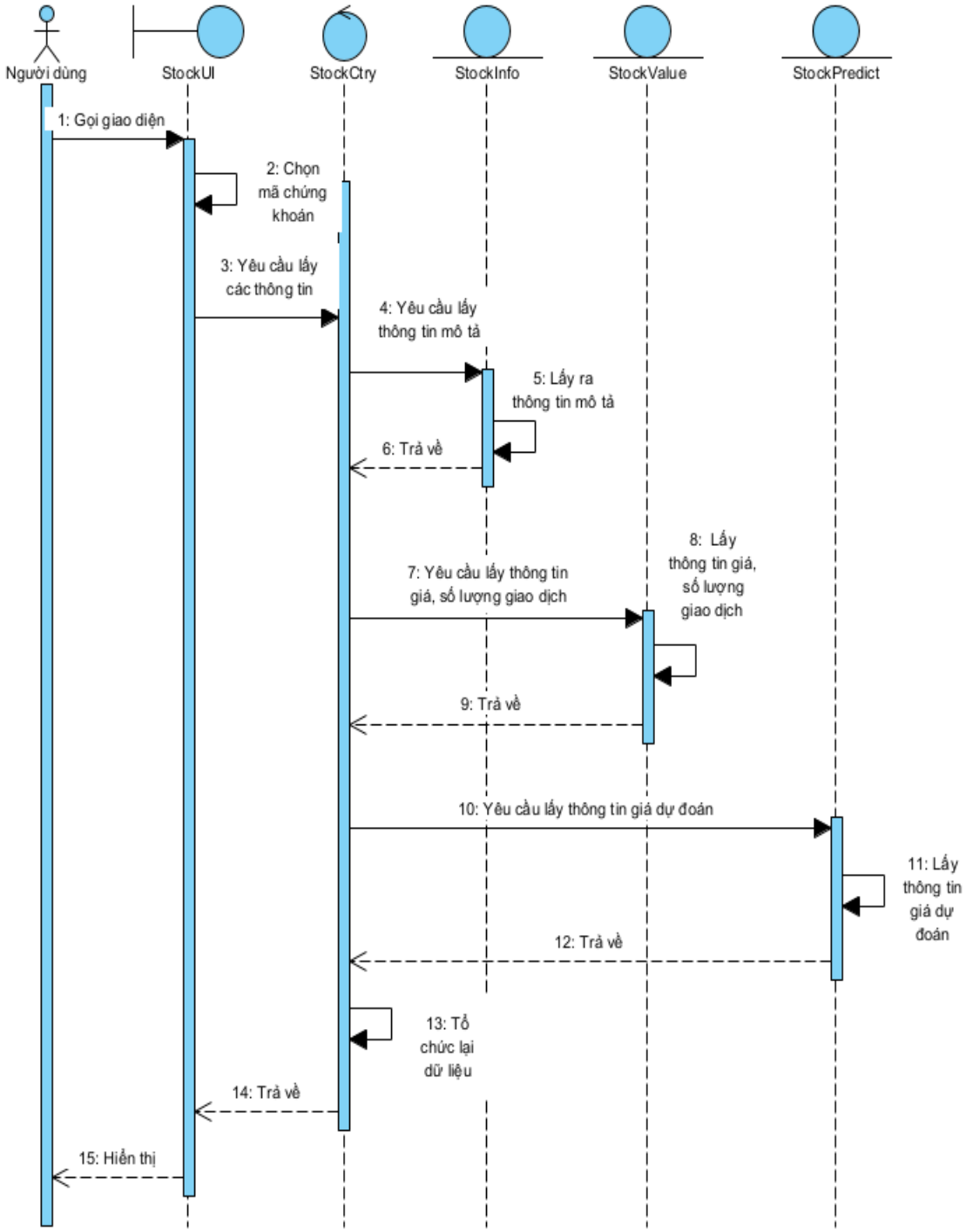
b) Kịch bản

Tên usecase	Dự đoán giá chứng khoán
Actor	Người dùng
Tiền điều kiện	Người dùng cài đặt app thành công
Hậu điều kiện	Người dùng xem được các thông tin như: lịch sử sự kiện, biểu đồ trực quan hoá, thông tin mô tả chi tiết, giá dự đoán, thông tin giá, giao dịch
Kịch bản	1. Người dùng mở app 2. Giao diện trang dự đoán chứng khoán hiện ra 3. Người dùng chọn một mã chứng khoán ở scroll box hoặc nhập mã chứng khoán vào đó

	4. Hệ thống tiếp nhận code của mã chứng khoán mà người dùng chọn sau đó sẽ truy cập vào cơ sở dữ liệu để lấy các thông tin về mã chứng khoán đó để hiển thị ra 5. Thông tin mã chứng khoán người dùng chọn sẽ hiển thị ra ở giao diện bao gồm : thông tin lịch sử sự kiện, thông tin về giá và khối lượng giao dịch, giá cả tăng giảm, giá đóng cửa dự đoán và thông tin về mã chứng khoán. Biểu đồ trực quan hoá sẽ là biểu đồ đường mô tả giá đóng cửa qua các ngày 6. Người dùng chọn vào một điểm trong phần giá thực tế ở biểu đồ trực quan hoá 7. Hệ thống hiển thị các thông tin về giá mở cửa, giá cao nhất, giá thấp nhất và giá đóng cửa của ngày hôm đó 8. Người dùng chọn vào một điểm trong phần giá dự đoán ở biểu đồ trực quan hoá 9. Hệ thống hiển thị thông tin về giá đóng cửa dự đoán và ngày dự đoán
Ngoại lệ	3.1 Mã chứng khoán không tồn tại 3.1.1 Hệ thống không hiển thị mã chứng khoán nào cả, hiển thị ra là không tồn tại mã chứng khoán 3.1.2 Người dùng gõ lại mã chứng khoán khác muốn tìm kiếm 5.1 Giá dự đoán của mã chứng khoán chưa được cập nhật 5.1.1 Giá dự đoán và giá của ngày gần nhất được crawl sẽ hiển thị ra

