

ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN

—o0o—



NGUYỄN THỊ NHÃ LINH

linh.ntn210526@sis.hust.edu.vn

**ỨNG DỤNG MÔ HÌNH HỌC SÂU TIMEXER CHO
BÀI TOÁN DỰ BÁO GIÁ CHỨNG KHOÁN VỚI
BIẾN NGOẠI SINH**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Chuyên ngành: HỆ THỐNG THÔNG TIN QUẢN LÝ

HÀ NỘI – 2025

ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN

—o0o—



**ỨNG DỤNG MÔ HÌNH HỌC SÂU TIMEXER CHO
BÀI TOÁN DỰ BÁO GIÁ CHỨNG KHOÁN VỚI
BIẾN NGOẠI SINH**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Chuyên ngành: HỆ THỐNG THÔNG TIN QUẢN LÝ

Giảng viên hướng dẫn: TS. Trần Ngọc Thắng

Sinh viên thực hiện: Nguyễn Thị Nhã Linh

Chữ ký của GVHD

Mã sinh viên: 20210526

HÀ NỘI – 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Mục tiêu và nội dung của đề án

2. Kết quả đạt được

3. Ý thức làm việc của sinh viên

Hà Nội, Ngày 9 tháng 6 năm 2025

Giảng viên hướng dẫn

TS. TRẦN NGỌC THĂNG

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn chân thành đến TS. Trần Ngọc Thăng – người thầy đã tận tâm hướng dẫn em trong quá trình từ đồ án 1, đồ án 2 cho đến đồ án tốt nghiệp. Nhờ sự hướng dẫn nhiệt tình của thầy, em đã trải qua một hành trình đầy thử thách nhưng cũng là cơ hội quý báu để mở rộng kiến thức và phát triển kỹ năng. Em cũng xin gửi lời cảm ơn đến các thầy cô Đại học Bách khoa Hà Nội đã cung cấp cho em những kiến thức nền tảng vững chắc và tạo điều kiện thuận lợi để em có thể hoàn thành đồ án tốt nghiệp này.

Em xin chân thành cảm ơn!

TÓM TẮT ĐỒ ÁN

Đồ án tập trung xây dựng và đánh giá mô hình TimeXer, so sánh với LSTM, TimeMixer và TSMixer để dự báo chỉ số VNINDEX, nghiên cứu khả năng tích hợp biến ngoại sinh của TimeXer. Siêu tham số được tối ưu bằng hai phương pháp Bayes Optimization: Gaussian Process và Gradient Boosted Trees. Qua nhiều kịch bản thực nghiệm, đồ án đưa ra năm nhận định quan trọng về hiệu suất và tiềm năng ứng dụng của TimeXer. Kết quả cho thấy tính thực tiễn cao, có thể mở rộng bằng cách tích hợp thêm biến ngoại sinh vĩ mô hoặc sử dụng mô hình phức tạp hơn. Trong quá trình thực hiện, em đã nâng cao kỹ năng phân tích dữ liệu, tối ưu hóa mô hình học sâu và hiểu sâu hơn về dự báo chuỗi thời gian.

Hà Nội, ngày ... tháng ... năm 2025

Sinh viên thực hiện

Nguyễn Thị Nhã Linh

Mục lục

1	GIỚI THIỆU ĐỀ TÀI	11
1.1	Động lực phát triển đề án	11
1.2	Mục tiêu của đề án	12
1.3	Bố cục đề án	13
2	CƠ SỞ LÝ THUYẾT	15
2.1	Tổng quan về bài toán dự báo chuỗi thời gian	15
2.1.1	Dữ liệu chuỗi thời gian	15
2.1.2	Các thành phần của một chuỗi thời gian	16
2.2	Bài toán dự báo dữ liệu chuỗi thời gian	18
2.3	Các chỉ số đánh giá bài toán dự báo dữ liệu chuỗi thời gian	20
2.3.1	Chỉ số Mean Absolute Error (MAE)	20
2.3.2	Chỉ số Root Mean Squared Error (RMSE)	21
2.3.3	Chỉ số Mean Absolute Percentage Error (MAPE)	22
2.4	Một số mô hình học sâu dự báo chuỗi thời gian	22
2.4.1	Mô hình Long Short-Term Memory (LSTM)	22
2.4.2	Mô hình TimeMixer và TSMixer	25
2.4.3	Mô hình TimeXer	31
2.5	Phương pháp tối ưu hóa siêu tham số của mô hình	36
2.5.1	Gaussian Process Optimization	37
2.5.2	Gradient Boosted Trees Optimization	38

3	PHÂN TÍCH DỮ LIỆU CHỨNG KHOÁN	
	VỚI MICROSOFT FABRIC	40
3.1	Giới thiệu nền tảng phân tích dữ liệu Microsoft Fabric	40
3.2	Chủ điểm phân tích	42
3.2.1	Giới thiệu	42
3.2.2	Chủ điểm phân tích	43
3.3	Quy trình thu thập, lưu trữ và xử lý dữ liệu	44
3.3.1	Thu thập và lưu trữ dữ liệu ban đầu	44
3.3.2	Xây dựng Pipeline và lịch trình thu thập dữ liệu mới	45
3.4	Xây dựng báo cáo	49
4	ỨNG DỤNG MÔ HÌNH	
	HỌC SÂU TRONG BÀI TOÁN DỰ BÁO CHỈ SỐ VNINDEX	53
4.1	Phát biểu bài toán	53
4.2	Khám phá dữ liệu	54
4.2.1	Mô tả dữ liệu	55
4.2.2	Chuẩn bị dữ liệu	57
4.3	Xây dựng mô hình LSTM dự báo chỉ số VNindex	60
4.4	Xây dựng mô hình TimeMixer và TSMixer dự báo chỉ số VNindex . .	62
4.5	Xây dựng mô hình TimeXer dự báo chỉ số VNindex	66
4.5.1	Mô hình TimeXer đơn biến	68
4.5.2	Mô hình TimeXer đa biến cho dự báo kết hợp một số biến ngoại sinh	70
5	KẾT QUẢ CHẠY KIỂM THỬ MÔ HÌNH	73
5.1	So sánh kết quả của các mô hình	73
5.1.1	So sánh kết quả mô hình LSTM đơn biến và TimeXer đơn biến	73
5.1.2	So sánh kết quả mô hình TimeMixer, TSMixer và TimeXer . .	74

5.1.3	So sánh kết quả mô hình TimeXer đơn biến và mô hình TimeXer kết hợp biến ngoại sinh	75
5.2	Đánh giá và kết luận	77
Kết luận		79
Tài liệu tham khảo		81

Bảng ký hiệu và chữ viết tắt

MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
TSMixer	Time-Series Mixer
FC	Fully-Connected
ReLU	Rectified Linear Unit
GPO	Gaussian Process Optimization
GBTO	Gradient Boosted Trees Optimization
MS Fabric	Microsoft Fabric

Danh sách bảng

4.1	Không gian tìm kiếm siêu tham số tối ưu cho mô hình LSTM	60
4.2	Kết quả tối ưu hóa siêu tham số mô hình LSTM	61
4.3	Siêu tham số tối ưu cho mô hình LSTM	61
4.4	Không gian tìm kiếm siêu tham số cho mô hình TimeMixer	62
4.5	Không gian tìm kiếm siêu tham số cho mô hình TSMixer	63
4.6	Kết quả tối ưu hóa siêu tham số mô hình TimeMixer	64
4.7	Siêu tham số tối ưu cho mô hình TimeMixer	64
4.8	Kết quả tối ưu hóa siêu tham số mô hình TSMixer	65
4.9	Siêu tham số tối ưu cho mô hình TSMixer	65
4.10	Không gian tìm kiếm siêu tham số cho mô hình TimeXer	67
4.11	Kết quả tối ưu hóa siêu tham số mô hình TimeXer đơn biến (dự báo 1 bước)	68
4.12	Kết quả tối ưu hóa siêu tham số mô hình TimeXer đơn biến (dự báo 20 bước)	68
4.13	Siêu tham số tối ưu cho mô hình TimeXer	69
4.14	Kết quả tối ưu hóa siêu tham số mô hình TimeXer kết hợp biến ngoại sinh (dự báo 1 bước)	70
4.15	Kết quả tối ưu hóa siêu tham số mô hình TimeXer kết hợp biến ngoại sinh (dự báo 20 bước)	70
4.16	Siêu tham số tối ưu cho mô hình TimeXer kết hợp biến ngoại sinh . . .	71

5.1	So sánh kết quả mô hình LSTM đơn biến- TimeXer đơn biến	73
5.2	So sánh kết quả mô hình TimeMixer đơn biến- TSMixer đơn biến- TimeXer đơn biến	74
5.3	So sánh kết quả mô hình TimeXer đơn biến- TimeXer có kết hợp biến ngoại sinh ($h=1$)	75
5.4	So sánh kết quả mô hình TimeXer đơn biến- TimeXer có kết hợp biến ngoại sinh ($h=20$)	76

Danh sách hình vẽ

2.1	Ví dụ về dữ liệu chuỗi thời gian	15
2.2	Ví dụ về phân tách chuỗi thời gian ra thành các thành phần riêng lẻ bằng phân rã cộng	18
2.3	Minh họa khối bộ nhớ LSTM	23
2.4	Minh họa kiến trúc tổng thể của TimeMixer, bao gồm Past-Decomposable- Mixing và Future-Multipredictor-Mixing	26
2.5	Past-Decomposable-Mixing	27
2.6	Minh họa mô hình TSMixer	29
2.7	Minh họa mô hình TimeXer	32
3.1	Fabric compute engines	41
3.2	Pipeline thu thập dữ liệu mới	45
3.3	Dataflow xử lý dữ liệu mới cập nhật	46
3.4	Thông tin hoạt động của pipeline được theo dõi	47
3.5	Đặt lịch trình cho pipeline chạy hàng ngày	48
3.6	Dashboard phân tích tổng quan (6/6/2022-6/6/2025)	49
3.7	Dashboard phân tích mức độ biến động (6/6/2022-6/6/2025)	52
4.1	Dữ liệu chỉ số VNindex	55
4.2	Dữ liệu biến ngoại sinh tỷ giá hối đoái VND/USD	56
4.3	Dữ liệu biến ngoại sinh DowJones	57
4.4	Gộp các file dữ liệu và xử lý null	58

4.5	Thông tin về dữ liệu sau khi gộp	58
4.6	Dữ liệu được chỉnh sửa phù hợp yêu cầu của các mô hình	59
4.7	Mô hình TimeMixer dự báo 20 bước	65
4.8	Mô hình TSMixer dự báo 20 bước	66
4.9	Mô hình TimeXer đơn biến dự báo 1 bước	69
4.10	Mô hình TimeXer đơn biến dự báo 20 bước	70
4.11	Mô hình TimeXer dự báo 1 bước kết hợp biến ngoại sinh	71
4.12	Mô hình TimeXer dự báo 20 bước kết hợp biến ngoại sinh	72

Chương 1

GIỚI THIỆU ĐỀ TÀI

1.1 Động lực phát triển đề án

Trong bối cảnh kinh tế toàn cầu hiện nay, thị trường chứng khoán ngày càng trở nên phức tạp và biến động mạnh mẽ do chịu ảnh hưởng của nhiều yếu tố như tình hình kinh tế vĩ mô, chính sách tài chính, các biến động chính trị và tâm lý nhà đầu tư. Việc dự báo chính xác giá cổ phiếu là một trong những thách thức quan trọng đối với các nhà đầu tư và doanh nghiệp tài chính, giúp họ đưa ra quyết định đầu tư hiệu quả và tối ưu hóa lợi nhuận. Một dự báo chính xác có thể mang lại nhiều lợi ích:

- Giảm thiểu rủi ro đầu tư: Giúp các nhà đầu tư có cơ sở ra quyết định dựa trên dự báo hợp lý, từ đó giảm thiểu rủi ro thất thoát vốn do biến động thị trường không lường trước.
- Nâng cao hiệu quả quản lý danh mục đầu tư: Giúp nhà đầu tư và doanh nghiệp tối ưu hóa việc phân bổ nguồn vốn vào các cổ phiếu tiềm năng, góp phần gia tăng lợi nhuận dài hạn.
- Thúc đẩy phát triển các công cụ phân tích tài chính: Nghiên cứu và ứng dụng các mô hình dự báo tiên tiến giúp cung cấp cho thị trường các công cụ phân tích tài chính hiện đại, tăng cường tính minh bạch và tin cậy

Ngoài ra, sự biến động không ngừng của thị trường chứng khoán đặt ra yêu cầu cao về việc phát triển các mô hình dự báo linh hoạt và chính xác. Các nghiên cứu trước đây đã ứng dụng thành công các mô hình học sâu như LSTM, RNN, GRU vào dự báo chuỗi thời gian của chỉ số VNIndex, cho thấy tiềm năng lớn của học sâu trong lĩnh vực tài chính. Tuy nhiên, hầu hết các mô hình này chủ yếu tập trung vào dữ liệu đơn biến hoặc chưa khai thác triệt để các biến ngoại sinh như tỷ giá USD/VND hay một số mã chứng khoán có tầm ảnh hưởng, dẫn đến giới hạn trong khả năng dự báo chính xác và toàn diện.