Bảng 3.2 Kịch bản module dự đoán giá chứng khoán

c) Biểu đồ tuần tự

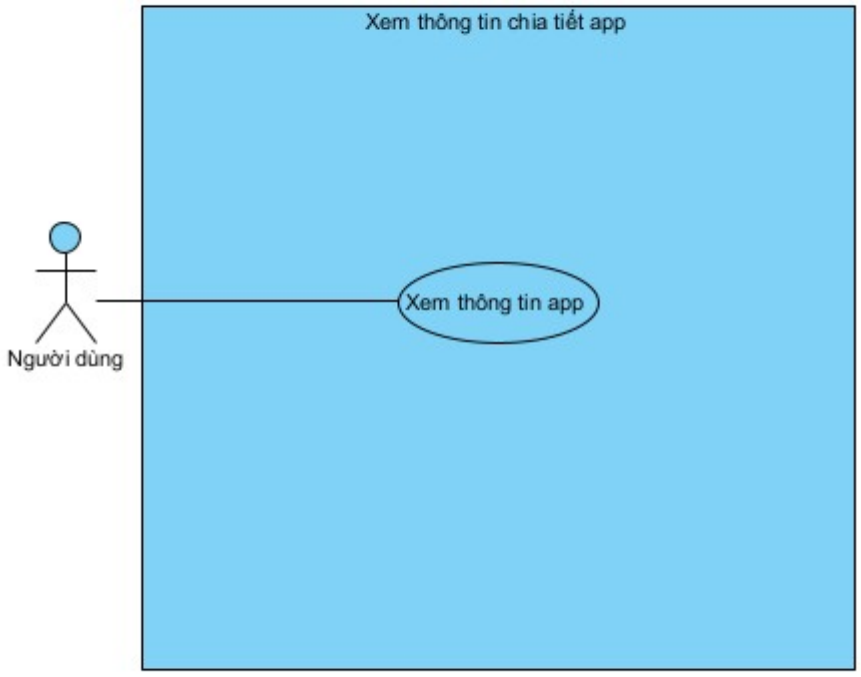


Hình 3.48: Sơ đồ hoạt động module dự đoán giá chứng khoán

3.3.2 Module xem thông tin chi tiết về app

Trong phần này sẽ trình bày về usecase chi tiết, kịch bản và biểu đồ tuần tự của module xem thông tin chi tiết về app.

a) *Usecase chi tiết*


Xem thông tin chia tiết app
Xem thông tin app
Người dùng

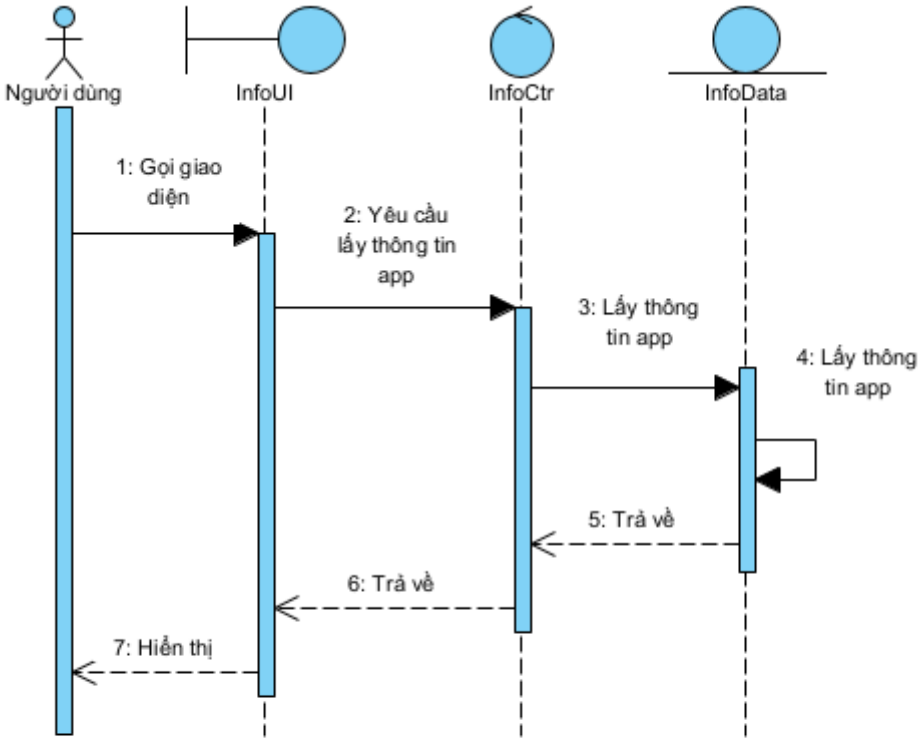
Hình 3.49: Usecase xem thông tin app

b) Kịch bản

Actor	Người dùng
Tiền điều kiện	Người dùng cài đặt app thành công
Hậu điều kiện	Người dùng xem được thông tin về app và thông tin về thuật toán được sử dụng trong app
Kịch bản	1. Người dùng mở app 2. Giao diện trang dự đoán chứng khoán hiện ra 3. Người dùng chọn tab "thông tin app" 4. Hệ thống tiếp nhận yêu cầu lấy thông tin của app và lấy ra thông tin của app ở trên cơ sở dữ liệu 5. Thông tin của app hiện ra màn hình tab "thông tin app" với các dữ liệu về : + Định nghĩa về dự đoán chứng khoán + Nguyên tắc hoạt động của app + Mô hình sử dụng và các lý thuyết về mô hình sử dụng trong app 6. Người dùng cuộn xuống dưới để xem thêm các thông tin và có thể quay trở lại trang xem thông tin chứng khoán
Ngoại lệ	**Không có ngoại lệ**

Bảng 3.3 Kịch bản module xem thông tin chi tiết về app

c) Biểu đồ tuần tự



Hình 3.50: Biểu đồ tuần tự module xem chi tiết app

3.4 Yêu cầu hệ thống

- Hệ điều hành: window, linux
- Cài đặt các trình biên dịch: c++, java, python,...
- Git: Quản lý source code

3.5 Một số công cụ, thư viện hỗ trợ

- Visual studio code
- Android Studio
- Tensorflow
- Sklearn
- Pandas
- Pickletools
- Kotlin
- Java
- Vnstock

3.6 Cài đặt

3.6.1 Cài đặt phân tích dữ liệu và crawl dữ liệu

Bước 1: Mở terminal trên visual studio code, hoặc cmd trên windows.

Bước 2: Chuyển hướng đến thư mục chứa mã nguồn:

Sử dụng câu lệnh: cd <directory path>

Bước 3: Cài đặt toàn bộ thư viện phụ thuộc của dự án.

Sử dụng câu lệnh: pip install + <tên thư viện>

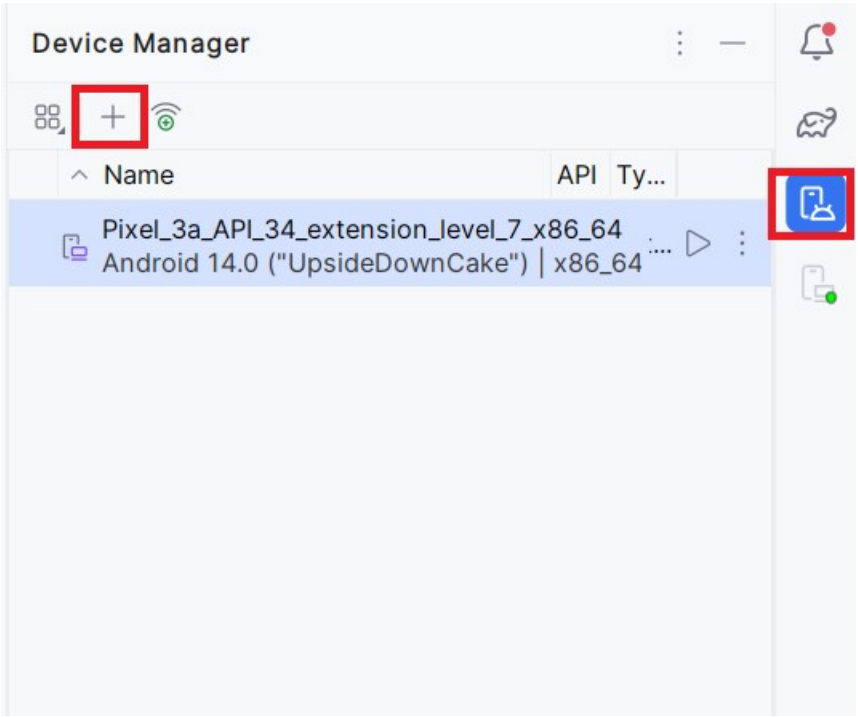
- Với các thư viện sau : Tensorflow
- Sklearn
- Pandas
- Pickletools
- Vnstock