Trong bối cảnh đó, mô hình TimeXer được lựa chọn để ứng dụng trong đề án nhằm tận dụng ưu thế của kiến trúc linear mixing (temporal và feature mixing), cho phép khai thác hiệu quả mối quan hệ phức tạp giữa biến mục tiêu (chỉ số VNIndex) và các biến ngoại sinh đa dạng. TimeXer không chỉ xử lý tốt chuỗi thời gian nội sinh mà còn tích hợp thông tin từ các biến ngoại sinh một cách linh hoạt và sâu sắc, từ đó nâng cao khả năng dự báo trong môi trường tài chính có nhiều biến động và yếu tố ảnh hưởng đa chiều.

1.2 Mục tiêu của đề án

Mục tiêu chính của đề án là xây dựng một mô hình TimeXer dự báo chỉ số VNIndex với các biến ngoại sinh, nhằm:

- Tăng cường độ chính xác của dự báo biến động chỉ số chứng khoán bằng cách kết hợp dữ liệu chuỗi thời gian lịch sử và các biến ngoại sinh quan trọng.
- Khai thác sức mạnh của kiến trúc Transformer trong việc học các mối quan hệ phức tạp, không tuyến tính giữa các yếu tố ảnh hưởng đến VNIndex, vượt trội hơn các mô hình học sâu truyền thống như LSTM hay RNN đã được nghiên cứu trước đây.
- Đánh giá hiệu quả của TimeXer trong bối cảnh dữ liệu thực tế của thị trường

chứng khoán Việt Nam, từ đó mở rộng hướng nghiên cứu ứng dụng mô hình mới cho các bài toán dự báo chuỗi thời gian đa biến trong tài chính.

Ngoài ra, đề án cũng giới thiệu đến Microsoft Fabric - nền tảng phân tích dữ liệu toàn diện, tích hợp các dịch vụ từ thu thập, xử lý, lưu trữ, phân tích đến trực quan hóa dữ liệu. Với công cụ này, dữ liệu chứng khoán có thể được thu thập hàng ngày, xử lý và chuyển hóa ngay lập tức thành các dashboard, cung cấp công cụ hỗ trợ ra quyết định đầu tư hiệu quả hơn cho các nhà đầu tư cá nhân, tổ chức quản lý quỹ và các nhà phân tích tài chính, giúp họ có cái nhìn dự báo toàn diện và chính xác hơn về xu hướng thị trường.

Đề án hướng tới việc phát triển một mô hình dự báo có tính ứng dụng cao, vừa đảm bảo tính khoa học trong việc khai thác dữ liệu đa chiều, vừa đáp ứng yêu cầu thực tiễn về độ chính xác và khả năng giải thích trong dự báo thị trường chứng khoán Việt Nam. Qua đó, góp phần nâng cao hiệu quả đầu tư và quản lý rủi ro trong môi trường tài chính ngày càng phức tạp và biến động mạnh mẽ hiện nay

1.3 Bố cục đề án

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

- **Chương 2** trình bày cơ sở lý thuyết nền tảng cho bài toán dự báo chuỗi thời gian. Nội dung chương cũng giới thiệu các chỉ số đánh giá hiệu suất, đồng thời phân tích các mô hình học sâu phổ biến như Long Short-Term Memory (LSTM), TimeMixer, TSMixer và TimeXer. Đặc biệt, chương làm rõ đặc điểm nổi bật của TimeXer trong việc tích hợp biến ngoại sinh. Ngoài ra, các phương pháp tối ưu hóa siêu tham số như Gaussian Process và Gradient Boosted Trees cũng được trình bày chi tiết để làm nền tảng cho việc xây dựng mô hình.
- **Chương 3** tập trung vào phân tích dữ liệu chứng khoán, giới thiệu và sử dụng nền tảng Microsoft Fabric để thu thập, xử lý và xây dựng báo cáo cập nhật hàng ngày cho dữ liệu chứng khoán.

- **Chương 4** mô tả quá trình ứng dụng các mô hình học sâu để dự báo chỉ số VNINDEX, bắt đầu từ việc phát biểu bài toán, khám phá và chuẩn bị dữ liệu. Nội dung chương trình bày chi tiết việc xây dựng các mô hình LSTM, TimeMixer, TSMixer và TimeXer, trong đó TimeXer được triển khai ở cả hai dạng đơn biến và đa biến để tích hợp các biến ngoại sinh, nhằm nâng cao độ chính xác của dự báo.
- **Chương 5** tập trung vào kết quả kiểm thử và so sánh hiệu suất giữa các mô hình, bao gồm LSTM đơn biến, TimeXer đơn biến, TimeMixer, TSMixer và TimeXer kết hợp biến ngoại sinh. Dựa trên các kết quả này, chương đưa ra 5 nhận định về tính hiệu quả cũng như tiềm năng ứng dụng của TimeXer trong dự báo giá chứng khoán.

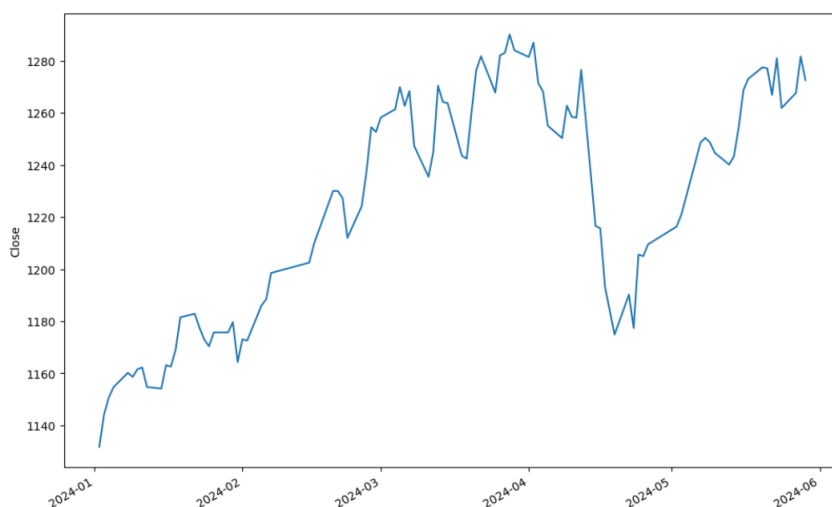
Chương 2

CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về bài toán dự báo chuỗi thời gian

2.1.1 Dữ liệu chuỗi thời gian

Chuỗi thời gian là một tập hợp các quan sát của một biến tại các thời điểm nhất định hay trong những khoảng thời gian nhất định như: giờ, ngày, tháng hoặc năm. Còn trong toán học, dữ liệu chuỗi thời gian được định nghĩa là những điểm dữ liệu đã được đánh chỉ số theo thời gian và có khoảng cách đều nhau giữa những quan sát liên tiếp.



Hình 2.1: Ví dụ về dữ liệu chuỗi thời gian

Dữ liệu chuỗi thời gian có hai thành phần chính: giá trị quan sát và thời gian tương ứng với mỗi quan sát. Giá trị quan sát có thể biểu thị các chỉ số, thu nhập, giá cả, số liệu kinh tế hoặc bất kỳ thông tin nào khác mà chúng ta quan tâm đến trong quá trình theo dõi và phân tích. Thời gian cung cấp thông tin về thứ tự và tần suất thu thập dữ liệu, cho phép chúng ta xác định xu hướng, mô hình hóa và dự đoán các sự thay đổi trong dữ liệu theo thời gian.

2.1.2 Các thành phần của một chuỗi thời gian

Kiến biến thiên của dữ liệu trong một chuỗi thời gian liên quan đến một số thành phần. Thông thường chuỗi thời gian gồm 4 thành phần: xu hướng, chu kỳ, mùa vụ và bất thường.

Thành phần xu hướng

Theo định nghĩa trên, dữ liệu chuỗi thời gian cho thấy biến động ngẫu nhiên nhưng chuỗi thời gian vẫn cho thấy có sự dịch chuyển tăng hoặc giảm dần trong khoảng thời gian dài. Sự dịch chuyển này được xem là thành phần xu hướng. Có 2 dạng chính

- Xu hướng tuyến tính: Xu hướng tăng hoặc giảm của dữ liệu theo thời gian có dạng tuyến tính, tức là tăng/giảm theo một đường thẳng
- Xu hướng phi tuyến: Xu hướng tăng hoặc giảm của dữ liệu theo thời gian có dạng phi tuyến tính. Điều này có nghĩa là xu hướng không theo một đường thẳng, mà có thể là đường cong hoặc theo một mô hình phức tạp hơn.

Thành phần chu kỳ

Chuỗi thời gian có thể biểu hiện tính xu hướng qua khoảng thời gian dài nhưng tất cả các giá trị tương lai sẽ không nằm chính xác trên đường xu hướng. Trong thực tế, chuỗi thời gian thường có hiện tượng luân phiên lên trên và dưới đường xu hướng. Mọi kết quả lặp đi lặp lại của các điểm đó kéo dài hơn 1 năm có thể là do thành phần chu kỳ.

Thành phần mùa

Thành phần mùa vụ thể hiện biến thiên của hiện tượng có tính chất lặp đi lặp lại trong từng thời gian nhất định của năm. Ngoài ra thành phần mùa cũng được sử dụng cho mô hình có sự thường xuyên lặp đi lặp lại với thời gian ít hơn 1 năm.

Thành phần bất thường

Các thành phần bất thường đề cập đến sự sai lệch giữa các giá trị thực tế của chuỗi thời gian với các giá trị dự báo. Nó thường ngắn hạn, bất ngờ, do các yếu tố ngẫu nhiên không thể đoán trước được, biểu thị phần của dữ liệu không thể giải thích được bằng thành phần xu hướng, thành phần chu kỳ, thành phần mùa vụ và các yếu tố khác đã được xác định.