3.6.2 Cài đặt mã nguồn ứng dụng

Bước 1: Mở Android Studio

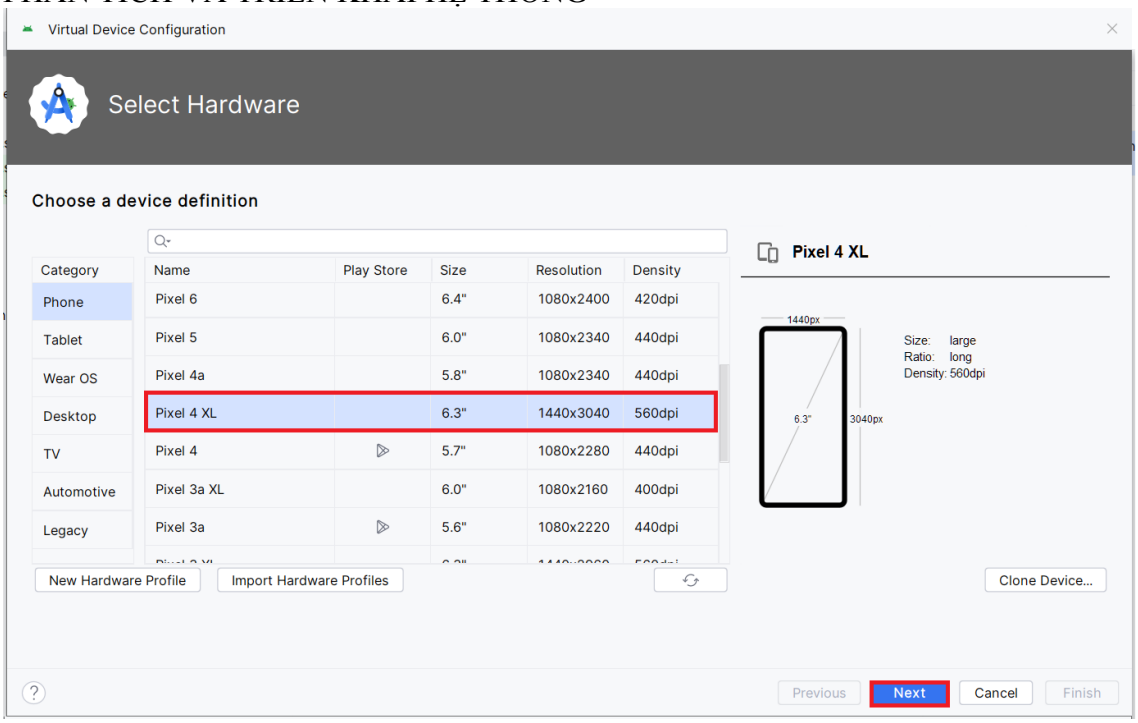
Bước 2: Mở thư mục chứa source code của ứng dụng

Bước 3: Tạo máy ảo bằng cách chọn dấu cộng ở trong mục Device Manager



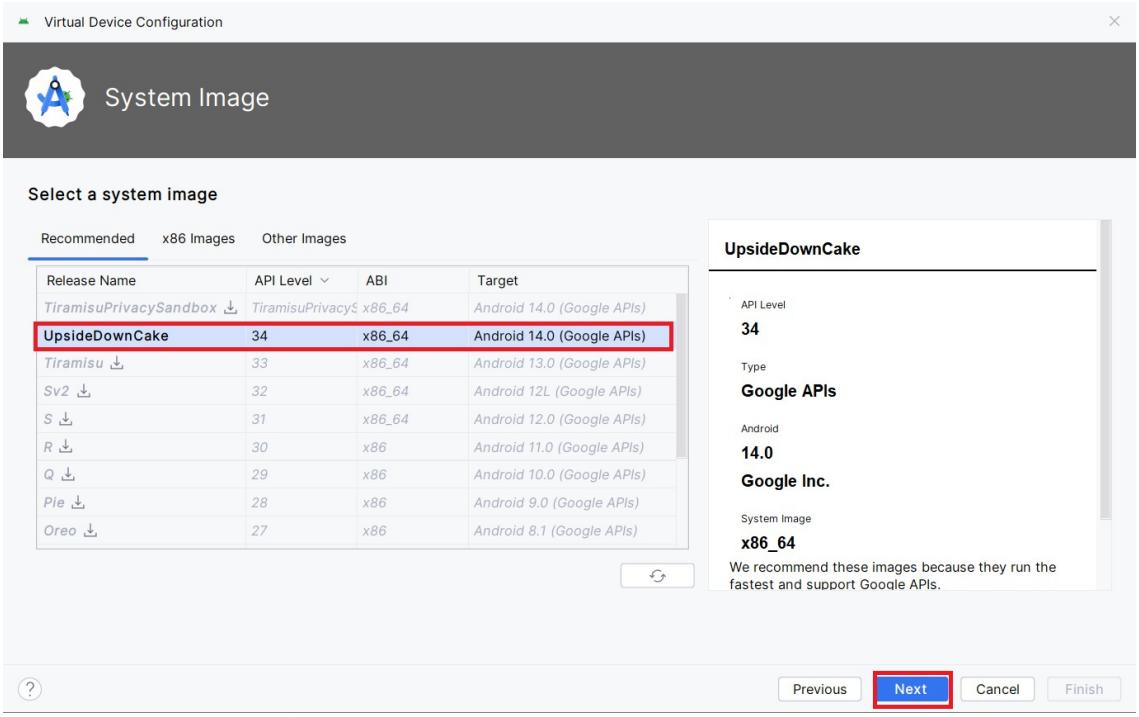
Hình 3.51 Chọn tạo máy ảo

Bước 4 : Trong màn hình chọn thiết bị, chọn Pixel 4 như hình sau đó chọn next



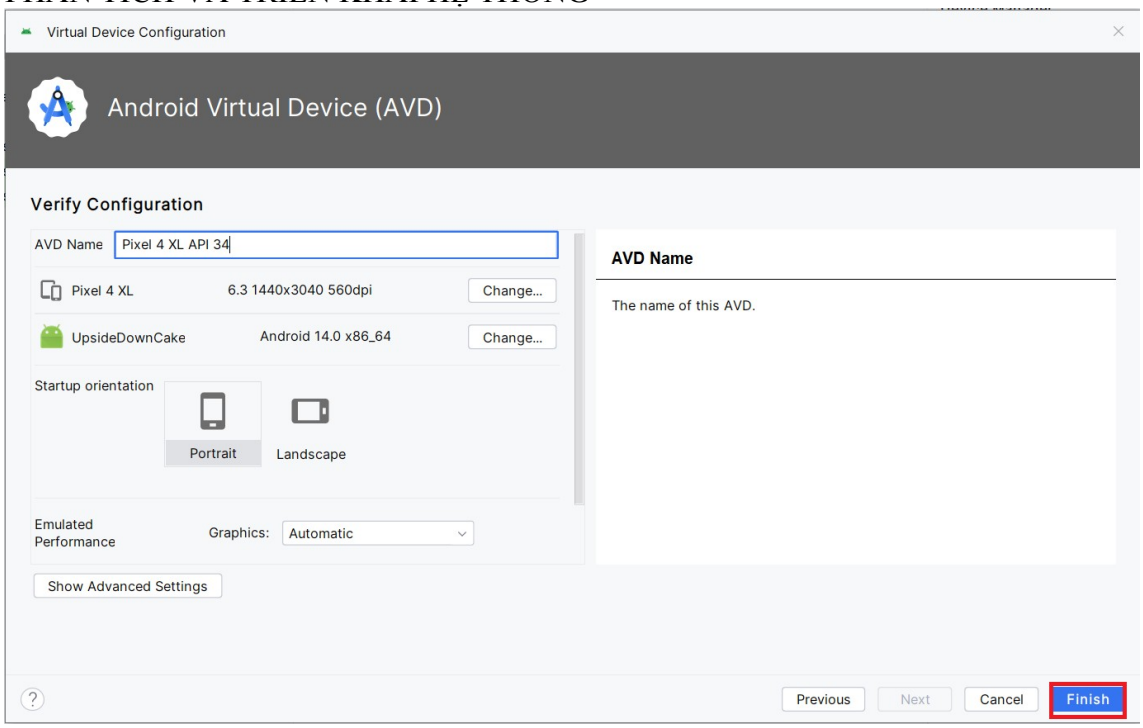
Hình 3.52 Chọn thiết bị Pixel 4

Bước 5 : Trong System Image chọn UpsideDownCake sau đó chọn next



Hình 3.53 Chọn System Image

Bước 4: Nhấn finish để hoàn thành



Hình 3.54 Hoàn thành tạo máy ảo

3.6.3 Chạy ứng dụng trên máy ảo

3.7 Kết quả cài đặt

3.7.1 Kết quả cài đặt dữ liệu

Dữ liệu được chia thành 4 thư mục bao gồm :

- Thư mục dữ liệu về giá chứng khoán
- Thư mục dữ liệu về model dự đoán
- Thư mục dữ liệu về giá dự đoán
- Thư mục dữ liệu về thông tin mã chứng khoán

Dữ liệu về giá chứng khoán thực tế và giá chứng khoán dự đoán được lưu dưới dạng json. Thông tin về các mã chứng khoán được lưu trong file excel. Tất cả các dữ liệu trên tương ứng với 1600 mã chứng khoán và phái sinh.

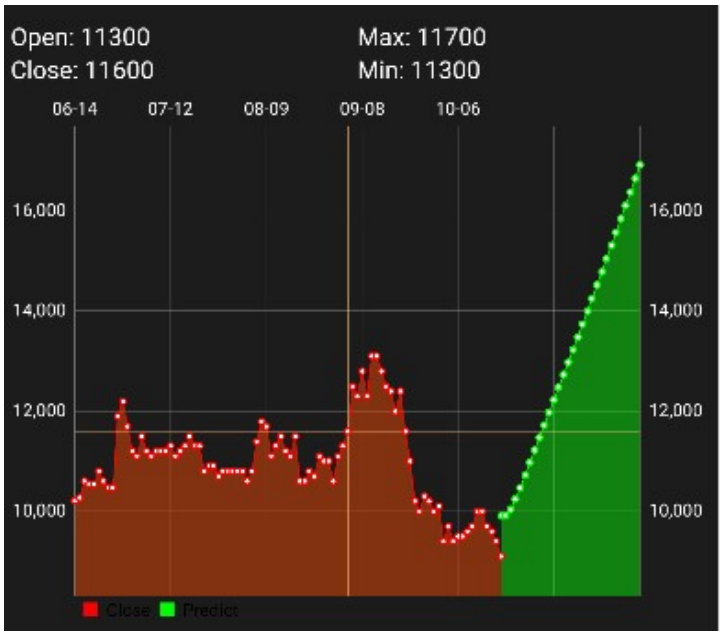
3.7.2 Kết quả cài đặt app

Hiện tại thì ứng dụng đang được phát triển cho hệ điều hành Android. Cần phải có file cài đặt APK cho điện thoại chạy bằng hệ điều hành Android. Sau khi app được cài đặt trên hệ thống bằng Android Studio xong thì sẽ được build ra file APK. Truyền file APK vào điện thoại sau đó cài đặt app bằng cách mở file APK rồi cài đặt như bình thường.