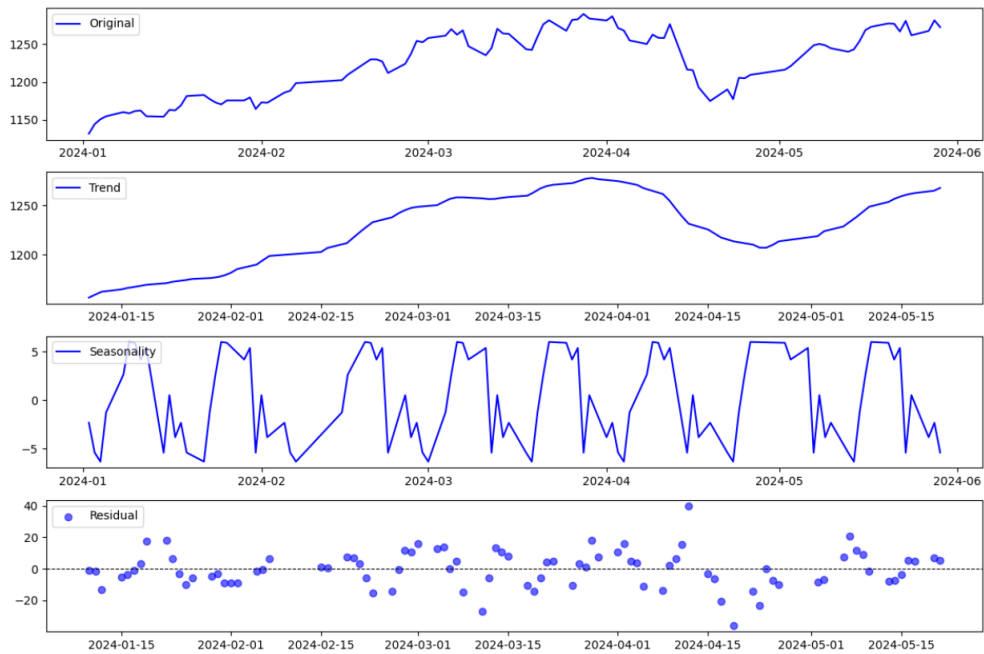
Trong chuỗi thời gian, quá trình phân rã có thể được phân loại thành phân rã cộng và phân rã nhân, dựa trên bản chất của các thành phần khác nhau và cách chúng được cấu thành:

- Phân rã cộng:

$$\text{Chuỗi thời gian} = \text{Xu hướng} + \text{Chu kỳ} + \text{Mùa vụ} + \text{Nhiều}$$

- Phân rã nhân:

$$\text{Chuỗi thời gian} = \text{Xu hướng} \times \text{Chu kỳ} \times \text{Mùa vụ} \times \text{Nhiều}$$



Hình 2.2: Ví dụ về phân tách chuỗi thời gian ra thành các thành phần riêng lẻ bằng phân rã cộng

2.2 Bài toán dự báo dữ liệu chuỗi thời gian

Dự báo chuỗi thời gian là một kỹ thuật thống kê đã và đang được ứng dụng rộng rãi trong nhiều lĩnh vực như tài chính, kinh tế và công nghiệp. Mục tiêu chính của kỹ thuật này là dự đoán giá trị của các biến trong tương lai dựa trên lịch sử của chúng. Khả năng dự báo này mang lại lợi ích lớn, chẳng hạn như trong phát hiện bất thường, giúp ước lượng các giá trị quan trọng có thể đe dọa đến hệ thống đang giám sát; hỗ trợ dự báo các chỉ số rủi ro trong tài chính; hoặc dự đoán xu hướng và mô hình hành vi tiêu dùng, doanh số bán hàng và các chỉ số tiếp thị khác trong marketing.

Ở dạng đơn giản nhất, một mô hình dự báo một bước tiếp theo có dạng:

$$\hat{y}_{i,t+1} = f(\mathbf{y}_{i,t-k:t}, \mathbf{x}_{i,t-k:t}, s_i)$$

Trong đó:

- $\hat{y}_{i,t+1}$: giá trị dự báo của mô hình.

- $\mathbf{y}_{i,t-k:t} = \{y_{i,t-k}, \dots, y_{i,t}\}$: các giá trị quan sát được của biến mục tiêu trong khoảng thời gian k giai đoạn trước đó.
- $\mathbf{x}_{i,t-k:t} = \{x_{i,t-k}, \dots, x_{i,t}\}$: các giá trị quan sát được của các biến đầu vào từ bên ngoài trong khoảng thời gian tương tự.
- s_i : siêu dữ liệu tĩnh liên quan đến thực thể i .
- $f(\cdot)$: hàm dự báo học được từ mô hình.

Các hàm dự báo $f(\cdot)$ có thể được thực hiện thông qua các thiết kế mô hình phù hợp với từng lĩnh vực cụ thể, sẽ được mô tả chi tiết trong các phần sau. Một mô hình cùng hàm dự báo tương ứng có thể được huấn luyện và kiểm định dựa trên các tập dữ liệu ứng dụng cụ thể.

Phân tích và dự báo chuỗi thời gian đã được nghiên cứu chuyên sâu hơn 40 năm. Một trong những phương pháp truyền thống để dự báo chiều biến động của cổ phiếu là mô hình Trung bình di động tích hợp tự hồi quy (ARIMA) được sử dụng để nghiên cứu các quá trình thay đổi theo thời gian. Tuy nhiên, một hạn chế của ARIMA là xu hướng tự nhiên của nó tập trung vào các giá trị trung bình của chuỗi dữ liệu quá khứ. Do đó, vẫn còn khó khăn để nắm bắt một quá trình thay đổi nhanh chóng. Hỗ trợ hồi quy vectơ (SVR) đã được áp dụng thành công để dự đoán chuỗi thời gian, nhưng nó cũng có những nhược điểm như thiếu phương pháp có cấu trúc để xác định một số tham số chính của mô hình. Tuy nhiên, dữ liệu cổ phiếu là phi tuyến, không cố định thậm chí có yếu tố mùa vụ, vì vậy khó đáp ứng các giả định của mô hình. Để khắc phục những hạn chế này, các nhà nghiên cứu đã bắt đầu ứng dụng các mô hình học sâu vào bài toán dự báo chuỗi thời gian. Các mô hình học máy phổ biến nhất hiện nay như mạng nơ-ron hồi quy (RNN - Recurrent Neural Network), mạng nơ-ron với bộ nhớ ngắn hạn định hướng dài hạn (LSTM - Long Short Term Memory networks), mạng nơ-ron hồi tiếp với nút cổng (GRU – Gated Recurrent Unit) đã chứng minh được khả năng vượt trội trong việc xử lý và dự báo các chuỗi dữ liệu phức tạp. Gần đây, kiến trúc Transformer, vốn rất thành công trong xử lý ngôn ngữ tự nhiên, đã được áp

dụng hiệu quả cho bài toán dự báo chuỗi thời gian nhờ khả năng học mối quan hệ dài hạn và song song hóa quá trình huấn luyện. Transformer sử dụng cơ chế attention giúp mô hình tập trung vào các phần quan trọng trong chuỗi dữ liệu mà không bị giới hạn bởi tính tuần tự như RNN hay LSTM. Tuy nhiên, trong các bài toán dự báo chuỗi thời gian có biến ngoại sinh phức tạp như dữ liệu chứng khoán, việc tận dụng hiệu quả thông tin từ nhiều nguồn biến đa dạng vẫn là thách thức. Để giải quyết vấn đề này, mô hình TimeXer được phát triển như một biến thể nâng cao của Transformer, thiết kế đặc biệt để khai thác đồng thời thông tin từ chuỗi biến nội sinh (như giá cổ phiếu lịch sử) và các biến ngoại sinh. TimeXer sử dụng cơ chế attention đa chiều, kết hợp self-attention theo từng đoạn thời gian và cross-attention giữa các biến, cùng với token toàn cục giúp truyền tải thông tin tổng hợp giữa các biến. Nhờ đó, TimeXer có khả năng nắm bắt các mối quan hệ phi tuyến, phức tạp và thay đổi nhanh chóng trong dữ liệu chứng khoán.

2.3 Các chỉ số đánh giá bài toán dự báo dữ liệu chuỗi thời gian

Trong việc nghiên cứu bài toán xây dựng mô hình dự báo chuỗi thời gian, việc đánh giá kết quả của các mô hình là cực kỳ quan trọng để xác định độ chính xác và hiệu quả của chúng. Để thực hiện điều này, chúng ta cần sử dụng các chỉ số đánh giá dự báo. Các chỉ số này phản ánh các tiêu chí khác nhau và phù hợp với các tình huống cụ thể. Sau khi tìm hiểu và tham khảo các nghiên cứu liên quan, em sẽ chọn lọc một số chỉ số đánh giá phù hợp như:

2.3.1 Chỉ số Mean Absolute Error (MAE)

Chỉ số Mean Absolute Error (MAE) là được sử dụng để đo lường độ lớn trung bình của sai số giữa giá trị dự báo và giá trị quan sát trong bài toán dự báo. MAE được tính bằng cách lấy trung bình của giá trị tuyệt đối của các sai số. Giá trị MAE càng

nhỏ thì mô hình dự báo càng chính xác. MSE được xác định dựa theo công thức:

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

trong đó:

- n là số lượng điểm dữ liệu
- y_i là giá trị quan sát tại điểm dữ liệu thứ i
- \hat{y}_i là giá trị dự báo tương ứng với điểm dữ liệu thứ i

2.3.2 Chỉ số Root Mean Squared Error (RMSE)

Chỉ số Root Mean Square Error (RMSE) là căn bậc hai của Mean Squared Error (MSE). MSE là một phương pháp thường được sử dụng để đo lường độ lớn của sai số giữa giá trị dự báo và giá trị quan sát trong bài toán dự báo.

MSE được xác định dựa theo công thức:

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

trong đó:

- n là số lượng điểm dữ liệu
- y_i là giá trị quan sát tại điểm dữ liệu thứ i
- \hat{y}_i là giá trị dự báo tương ứng với điểm dữ liệu thứ i

Khi đó chỉ số RMSE có cùng đơn vị giống với đơn vị của dữ liệu đầu vào và được tính bằng công thức:

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

Khi chỉ số MSE và RMSE càng nhỏ, tức là sự khác biệt giữa giá trị thực tế và giá trị dự đoán càng ít, điều này cho thấy mô hình có độ chính xác cao trong việc dự báo.

2.3.3 Chỉ số Mean Absolute Percentage Error (MAPE)

Tương tự MSE và MAE, chỉ số Mean Absolute Percentage Error (MAPE) là một phương pháp thường được sử dụng để đo lường độ chính xác của các dự báo trong bài toán dự báo. MAPE đo lường tỉ lệ trung bình của sai số tuyệt đối so với giá trị quan sát, tính bằng phần trăm. MAPE cũng được sử dụng để so sánh sự hiệu quả của các mô hình khác nhau. Mô hình có MAPE thấp hơn được coi là mô hình tốt hơn. MAPE được xác định dựa theo công thức:

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

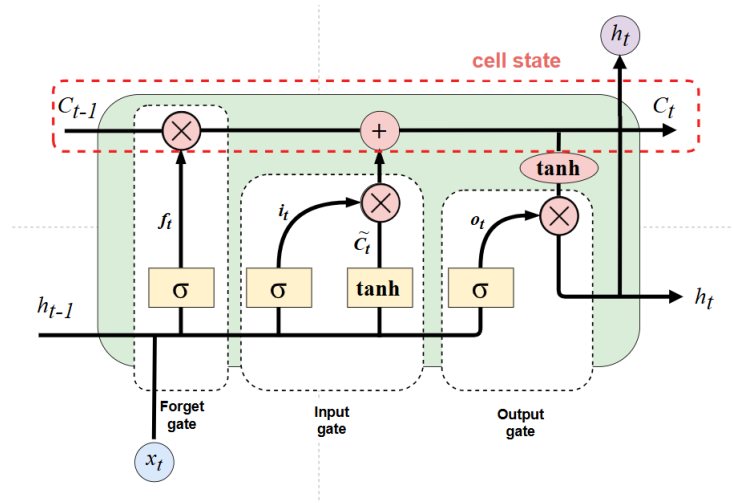
trong đó:

- n là số lượng điểm dữ liệu
- y_i là giá trị quan sát tại điểm dữ liệu thứ i
- \hat{y}_i là giá trị dự báo tương ứng với điểm dữ liệu thứ i

2.4 Một số mô hình học sâu dự báo chuỗi thời gian

2.4.1 Mô hình Long Short-Term Memory (LSTM)

Mô hình LSTM được thiết kế để giải quyết vấn đề vanishing gradient trong quá trình huấn luyện các mạng RNN thông thường. LSTM có khả năng học được sự phụ thuộc trong dài hạn (long-term dependencies) được giới thiệu bởi Hochreiter & Schmidhuber (1997). Kiến trúc này đã được phổ biến và sử dụng rộng rãi cho tới ngày nay.