Cách số 2 để chạy ứng dụng là chạy trực tiếp trên máy ảo của Android Studio. Hướng dẫn cài đặt máy ảo đã được mô tả ở phần trước. Dưới đây là một số hình ảnh demo giao diện của phần mềm:

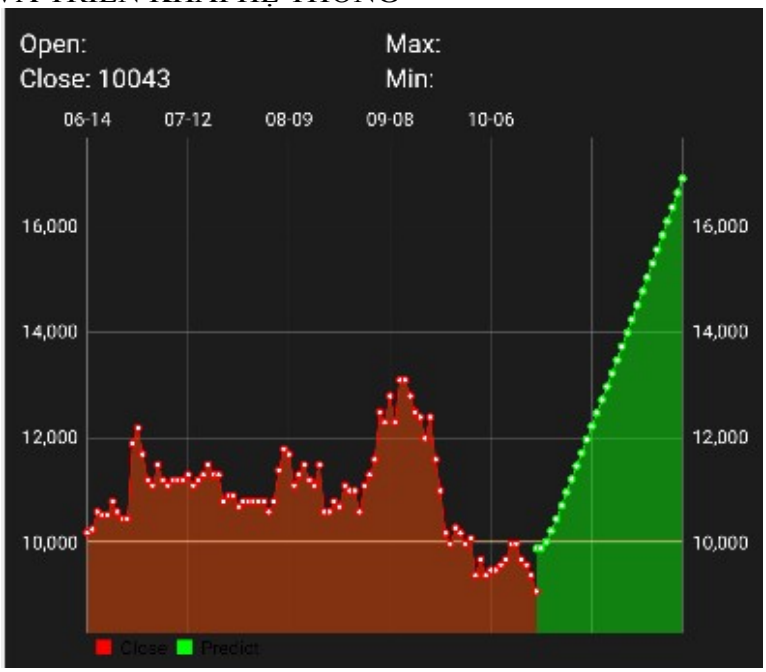
Hình 3.55: Màn hình chính khi vào app và thông tin mã chứng khoán đầu tiên mặc định

Màn hình mặc định khi mở app sẽ là vào màn hình dự đoán và hiện ra các thông tin chi tiết của mã chứng khoán đầu tiên có trong danh sách.



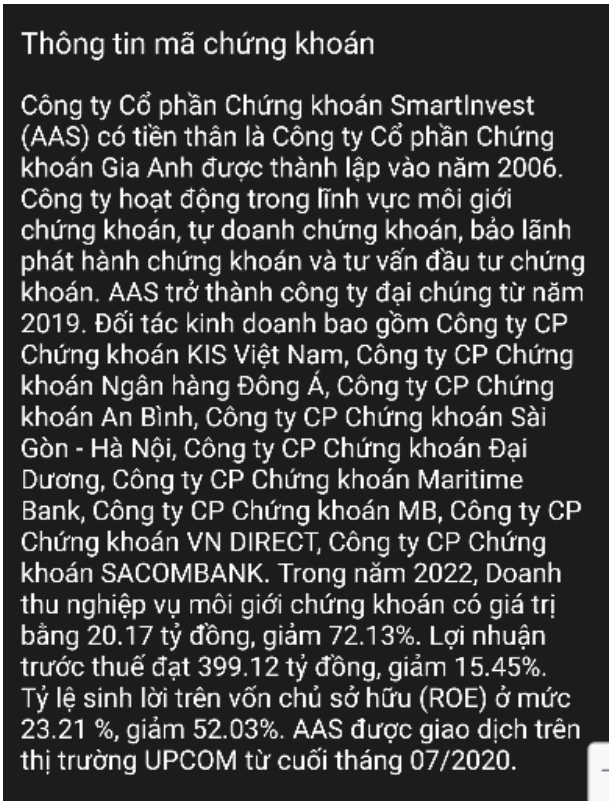
Hình 3.56: Biểu đồ trực quan hoá dữ liệu một mã chứng khoán

Sau khi cuộn màn hình xuống phía dưới sẽ hiển thị biểu đồ trực quan hoá của một mã chứng khoán. Phần màu đỏ là giá thực tế từ trước đến hiện tại. Trỏ vào một vị trí sẽ hiển thị thông tin chi tiết về giá của ngày hôm đó bao gồm giá mở cửa, cao nhất, thấp nhất và giá đóng cửa.



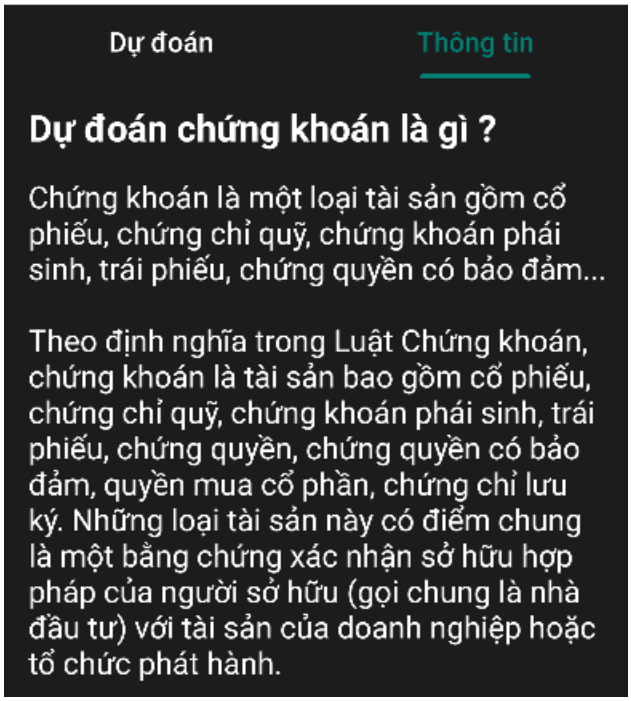
Hình 3.57 Trực quan hoá dữ liệu dự đoán

Khi trỏ vào một điểm trong phần màu xanh sẽ hiển thị giá đóng cửa dự đoán trong tương lai của mã chứng khoán đó. Mỗi điểm chấm tương ứng với một ngày tiếp theo.