Hình 2.3: Minh họa khối bộ nhớ LSTM

Thứ tự các bước của LSTM

- **Cell State**

Là thành phần chính của LSTM, được coi như "bộ nhớ dài hạn" (long-term memory). Cell state hoạt động như một đường dẫn lưu giữ thông tin, nơi thông tin có thể được điều chỉnh bởi các cổng (*gates*) trong mạng. Cell state cho phép thông tin được lưu trữ qua nhiều bước thời gian và chỉ thay đổi khi các cổng quyết định thêm hoặc loại bỏ thông tin.

- **Forget Gate (Cổng quên)**

Bước đầu tiên trong LSTM sẽ quyết định xem thông tin nào chúng ta sẽ cho phép đi qua ô trạng thái (*cell state*). Nó được kiểm soát bởi hàm *sigmoid* trong một tầng gọi là tầng quên (*forget gate layer*). Đầu tiên nó nhận đầu vào là 2 giá trị h_{t-1} và x_t sau đó trả về một giá trị f_t nằm trong khoảng 0 và 1 cho mỗi giá trị của ô trạng thái C_{t-1} . Nếu f_t bằng 1 thể hiện 'giữ toàn bộ thông tin' và bằng 0 thể hiện 'bỏ qua toàn bộ thông tin'.

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

trong đó:

- W_f : Trọng số của cổng quên.
- h_{t-1} : Hidden state từ bước thời gian trước.
- x_t : Đầu vào tại bước thời gian hiện tại.
- b_f : Bias của cổng quên.

• **Input Gate (Cổng đầu vào)**

Bước tiếp theo là quyết định thông tin mới nào sẽ được thêm vào trạng thái ô C_t . Điều này được thực hiện qua hai phần:

- Tầng sigmoid quyết định mức độ thông tin mới sẽ được thêm vào:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

- Tầng tanh sẽ tạo ra một véc tơ của một giá trị trạng thái mới \tilde{C}_t mà có thể được thêm vào trạng thái.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cuối cùng là cập nhật trạng thái mới trong cell state. Trạng thái tế bào C_t được ví như "bộ nhớ" và sẽ được truyền đi vào những bước thời gian tiếp theo. Để tạo ra C_t , ta tiến hành nhân trạng thái cũ C_{t-1} với hàm f_t để loại bỏ những thông tin không cần thiết. Sau đó cộng thêm phần $i_t * \tilde{C}_t$ để thêm phần thông tin mới được tạo ra

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

• **Output Gate (Cổng đầu ra)**

Output gate quyết định phần nào của trạng thái ô C_t sẽ được dùng để tính hidden state h_t – đầu ra tại bước thời gian hiện tại. Quy trình gồm hai bước:

- Sử dụng tầng *sigmoid* để xác định phần thông tin nào của trạng thái ô sẽ được đưa ra:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

- Trạng thái ô C_t được đưa qua hàm \tanh (đưa giá trị về khoảng $[-1, 1]$) và nhân với o_t để tạo ra hidden state:

$$h_t = o_t * \tanh(C_t)$$

Hidden state h_t là đầu ra cuối cùng tại bước thời gian hiện tại, đồng thời được sử dụng trong bước tiếp theo của LSTM.

2.4.2 Mô hình TimeMixer và TSMixer

1. Mô hình TimeMixer

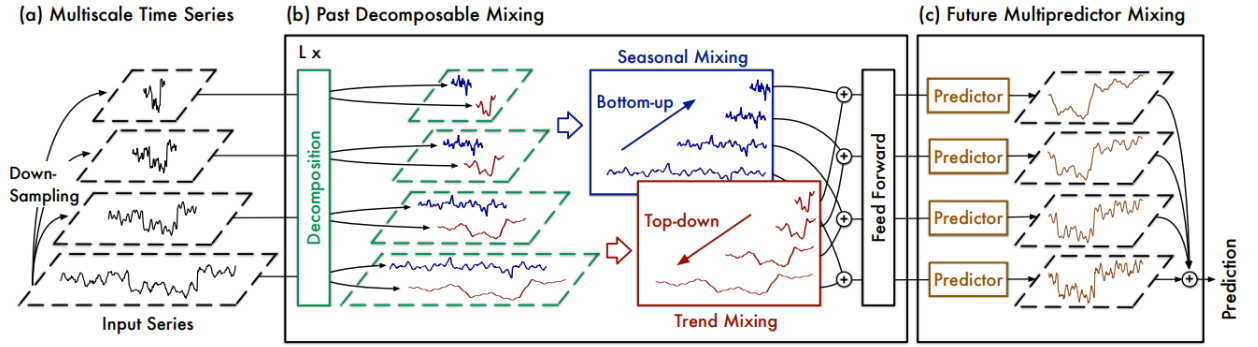
TimeMixer là mô hình được công bố vào tháng 5/2024, là một kiến trúc hoàn toàn dựa trên MLP (Multilayer Perceptron là một loại mạng nơ-ron nhân tạo bao gồm nhiều lớp và thường được sử dụng trong các bài toán học sâu). TimeMixer bao gồm một kiến trúc multiscale-mixing với 2 khối Past-Decomposable-Mixing và Future-Multipredictor-Mixing để trích xuất thông tin trong quá khứ và dự đoán tương lai tương ứng.

- **Multiscale Mixing**

Như được minh họa trong Hình 2.5, để tách biệt các biến động phức tạp, trước tiên thực hiện giảm mẫu (downsample) các quan sát quá khứ $\mathbf{x} \in \mathbb{R}^{P \times C}$ thành M cấp độ bằng cách sử dụng phép gộp trung bình (average pooling), và cuối cùng thu được một tập hợp các chuỗi thời gian multiscale $X = \{\mathbf{x}_0, \dots, \mathbf{x}_M\}$, trong đó $\mathbf{x}_m \in \mathbb{R}^{\lfloor \frac{P}{2^m} \rfloor \times C}$, $m \in \{0, \dots, M\}$, và C biểu thị số lượng biến. Chuỗi cấp thấp nhất $\mathbf{x}_0 = \mathbf{x}$ là chuỗi đầu vào, chứa các biến động thời gian chi tiết nhất, trong khi chuỗi cấp cao nhất \mathbf{x}_M thể hiện các biến động vĩ mô. Sau đó, các chuỗi multiscale được ánh xạ thành các đặc trưng sâu X^0 thông qua tầng embedding, được biểu diễn bởi $X^0 = \text{Embed}(X)$, từ đó tạo ra các biểu diễn multiscale của chuỗi đầu vào.

- **Past-Decomposable-Mixing**

Tiếp theo, kiến trúc sử dụng các khối Past-Decomposable-Mixing (PDM)



Hình 2.4: Minh họa kiến trúc tổng thể của TimeMixer, bao gồm Past-Decomposable-Mixing và Future-Multipredictor-Mixing

được xếp chồng để trộn thông tin quá khứ qua các cấp độ khác nhau. Tổng quan, đối với tầng thứ l , đầu vào là X^{l-1} , và quá trình của **PDM** có thể được biểu diễn như sau:

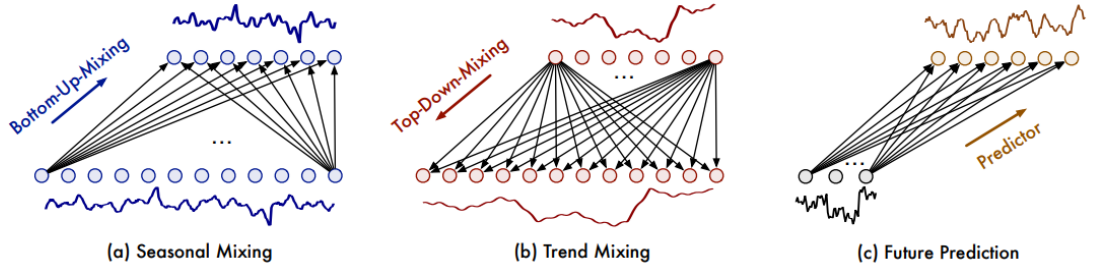
$$X^l = \text{PDM}(X^{l-1}), \quad l \in \{0, \dots, L\}, \quad (2.1)$$

trong đó L là tổng số tầng, và $X^l = \{\mathbf{x}_0^l, \dots, \mathbf{x}_M^l\}$, $\mathbf{x}_m^l \in \mathbb{R}^{\lfloor \frac{P}{2^m} \rfloor \times d_{\text{model}}}$ biểu thị các biểu diễn quá khứ đã được trộn với d_{model} kênh.

Về chi tiết phương thức hoạt động, khối PDM trộn các thành phần mùa vụ và xu hướng đã được phân tách ở các thang đo khác nhau một cách riêng biệt. Điều này cho phép mô hình tập trung vào các xu hướng lịch sử quan trọng và các mẫu theo mùa, đồng thời loại bỏ các biến động ngắn hạn không liên quan hoặc nhiễu.

Từ các tầng tương ứng với các chuỗi thời gian được phân tích với các thang đo khác nhau, chúng được tách thành các thành phần mùa vụ và xu hướng tương ứng:

$$s_m^l, \quad t_m^l = \text{SeriesDecomp}(x_m^l), \quad m \in \{0, \dots, M\}.$$



Hình 2.5: Past-Decomposable-Mixing

Trong đó x_m^l là chuỗi thời gian tại tầng đo m và tầng l . Thành phần mùa vụ và xu hướng của mỗi layer được phân tích qua hàm SeriesDecomp.

$$X^l = X^{l-1} + \text{FeedForward} \left(S\text{-Mix} \left(\left\{ s_m^l \right\}_{m=0}^M \right) + T\text{-Mix} \left(\left\{ t_m^l \right\}_{m=0}^M \right) \right).$$

Một mạng MLP (Multilayer Perceptron) được sử dụng để xử lý và kết hợp thông tin từ các thành phần mùa vụ và xu hướng sau khi trộn. Để tạo ra các chuỗi X^l được làm giàu thông tin từ các layer trước đó.

Seasonal Mixing: Tính mùa vụ gồm các thay đổi ngắn hạn và mang tính lặp lại. Vì vậy thông tin chi tiết từ quy mô nhỏ (fine-scale) lần quy mô lớn (coarse-scale) để tăng cường phân tích mùa vụ.

Với $m: 1 \rightarrow M$ thì:

$$s_m^l = s_m^l + \text{Bottom-Up-Mixing}(s_{m-1}^l)$$

Trend Mixing: Xu hướng phản ánh các thay đổi chậm hơn trong khoảng thời gian dài. Sử dụng kết hợp thông tin tổng quan từ quy mô lớn (coarse-scale) xuống quy mô nhỏ (fine-scale) để dẫn dắt phân tích xu hướng.

Với $m: (M \rightarrow 0)$ thì:

$$t_m^l = t_m^l + \text{Top-Down-Mixing}(t_{m+1}^l)$$

- **Future-Multipredictor-Mixing (FMM)**

Đối với giai đoạn dự đoán tương lai, áp dụng khối Future-Multipredictor-Mixing (FMM) để tổng hợp thông tin quá khứ multiscale đã trích xuất X^L và tạo ra các dự đoán tương lai, được biểu diễn như sau:

$$\hat{\mathbf{x}} = \text{FMM}(X^L), \quad (2.2)$$

trong đó $\hat{\mathbf{x}} \in \mathbb{R}^{F \times C}$ biểu thị dự đoán cuối cùng.

Với các thiết kế trên, **TimeMixer** có thể nắm bắt thành công thông tin quá khứ thiết yếu từ các quan sát đã được tách biệt, và dự đoán tương lai với thông tin hữu ích từ quá khứ.