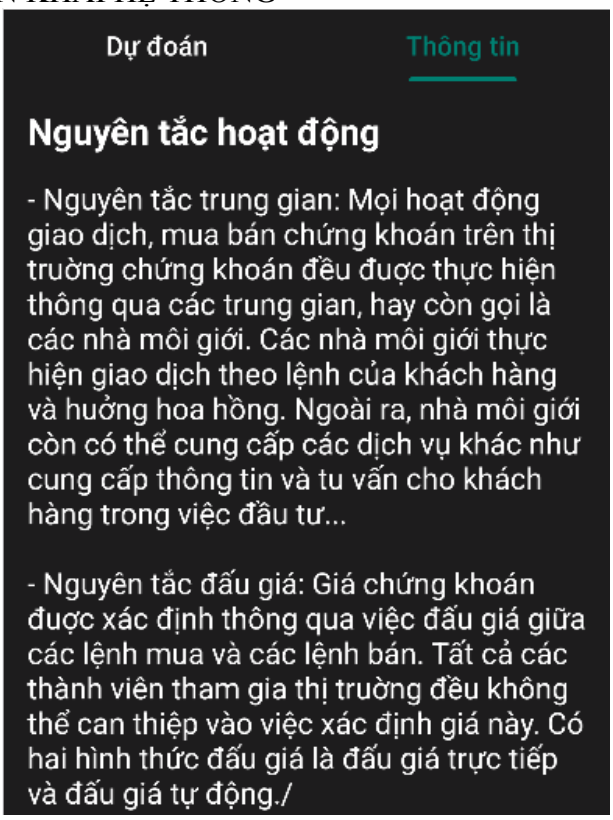
Hình 3.58: Thông tin chi tiết mã chứng khoán

Trang thông tin mã chứng khoán nằm ở dưới cùng phần dự đoán cho biết thông tin chi tiết về mã chứng khoán đang chọn. Bao gồm thông tin về công ty cũng như một số thông tin về tình hình hoạt động của công ty.



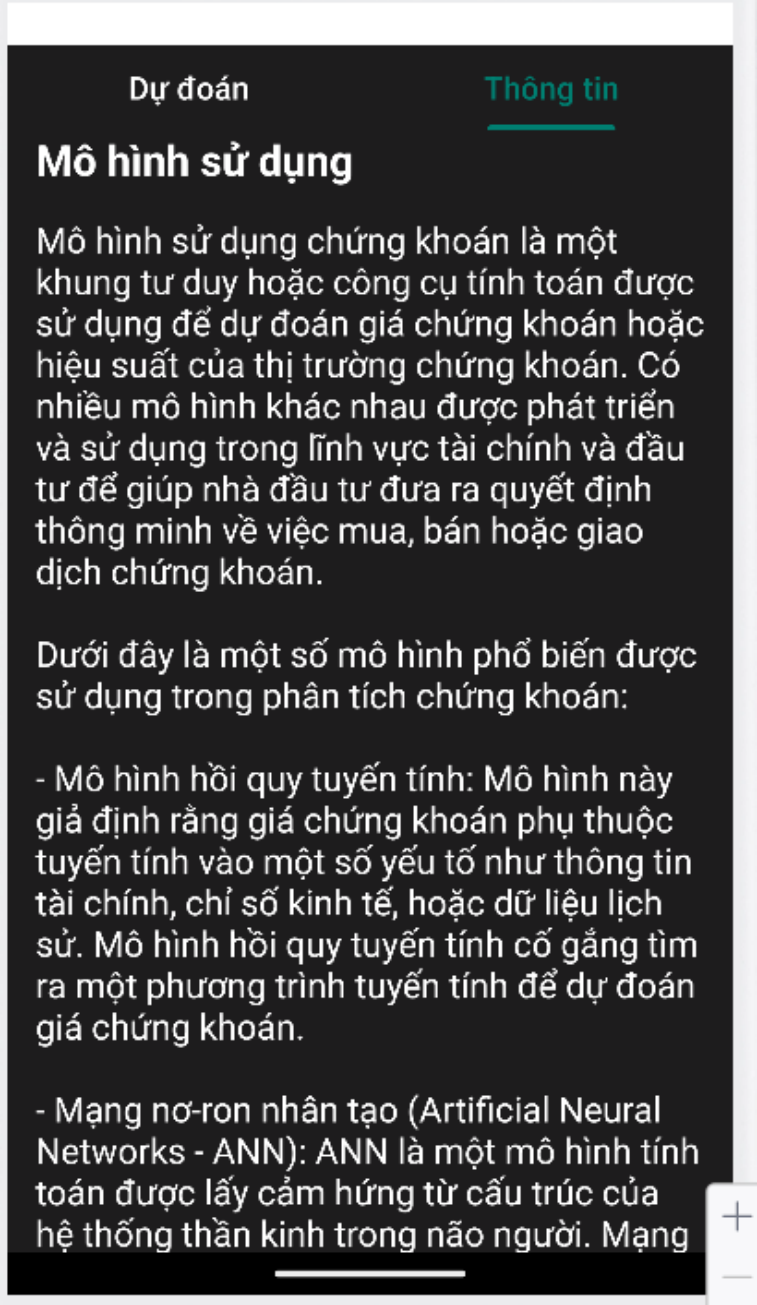
Hình 3.59: Trang thông tin app

Trong tab thông tin của app sẽ bao gồm các thông tin về dự đoán giá chứng khoán, giải thích một số định nghĩa và thông tin đảm bảo không vi phạm pháp luật.