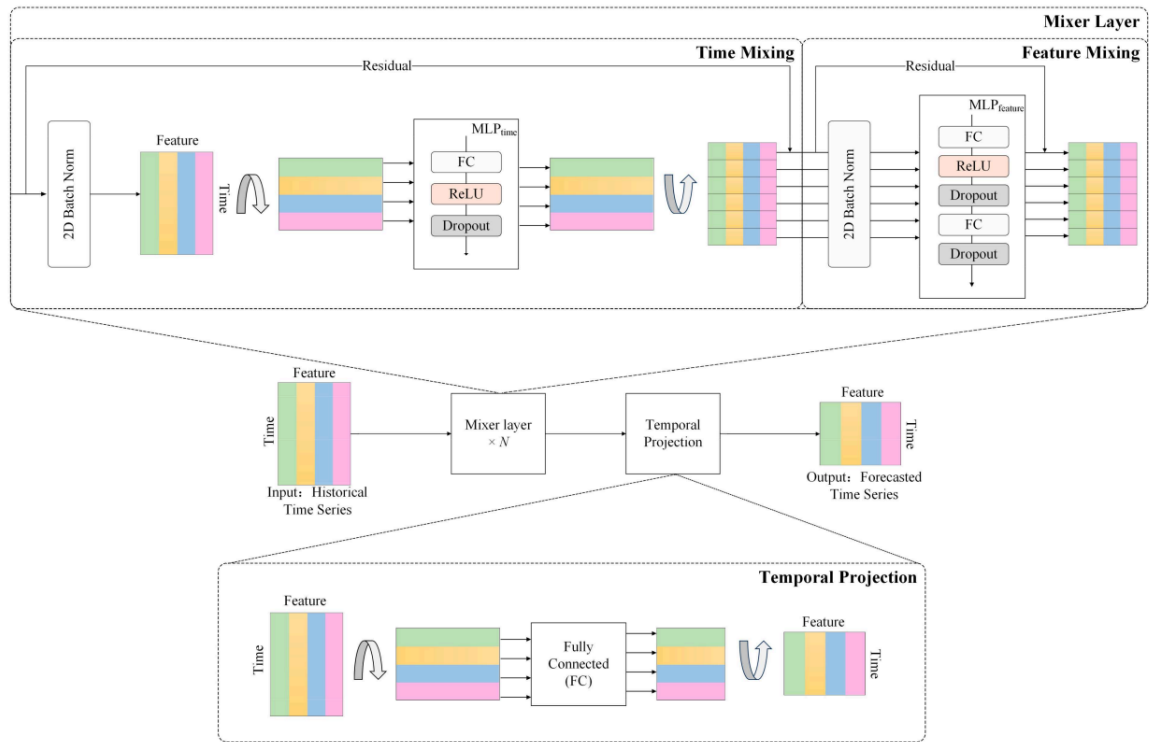
2. Mô hình TSMixer

TSMixer (Time-Series Mixer) là một mô hình dự báo chuỗi thời gian đa biến được công bố bởi Google Cloud AI Research vào tháng 9/2023. Mô hình này được phát triển để giải quyết các thách thức trong dự báo chuỗi thời gian, đặc biệt là trong bối cảnh các mô hình sâu phức tạp như recurrent hoặc attention-based thường không đạt hiệu quả như kỳ vọng so với các mô hình tuyến tính đơn giản. TSMixer được thiết kế để kết hợp sức mạnh của các mô hình tuyến tính với khả năng phi tuyến của các perceptron đa lớp (MLPs), đồng thời giới thiệu các phép trộn để khai thác hiệu quả cả thông tin tạm thời và liên biến.

- **Kiến trúc:** TSMixer được thiết kế như một kiến trúc toàn MLP, trong đó time-mixing và feature-mixing được xen kẽ để xử lý dữ liệu chuỗi thời gian đa biến.

Hình 2.6 thể hiện kiến trúc của mô hình với các cột đầu vào đại diện cho đặc trưng/biến khác nhau, và các hàng là bước thời gian. Các phép toán fully-connected được thực hiện theo hàng. TSMixer bao gồm các MLP time-mixing và feature-mixing xen kẽ để tổng hợp thông tin. Số lượng lớp mixer được ký hiệu là N . Các MLP time-mixing được chia sẻ trên tất cả

các đặc trưng, và các MLP feature-mixing được chia sẻ trên tất cả các bước thời gian. Thiết kế này cho phép TSMixer tự động thích nghi với việc sử dụng cả thông tin tạm thời và liên biến với số lượng tham số hạn chế, mang lại khả năng tổng quát hóa vượt trội.



Hình 2.6: Minh họa mô hình TSMixer

Kiến trúc được minh họa bao gồm các thành phần

- **MLP Time-Mixing:** Trong Time-mixing, kiến trúc sử dụng một MLP một lớp, nơi một mô hình tuyến tính đơn giản để học các mẫu thời gian phức tạp. Các MLP time-mixing mô hình hóa các mẫu tạm thời trong chuỗi thời gian. Chúng bao gồm một lớp fully-connected (FC), theo sau là một hàm kích hoạt (ReLU) và dropout để giảm hiện tượng overfitting bằng cách ngẫu nhiên loại bỏ một số nút trong quá trình huấn luyện. Các MLP này chuyển vị đầu vào để áp dụng các lớp fully-

- connected dọc theo miền thời gian và được chia sẻ bởi các đặc trưng.
- MLP Feature-Mixing: Đối với feature-mixing, một MLP hai lớp được sử dụng để học các biến đổi đặc trưng phức tạp. Các MLP feature-mixing được chia sẻ bởi các bước thời gian và được sử dụng để tận dụng thông tin giữa các biến.
 - Kết nối Residual: Kiến trúc áp dụng các residual connections giữa mỗi lớp time-mixing và feature-mixing. Các kết nối này cho phép mô hình học các kiến trúc sâu hơn một cách hiệu quả và cho phép mô hình bỏ qua các phép toán time-mixing và feature-mixing không cần thiết một cách hiệu quả.
 - Temporal Projection: Temporal projection là một lớp fully-connected được áp dụng trên miền thời gian. Chúng không chỉ học các mẫu tạm thời mà còn ánh xạ chuỗi thời gian từ độ dài đầu vào ban đầu L sang độ dài dự báo mục tiêu T .
 - Normalization: TSMixer áp dụng chuẩn hóa 2D trên cả chiều thời gian và chiều đặc trưng do sự hiện diện của các phép toán time-mixing và feature-mixing.
 - **TSMixer cho dự báo chuỗi thời gian:** Mô hình hoạt động bằng cách áp dụng nhiều lớp xen kẽ giữa Time Mixing và Feature Mixing.
 - Input: $X \in \mathbb{R}^{L \times C}$ với L đặc trưng và C bước thời gian.
 - Lớp Mixer (gồm Time Mixing và Feature Mixing) được áp dụng K lần, với output từ lớp Mixer trước được sử dụng làm input cho lớp Mixer sau. Sau khi đi qua K lớp Mixer, lớp TP (Temporal Projection) thực hiện chiếu dữ liệu từ chiều thời gian L sang chiều thời gian T .
 - Output = $TP_{L \rightarrow T}(\text{Mix}^K(X))$

2.4.3 Mô hình TimeXer

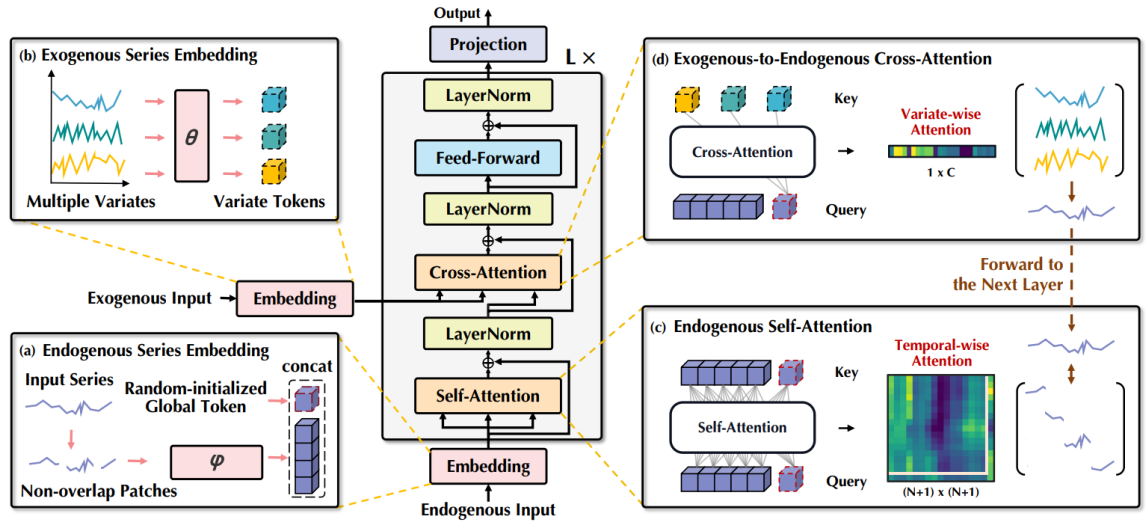
Trong lĩnh vực dự báo chuỗi thời gian, việc tích hợp thông tin từ các biến ngoại sinh (exogenous variables) luôn là thách thức lớn do sự phức tạp trong mối quan hệ đa chiều giữa các biến. Sự ra đời của TimeXer bắt nguồn từ nhu cầu thực tế về các hệ thống dự báo có khả năng xử lý đồng thời cả thông tin nội sinh (endogenous) và ngoại sinh, vượt qua những hạn chế của các phương pháp truyền thống.

Trong bài toán dự báo với biến ngoại sinh, dữ liệu cần có bao gồm:

- Một chuỗi thời gian nội sinh $\mathbf{x}_{1:T} = \{x_1, x_2, \dots, x_T\} \in \mathbb{R}^{T \times 1}$, trong đó x_i biểu thị giá trị tại thời điểm thứ i .
- Nhiều chuỗi ngoại sinh $\mathbf{z}_{1:T_{\text{ex}}} = \{\mathbf{z}_{1:T_{\text{ex}}}^{(1)}, \mathbf{z}_{1:T_{\text{ex}}}^{(2)}, \dots, \mathbf{z}_{1:T_{\text{ex}}}^{(C)}\} \in \mathbb{R}^{T_{\text{ex}} \times C}$, trong đó $\mathbf{z}_{1:T_{\text{ex}}}^{(i)}$ là chuỗi ngoại sinh thứ i , và C là số lượng chuỗi ngoại sinh.

T và T_{ex} lần lượt là độ dài của sổ lịch sử của chuỗi nội sinh và ngoại sinh. Đáng chú ý, bất kỳ chuỗi nào cung cấp thông tin hữu ích cho việc dự báo chuỗi nội sinh đều có thể được sử dụng làm chuỗi ngoại sinh, bất kể độ dài lịch sử của chúng, do đó $T_{\text{ex}} \neq T$ là khả thi. Mục tiêu của mô hình dự báo \mathcal{F}_θ với tham số θ là dự đoán S bước thời gian tương lai $\hat{\mathbf{x}} = \{x_{T+1}, x_{T+2}, \dots, x_{T+S}\}$ dựa trên cả quan sát lịch sử $\mathbf{x}_{1:T}$ và các chuỗi ngoại sinh tương ứng $\mathbf{z}_{1:T_{\text{ex}}}$:

$$\hat{\mathbf{x}}_{T+1:T+S} = \mathcal{F}_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T_{\text{ex}}})$$



Hình 2.7: Minh họa mô hình TimeXer

Cấu trúc bao gồm các thành phần chính như sau:

- **(a) Mô-đun nhúng nội sinh:** Tạo ra nhiều token tạm thời (patch tokens) và một nhúng token toàn cục (global token) cho biến nội sinh.
- **(b) Mô-đun nhúng ngoại sinh:** Tạo ra một token cấp biến (variate-level token) cho mỗi biến ngoại sinh.
- **(c) Self-Attention:** Được áp dụng đồng thời trên các token tạm thời nội sinh và token toàn cục để nắm bắt các phụ thuộc cấp đoạn (patch-level dependencies).
- **(d) Cross-Attention:** Được áp dụng trên các biến nội sinh và ngoại sinh để tích hợp thông tin từ các nguồn bên ngoài.

Như minh họa trong Hình 4.3, mô hình TimeXer tái sử dụng kiến trúc Transformer gốc mà không sửa đổi bất kỳ thành phần nào. Các biến nội sinh và ngoại sinh được xử lý bởi các chiến lược nhúng khác nhau. TimeXer tận dụng:

- **Self-attention:** Để nắm bắt các phụ thuộc theo thời gian.
- **Cross-attention:** Để nắm bắt các phụ thuộc giữa các biến.

1. **Nhúng biến nội sinh:** Hầu hết các mô hình dự báo dựa trên Transformer hiện nay nhúng mỗi điểm thời gian hoặc một đoạn của chuỗi thời gian thành một token tạm thời và áp dụng chú ý tự thân để học các phụ thuộc thời gian. Để nắm bắt chi tiết các biến đổi thời gian trong biến nội sinh, TimeXer áp dụng biểu diễn theo đoạn (patch-wise representations). Cụ thể, chuỗi nội sinh được chia thành các đoạn không chồng lấn, mỗi đoạn được chiếu thành một token tạm thời. Do vai trò khác nhau của biến nội sinh và ngoại sinh trong dự báo, TimeXer nhúng chúng ở các mức độ chi tiết khác nhau. Việc kết hợp trực tiếp các token nội sinh và ngoại sinh ở các mức độ chi tiết khác nhau có thể dẫn đến sự không đồng nhất về thông tin. Để giải quyết vấn đề này, TimeXer giới thiệu một global token có thể học được cho mỗi biến nội sinh, đóng vai trò là biểu diễn vĩ mô để tương tác với các biến ngoại sinh. Thiết kế này giúp kết nối thông tin nhân quả từ các chuỗi ngoại sinh đến các đoạn tạm thời của biến nội sinh. Quá trình nhúng biến nội sinh được mô tả như sau:

$$\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N\} = \text{Patchify}(\mathbf{x}),$$

$$\mathbf{P}_{\text{en}} = \text{PatchEmbed}(\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N),$$

$$\mathbf{G}_{\text{en}} = \text{Learnable}(\mathbf{x}).$$

Trong đó:

- P : Độ dài của mỗi đoạn.
- $N = \lfloor \frac{T}{P} \rfloor$: Số lượng đoạn được chia từ chuỗi nội sinh.
- \mathbf{s}_i : Đoạn thứ i .
- $\text{PatchEmbed}(\cdot)$: Ánh xạ mỗi đoạn có độ dài P , sau khi cộng với vị trí nhúng, thành một vector D -chiều thông qua một projector tuyến tính có thể huấn luyện.
- \mathbf{P}_{en} : N nhúng token tạm thời cấp đoạn.

- \mathbf{G}_{en} : 1 nhúng token toàn cục cấp chuỗi.

Tổng cộng, N nhúng token tạm thời \mathbf{P}_{en} và 1 nhúng token toàn cục \mathbf{G}_{en} được đưa vào bộ mã hóa Transformer.

2. **Nhúng biến ngoại sinh:** Mục đích chính của các biến ngoại sinh là hỗ trợ dự báo chính xác cho các biến nội sinh. Các tương tác giữa các biến khác nhau có thể được nắm bắt tự nhiên hơn thông qua biểu diễn cấp biến (variate-level representations), thích ứng với các bất thường như giá trị thiếu, thời gian không đồng bộ, tần suất không khớp hoặc độ dài lịch sử khác nhau. Ngược lại, biểu diễn cấp đoạn quá chi tiết cho các biến ngoại sinh sẽ gây ra độ phức tạp tính toán đáng kể và đưa vào thông tin nhiễu không cần thiết. Do đó, TimeXer nhúng mỗi chuỗi ngoại sinh thành một token cấp biến cho toàn bộ chuỗi, được mô tả như sau:

$$\mathbf{V}_{\text{ex},i} = \text{VariateEmbed}(\mathbf{z}^{(i)}), \quad i \in \{1, \dots, C\}.$$

3. **Self-Attention của biến nội sinh:** TimeXer sử dụng sự hỗ trợ của token toàn cục, đóng vai trò như cầu nối giữa biến nội sinh và ngoại sinh. Cụ thể, token toàn cục hoạt động không đối xứng trong cross-attention:

- (a) **Patch-to-Global:** Token toàn cục chú ý đến các token tạm thời để tổng hợp thông tin cấp đoạn trên toàn bộ chuỗi.
- (b) **Global-to-Patch:** Mỗi token tạm thời chú ý đến token toàn cục để nhận các tương quan cấp biến.

Cơ chế này cung cấp một cái nhìn toàn diện về các phụ thuộc thời gian bên trong biến nội sinh và cải thiện tương tác với các biến ngoại sinh không đều. Quá trình self-attention nội sinh có thể được mô tả bằng các phương trình sau:

- **Patch-to-Patch:**

$$\bar{\mathbf{P}}_{\text{en}}^{l,1} = \text{LayerNorm} \left(\mathbf{P}_{\text{en}}^l + \text{Self-Attention} \left(\mathbf{P}_{\text{en}}^l \right) \right)$$

- **Global-to-Patch:**

$$\bar{\mathbf{P}}_{\text{en}}^{l,2} = \text{LayerNorm} \left(\mathbf{P}_{\text{en}}^l + \text{Cross-Attention} \left(\mathbf{P}_{\text{en}}^l, \mathbf{G}_{\text{en}}^l \right) \right)$$

- **Patch-to-Global:**

$$\bar{\mathbf{G}}_{\text{en}}^l = \text{LayerNorm} \left(\mathbf{G}_{\text{en}}^l + \text{Cross-Attention} \left(\mathbf{G}_{\text{en}}^l, \mathbf{P}_{\text{en}}^l \right) \right)$$

Quá trình tổng thể được đơn giản hóa thành một phép tính self-attention nội sinh:

$$\bar{\mathbf{P}}_{\text{en}}^l, \bar{\mathbf{G}}_{\text{en}}^l = \text{LayerNorm} \left(\left[\mathbf{P}_{\text{en}}^l, \mathbf{G}_{\text{en}}^l \right] + \text{Self-Attention} \left(\left[\mathbf{P}_{\text{en}}^l, \mathbf{G}_{\text{en}}^l \right] \right) \right)$$

Trong đó:

- $l \in \{0, \dots, L-1\}$ biểu thị khối TimeXer thứ l .
- $\mathbf{P}_{\text{en}}^0 = \mathbf{P}_{\text{en}}, \mathbf{G}_{\text{en}}^0 = \mathbf{G}_{\text{en}}$ là giá trị ban đầu.
- $[\cdot, \cdot]$ biểu thị sự nối (concatenation) của các token tạm thời và token toàn cục dọc theo chiều chuỗi.

Bằng cách áp dụng self-attention trên các token được nối $[\mathbf{P}_{\text{en}}^l, \mathbf{G}_{\text{en}}^l]$, TimeXer đồng thời nắm bắt được phụ thuộc thời gian giữa các đoạn (patches) và mối quan hệ giữa mỗi đoạn và toàn bộ chuỗi.

4. **Cross-Attention của biến nội sinh và biến ngoại sinh** Cross-Attention được sử dụng để tích hợp thông tin từ biến ngoại sinh vào biến nội sinh. Trong TimeXer, lớp cross-attention coi:

- Biến nội sinh là truy vấn (query).
- Biến ngoại sinh là khóa (key) và giá trị (value).

Vì biến ngoại sinh được nhúng thành các token cấp biến, TimeXer sử dụng token toàn cục của biến nội sinh để tổng hợp thông tin từ các biến ngoại sinh. Quá trình này được mô tả chính thức như sau:

$$\text{Variate-to-Global: } \bar{\mathbf{G}}_{\text{en}}^l = \text{LayerNorm} \left(\bar{\mathbf{G}}_{\text{en}}^l + \text{Cross-Attention} \left(\bar{\mathbf{G}}_{\text{en}}^l, \mathbf{V}_{\text{ex}} \right) \right)$$

Cơ chế này cho phép TimeXer xây dựng các kết nối thích ứng giữa biến nội sinh và ngoại sinh, cải thiện khả năng dự báo bằng cách tận dụng thông tin bên ngoài một cách hiệu quả. Cuối cùng tất cả các token tạm thời và token toàn cục có thể học được sẽ được biến đổi bởi lớp feedforward:

$$\mathbf{P}_{\text{en}}^{l+1} = \text{Feed-Forward}(\bar{\mathbf{P}}_{\text{en}}^l), \quad \mathbf{G}_{\text{en}}^{l+1} = \text{Feed-Forward}(\bar{\mathbf{G}}_{\text{en}}^l),$$

trong đó $l \in \{1, \dots, L\}$. Mỗi khối Transformer có thể được mô tả là $\text{TrmBlock}(\mathbf{P}_{\text{en}}^l, \mathbf{G}_{\text{en}}^l)$.

5. **Hàm mất mát dự báo:** Trong dự báo chuỗi thời gian với biến ngoại sinh, các biến ngoại sinh không cần được dự báo. Do đó, chúng ta tạo ra dự báo $\tilde{\mathbf{x}}$ bằng cách áp dụng một phép chiếu tuyến tính lên các nhúng đầu ra của biến nội sinh $[\mathbf{P}_{\text{en}}^L, \mathbf{G}_{\text{en}}^L]$, một thực hành phổ biến trong các mô hình dự báo chỉ sử dụng bộ mã hóa. Hàm mất mát bình phương (L2) được sử dụng để đo lường sự khác biệt giữa dự đoán và giá trị thực:

$$\text{Loss} = \sum_{i=1}^S \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2^2, \quad \text{trong đó} \quad \tilde{\mathbf{x}} = \text{Projection}([\mathbf{P}_{\text{en}}^L, \mathbf{G}_{\text{en}}^L]).$$

2.5 Phương pháp tối ưu hóa siêu tham số của mô hình

Tối ưu hóa đóng vai trò quan trọng trong việc đánh giá và so sánh hiệu quả của các mô hình. Bài toán tối ưu hóa tổng quát có dạng:

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$

trong đó $f : \mathcal{X} \rightarrow \mathbb{R}$ là hàm mục tiêu và $\mathcal{X} \subseteq \mathbb{R}^d$ là không gian tìm kiếm.

Trong đề án, em lựa chọn hai phương pháp là Gaussian Process Optimization và Gradient Boosted Trees Optimization.

2.5.1 Gaussian Process Optimization

Gaussian Process là một kỹ thuật học máy Bayesian dùng để mô hình hóa hàm mục tiêu trong Bayesian Optimization (BayesOpt).

Gaussian Process là một quá trình ngẫu nhiên mà bất kỳ tập hợp hữu hạn nào của các biến ngẫu nhiên tuân theo phân phối Gaussian chung, được xác định bởi hàm trung bình $m(\cdot)$ và hàm hiệp phương sai $k(\cdot, \cdot)$, thường gọi là kernel. Trong GPO, GP đóng vai trò là phân phối tiên nghiệm (prior) trên các hàm có thể đại diện cho mục tiêu, được cập nhật thành phân phối hậu nghiệm (posterior) khi các cặp điểm đầu vào-đầu ra (\mathbf{X}, \mathbf{y}) được quan sát. Phân phối hậu nghiệm cung cấp cả giá trị trung bình dự đoán và ước lượng độ không chắc chắn tại bất kỳ điểm nào trong không gian đầu vào, hỗ trợ cân bằng giữa thăm dò (lấy mẫu ở vùng có độ không chắc chắn cao) và khai thác (lấy mẫu ở vùng có hiệu suất dự đoán cao). Quá trình tối ưu hóa lặp lại việc chọn các điểm mới để đánh giá bằng cách tối đa hóa hàm thu nhận (acquisition function), định lượng giá trị của việc lấy mẫu tại một điểm. Các hàm thu nhận phổ biến bao gồm:

- Expected Improvement - EI: Đo lường cải tiến kỳ vọng so với quan sát tốt nhất hiện tại;
- Probability of Improvement - PI: Đánh giá xác suất cải thiện;
- Upper Confidence Bound - UCB: Cân bằng giữa dự đoán trung bình và độ bất định

Về mặt toán học, với tập hợp các điểm đã quan sát $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ và giá trị hàm tương ứng $\mathbf{y} = \{y_1, \dots, y_n\}$ trung bình hậu nghiệm $\mu(\mathbf{x}^*)$ và phương sai $\sigma^2(\mathbf{x}^*)$ tại một điểm mới \mathbf{x}^* được tính như sau:

$$\mu(\mathbf{x}^*) = m(\mathbf{x}^*) + \mathbf{k}(\mathbf{x}^*, \mathbf{X})[\mathbf{K} + \sigma^2 \mathbf{I}]^{-1}(\mathbf{y} - \mathbf{m}(\mathbf{X}))$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*, \mathbf{X})[\mathbf{K} + \sigma^2 \mathbf{I}]^{-1} \mathbf{k}(\mathbf{x}^*, \mathbf{X})^T$$

trong đó:

- $m(\cdot)$: Hàm trung bình tiên nghiệm, thường được đặt bằng 0 để đơn giản hóa,
- $k(\cdot, \cdot)$: Hàm hiệp phương sai, như kernel Matérn hoặc kernel hàm mũ bậc hai
- \mathbf{K} : Ma trận hiệp phương sai $n \times n$ của các điểm quan sát, với $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$
- σ^2 : Phương sai nhiễu, tính đến nhiễu trong quan sát,
- $\mathbf{m}(\mathbf{X})$: Vectơ các giá trị trung bình tiên nghiệm tại các điểm quan sát.