Hình 3.60: Nguyên tắc hoạt động của app

Về nguyên tắc hoạt động của app sẽ dựa trên những nguyên tắc và những thông tin về chứng khoán đảm bảo tính minh bạch và trung thực.



Hình 3.61: Mô hình sử dụng trong app

Mô hình được sử dụng trong app sẽ được mô tả và đặt vào trong phần thông tin của app nhằm giúp cho người dùng muốn phát triển app hay có ý tưởng trong tương lai thì sẽ dựa vào lý thuyết này của app.

3.8 Kết luận chương

Chương này đã trình bày về việc phân tích thiết kế hệ thống, phân tích chi tiết từng use case của hệ thống và cài đặt hệ thống, triển khai hệ thống cụ thể.

Trong chương tiếp theo sẽ đưa ra các kết luận, đánh giá, kết quả của hệ thống và phương hướng phát triển cho tương lai.

PHẦN TỔNG KẾT

Nội dung chính của chương là đưa ra đánh giá về kết quả thu được khi thực hiện đồ án và đề ra phương hướng phát triển cho đồ án

1. Đánh giá kết quả của đồ án

Sau thời gian tìm hiểu, nghiên cứu các công nghệ để thực hiện đồ án với đề tài áp dụng mô hình học máy xây dựng ứng dụng dự đoán chứng khoán Việt Nam, bản thân đã thu được những kết quả cụ thể như sau:

- Học thêm được ngôn ngữ lập trình Kotlin, Python
- Thực hiện crawl dữ liệu từ website
- Nắm cách xử lý dữ liệu và thực hiện xử lý dữ liệu với một khối lượng dữ liệu lớn
- Hiểu thêm lý thuyết về mạng nơ ron nhân tạo
- Lý thuyết về mô hình LSTM và áp dụng vào trong bài toán với tập dữ liệu Time series
- Thiết kế ra hệ thống dự đoán chứng khoán Việt Nam
- Xây dựng app cung cấp thông tin về các mã chứng khoán và dự đoán giá chứng khoán trong tương lai

Bên cạnh đó là những điều cần cải thiện trong đồ án là:

- Còn tồn tại một số dữ liệu khó khăn trong việc xử lý dữ liệu do khối lượng dữ liệu là vô cùng lớn
- Hạn chế về mặt tài nguyên nên chưa được triển khai theo chu kỳ
- Giao diện chỉ ở mức cơ bản hiển thị, chưa được bắt mắt

2. Phương hướng phát triển

Với kết quả như trên, định hướng phát triển trong tương lai cho đồ án là:

- Thêm tính năng mã chứng khoán yêu thích hiển thị ở 1 màn nữa giúp người dùng có thể quan tâm đến một số mã nhất định
- Dữ liệu được đẩy lên sever và lấy ra ở một hệ cơ sở dữ liệu nào đó
- Dữ liệu được cập nhật theo chu kỳ
- Cải thiện lại giao diện thân thiện với người dùng hơn
- Cải thiện ngôn ngữ lập trình để tối ưu code giúp tối ưu hiệu năng và tài nguyên sử dụng trong hệ thống

TÀI LIỆU THAM KHẢO

[1] Jose, J. (2022). Introduction to Time Series Analysis and its Applications. Christ University, Bangalore.

[2] Heck, J. C., & Salem, F. M. (2017, August). Simplified minimal gated unit variations for recurrent neural networks. In 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS) (pp. 1593-1596).

[3] Borovkova, S., & Tsiamas, I. (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. Journal of Forecasting, 38(6), 600-619.

[4] Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. PloS one, 14(2), e0212320.

[5] Nguyen, D. H. D., Tran, L. P., & Nguyen, V. (2019). Predicting stock prices using dynamic LSTM models. In Applied Informatics: Second International Conference, ICAI 2019, Madrid, Spain, November 7–9, 2019, Proceedings 2 (pp. 199-212). Springer International Publishing.

[6] Zhu, J., Yang, Z., Guo, Y., Zhang, J., & Yang, H. (2019). Short-term load forecasting for electric vehicle charging stations based on deep learning approaches. Applied sciences, 9(9), 1723.

[7] Dobrovolny, M., Soukal, I., Salamat, A., Cierniak-Emerych, A., & Krejcar, O. (2021). Forecasting of the Stock Price Using Recurrent Neural Network–Long Short-term Memory.

[8] Sierra-Canto, X., Madera-Ramirez, F., & Uc-Cetina, V. (2010, December). Parallel training of a back-propagation neural network using CUDA. In 2010 Ninth International Conference on Machine Learning and Applications (pp. 307-312).

[9] Li, X., Han, C., Lu, G., & Yan, Y. (2021). Online dynamic prediction of potassium concentration in biomass fuels through flame spectroscopic analysis and recurrent neural network modelling. Fuel, 304, 121376.

[10] Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3128-3137).