Các siêu tham số của kernel thường được tối ưu hóa bằng cách tối đa hóa xác suất log-tiên nghiệm (log-marginal-likelihood - LML), có thể cần nhiều lần khởi động lại để tránh tối ưu cục bộ.

2.5.2 Gradient Boosted Trees Optimization

Gradient Boosted Trees Optimization (GBTO) là một kỹ thuật học tập hợp xây dựng mô hình dự đoán bằng cách thêm tuần tự các mô hình học yếu—thường là cây quyết định—để giảm thiểu hàm mất mát được chỉ định. Phương pháp này tổng quát hóa các phương pháp boosting truyền thống bằng cách điều chỉnh các mô hình mới theo gradient âm của hàm mất mát so với dự đoán của mô hình hiện tại, cho phép tối ưu hóa các hàm mất mát khả vi bất kỳ.

GBTO hoạt động bằng cách xây dựng một mô hình cộng, trong đó mỗi mô hình học yếu sửa lỗi của tập hợp trước đó. Thuật toán khởi tạo với một mô hình hằng số, thường là trung bình của biến mục tiêu cho hồi quy hoặc log-odds cho phân loại, và tinh chỉnh mô hình qua các lần lặp. Ở mỗi lần lặp, một mô hình học yếu mới được điều chỉnh với các dư số giả (pseudo-residuals), là gradient âm của hàm mất mát tại dự đoán hiện tại. Đóng góp của mỗi mô hình mới được nhân với tốc độ học để kiểm soát bước nhảy, đảm bảo hội tụ ổn định.

Về mặt toán học, với tập dữ liệu $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, thuật toán GBTO xây dựng mô hình $F_m(\cdot)$ như sau:

Khởi tạo $F_0(\cdot) = \arg \min_c \sum_{i=1}^n L(y_i, c)$, trong đó L là hàm mất mát. Với mỗi lần lặp $m = 1$ đến M :

1. Tính dư số giả $r_{im} = -\frac{\partial L(y_i, F_{m-1}(\mathbf{x}_i))}{\partial F_{m-1}(\mathbf{x}_i)}$ cho mỗi mẫu i .
2. Điều chỉnh mô hình học yếu $h_m(\cdot)$, thường là cây quyết định, để dự đoán dư số giả r_{im} từ \mathbf{x}_i
3. Tính bước nhảy tối ưu $\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(\mathbf{x}_i) + \gamma h_m(\mathbf{x}_i))$
4. Cập nhật mô hình $F_m(\cdot) = F_{m-1}(\cdot) + \gamma_m h_m(\cdot)$.

Mô hình cuối cùng là tổng cộng: $F_M(\cdot) = F_0(\cdot) + \sum_{m=1}^M \gamma_m h_m(\cdot)$. Các hàm mất mát phổ biến bao gồm sai số bình phương trung bình cho hồi quy, mất mát logistic cho phân loại nhị phân, và mất mát entropy chéo cho phân loại đa lớp.

Chương 3

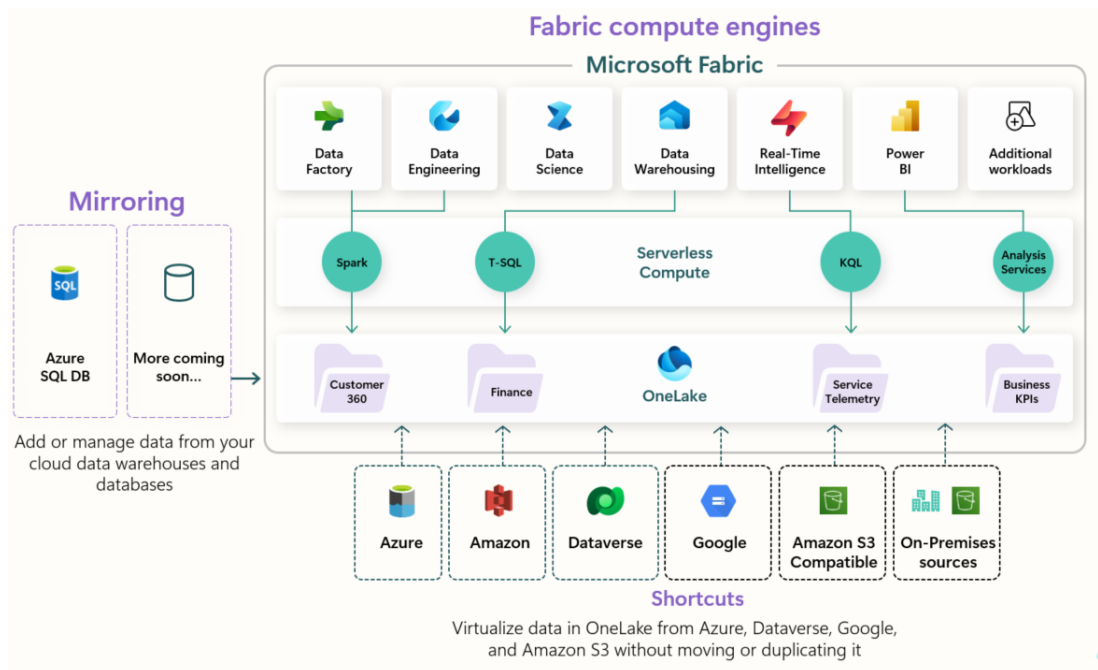
PHÂN TÍCH DỮ LIỆU CHỨNG KHOÁN VỚI MICROSOFT FABRIC

3.1 Giới thiệu nền tảng phân tích dữ liệu Microsoft Fabric

Microsoft Fabric là một nền tảng phân tích dữ liệu end-to-end, giúp hợp nhất tất cả các quy trình từ thu thập, di chuyển, lưu trữ, xử lý, biến đổi dữ liệu cho đến phân tích và trực quan hóa trong một hệ sinh thái thống nhất. Các thành phần chính của Microsoft Fabric:

- Power BI: Công cụ trực quan hóa dữ liệu
- Databases: Hệ thống cơ sở dữ liệu giao dịch, hỗ trợ Azure SQL Database và các công nghệ khác.
- Data Factory: Công cụ tích hợp dữ liệu, và xử lý đổi dữ liệu.

- Industry Solutions: Giải pháp chuyên biệt theo ngành, tối ưu hóa quản lý dữ liệu, phân tích và ra quyết định dựa trên đặc thù của từng lĩnh vực.
- Real-Time Intelligence: Hệ thống phân tích dữ liệu thời gian thực, cho phép xử lý luồng dữ liệu sự kiện, theo dõi, trực quan hóa và thực hiện hành động ngay lập tức.
- Data Engineering: Nền tảng Spark giúp thu thập, lưu trữ, xử lý và phân tích dữ liệu lớn.
- Data Science: Hỗ trợ xây dựng, triển khai và vận hành mô hình Machine Learning, tích hợp Azure Machine Learning.
- Data Warehouse – Hệ thống kho dữ liệu hiệu suất cao, tách biệt tài nguyên tính toán và lưu trữ, đồng thời hỗ trợ định dạng Delta Lake giúp tối ưu hóa truy vấn dữ liệu.



Hình 3.1: Fabric compute engines

Microsoft Fabric đơn giản hóa quy trình phát triển phân tích dữ liệu bằng cách hợp

nhất các công cụ vào một nền tảng SaaS thống nhất, cho phép các vai trò khác nhau trong tổ chức cộng tác hiệu quả mà không cần lặp lại công việc.

- **Data Engineer:** Thu thập, chuyển đổi và tải dữ liệu trực tiếp vào OneLake bằng Pipelines, tự động hóa quy trình làm việc và hỗ trợ lập lịch. Có thể lưu trữ dữ liệu trong lakehouse, sử dụng định dạng Delta-Parquet để lưu trữ và quản lý phiên bản hiệu quả. Ngoài ra, notebook cung cấp khả năng viết kịch bản nâng cao cho các chuyển đổi phức tạp.
- **Data Analyst:** Chuyển đổi data upstream bằng dataflows và kết nối trực tiếp với OneLake bằng chế độ Direct Lake; tạo báo cáo hiệu quả với PowerBI
- **Data Scientist:** Sử dụng notebooks tích hợp với hỗ trợ Python và Spark để xây dựng và kiểm tra các mô hình học máy. Dữ liệu được lưu trữ và truy cập trong lakehouse, và có thể tích hợp với Azure Machine Learning để triển khai và vận hành các mô hình.
- **Analytics engineers:** Cầu nối giữa data engineering and analysis bằng cách quản lý các tài sản dữ liệu trong lakehouse, đảm bảo chất lượng dữ liệu và hỗ trợ self-service analytics.
- **Low-to-no-code users and citizen developers:** Có thể khám phá các tập dữ liệu được quản lý thông qua OneLake Hub và sử dụng các mẫu Power BI để nhanh chóng tạo báo cáo và dashboard. Ngoài ra, họ có thể sử dụng dataflows để thực hiện các tác vụ ETL đơn giản mà không cần phụ thuộc vào data engineers.

3.2 Chủ điểm phân tích

3.2.1 Giới thiệu

Trong chương 3, thực nghiệm tập trung xây dựng một pipeline tự động hóa trong Microsoft Fabric để thu thập, lưu trữ, xử lý và trực quan hóa dữ liệu phục liên quan

đến bài toán dự báo được trình bày ở Chương 4. Mục tiêu là thu thập dữ liệu chỉ số VNindex và các biến ngoại sinh quan trọng như tỷ giá hối đoái VND/USD và chỉ số Dow Jones (DJI) nhằm phân tích insight và đánh giá mức độ tương quan của chúng đến VNindex.

Pipeline được xây dựng dựa trên các công cụ được tích hợp trong MS Fabric, bao gồm Jupyter Notebook để thu thập dữ liệu thông qua các thư viện Python vnstock và yfinance, lakehouse để lưu trữ dữ liệu thô, warehouse để lưu trữ dữ liệu đã xử lý, và trực quan hóa dữ liệu bằng Power BI. Lakehouse, với đặc điểm kết hợp giữa lưu trữ dữ liệu thô và khả năng xử lý linh hoạt, được sử dụng để lưu trữ dữ liệu ban đầu. Warehouse đảm bảo dữ liệu sau xử lý được tổ chức theo cấu trúc phù hợp cho phân tích nâng cao. Cuối cùng, dashboard được cập nhật tự động để phản ánh dữ liệu mới nhất, cung cấp cái nhìn tổng quan và chi tiết về các biến số.

3.2.2 Chủ điểm phân tích

Báo cáo tập trung phân tích mối quan hệ và tác động của tỷ giá VND/USD cùng với chỉ số Dow Jones tới chỉ số VNindex. Một số chủ điểm phân tích bao gồm:

1. Phân tích biến động của VNindex

- Biến động của VNindex qua thời gian, phân tích các giai đoạn tăng, giảm rõ rệt của chỉ số này.
- Phân tích % thay đổi hàng ngày của VNindex, để xác định mức độ biến động theo thời gian.

2. Phân tích mối tương quan giữa VNindex, tỷ giá VND/USD và chỉ số Dow Jones

- Tìm hiểu sự tương quan giữa VNindex với tỷ giá VND/USD và Dow Jones, nhằm đo lường mức độ gắn kết giữa các chỉ số.
- Kiểm tra sự thay đổi hệ số tương quan qua các giai đoạn thị trường

3. Phân tích mức độ biến động

- Biến động của tỷ giá VND/USD và chỉ số Dow Jones trong cùng khoảng thời gian, nhằm làm rõ sự đồng biến hay nghịch biến so với VNindex.
- So sánh mức độ biến động của VNindex so với Dow Jones và tỷ giá VND/USD, qua đó nhận diện các giai đoạn mà VNindex chịu tác động mạnh từ các yếu tố quốc tế.
- Đánh giá mức độ ổn định hay biến động của VNindex trong từng giai đoạn – đặc biệt khi Dow Jones có biến động lớn.

3.3 Quy trình thu thập, lưu trữ và xử lý dữ liệu

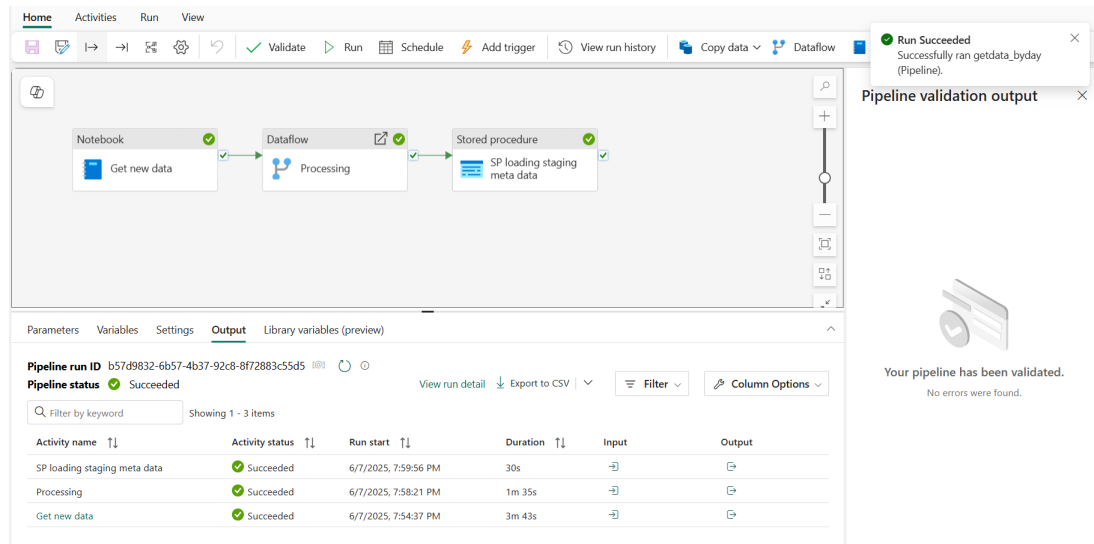
3.3.1 Thu thập và lưu trữ dữ liệu ban đầu

Quy trình thu thập và lưu trữ dữ liệu ban đầu được thực hiện trên nền tảng MS Fabric với các bước cụ thể sau:

- Sử dụng notebook(1) để thu thập dữ liệu lịch sử của chỉ số VNIndex, chỉ số Dow Jones, và tỷ giá VND/USD, với phạm vi thời gian từ ngày 1/5/2015 đến ngày hiện tại bằng thư viện vnstock và yfinance.
- Merge dữ liệu của ba chỉ số dựa trên cột 'date' của VNindex, chỉ giữ lại giá đóng cửa của mỗi chỉ số để đảm bảo tính đồng nhất. Thực hiện tiền xử lý để điền giá trị null bằng giá trị trước đó.
- Lưu trữ dữ liệu đã xử lý dưới dạng tệp 'stock_data.csv' trong Lakehouse, nơi đóng vai trò là kho dữ liệu thô ban đầu.
- Sao chép dữ liệu từ Lakehouse sang bảng 'dbo.stock_data_final' trong Warehouse. Đây là bảng dữ liệu để phục vụ cho việc xây dựng dashboard trên Power BI, đảm bảo dữ liệu đã được làm sạch và tối ưu hóa.

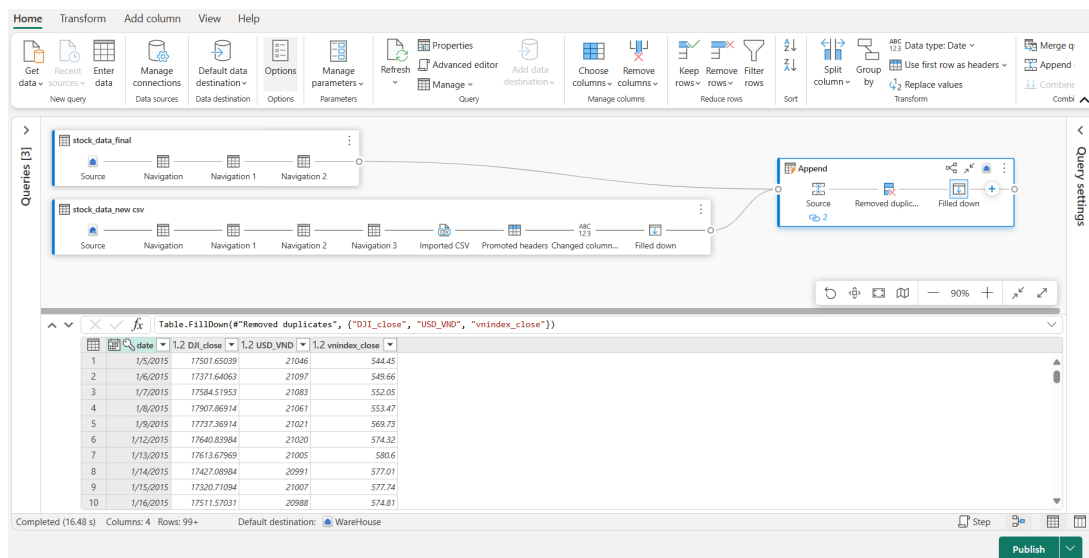
3.3.2 Xây dựng Pipeline và lịch trình thu thập dữ liệu mới

Quy trình xây dựng pipeline và lịch trình thu thập dữ liệu mới được triển khai để đảm bảo dữ liệu được cập nhật liên tục, với các bước chi tiết như sau:



Hình 3.2: Pipeline thu thập dữ liệu mới

- Sử dụng notebook(2) để thu thập dữ liệu của 5 ngày giao dịch gần nhất, tính đến ngày chạy pipeline.
- Thu thập dữ liệu của VNIndex, Dow Jones, và tỷ giá VND/USD thông qua các thư viện và xử lý hợp nhất dữ liệu tương tự notebook(1), giữ nguyên cấu trúc cột như trong tệp 'stock_data.csv'.
- Lưu trữ dữ liệu mới dưới dạng tệp 'stock_data_new.csv' để đảm bảo tính đồng nhất với dữ liệu lịch sử.
- Sử dụng Dataflow trong Microsoft Fabric để xử lý dữ liệu mới với các bước cụ thể:
 - Ghép nối (append) dữ liệu từ tệp 'stock_data_new.csv' vào bảng 'dbo.stock_data_final' trong Warehouse.



Hình 3.3: Dataflow xử lý dữ liệu mới cập nhật

- Loại bỏ các bản ghi trùng lặp dựa trên cột 'date' để tránh dữ liệu lặp lại.
- Xử lý các giá trị null bằng phương pháp forward fill (ffill) để giải quyết sự khác biệt trong lịch hoạt động thị trường và ngày lễ giữa các chỉ số.
- Cập nhật kết quả cuối cùng vào bảng 'dbo.stock_data_final' trong Warehouse.
- Làm mới (refresh) dashboard trên Power BI để các biểu đồ và phân tích phản ánh dữ liệu thời gian thực.

5. Lưu lại để theo dõi và kiểm soát thông tin chạy của pipeline bằng cách sử dụng bảng 'metadata.processing_log' được định nghĩa với cấu trúc sau:

- Cột 'pipeline_run_id' (kiểu VARCHAR(255)) để lưu mã định danh của lần chạy pipeline.
- Cột 'rows_processed' (kiểu INT) để ghi lại số lượng hàng đã được xử lý.
- Cột 'latest_date' (kiểu DATETIME2(6)) để lưu ngày mới nhất của dữ liệu được xử lý.

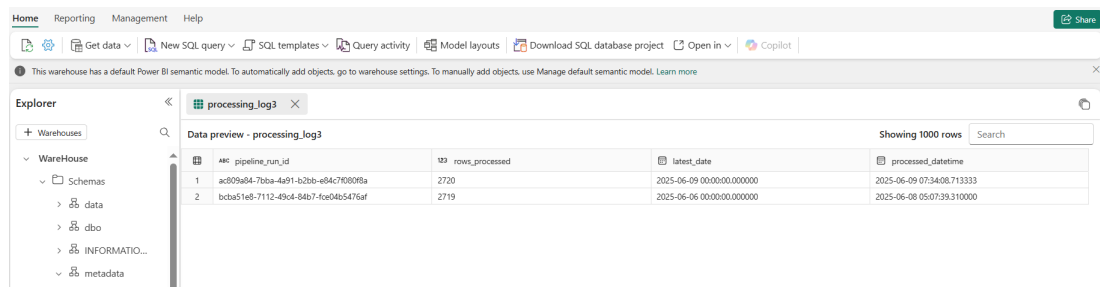
- Cột 'processed_datetime' (kiểu DATETIME2(6)) để ghi thời điểm xử lý dữ liệu.

6. Thực hiện lưu thông tin này thông qua procedure 'metadata.insert_staging_metadata', với tham số đầu vào:

- '@pipeline_run_id' (kiểu VARCHAR(255)) để cung cấp mã định danh lần chạy.
- '@processed_date' (kiểu DATETIME) để cung cấp thời điểm xử lý.

Thủ tục này thực hiện chèn dữ liệu vào bảng 'metadata.processing_log' dựa trên truy vấn SQL sau:

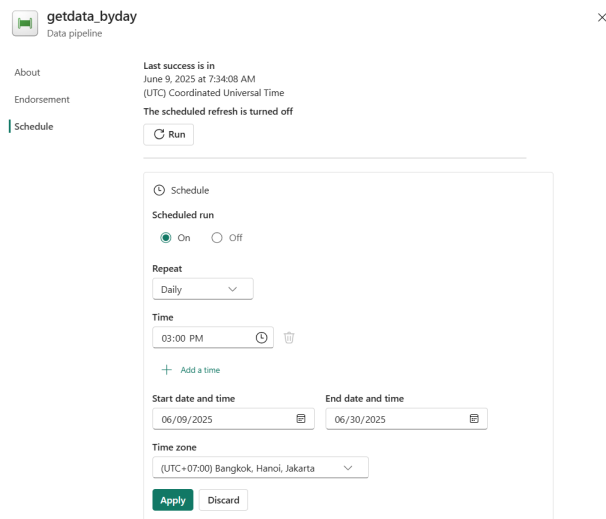
- Lấy '@pipeline_run_id' làm giá trị cho cột 'pipeline_run_id'.
- Đếm tổng số hàng (Count(*)) từ bảng 'dbo.stock_data_final' để tính 'rows_processed'.
- Sử dụng '@processed_date' làm giá trị cho cột 'processed_datetime'.



	pipeline_run_id	rows_processed	latest_date	processed_datetime
1	ac809a84-7bba-4a91-b2bb-e84c7f080f8a	2720	2025-06-09 00:00:00.000000	2025-06-09 07:34:06.713333
2	bcbaf51e8-7112-49c4-84b7-fce0465476af	2719	2025-06-06 00:00:00.000000	2025-06-08 05:07:39.310000

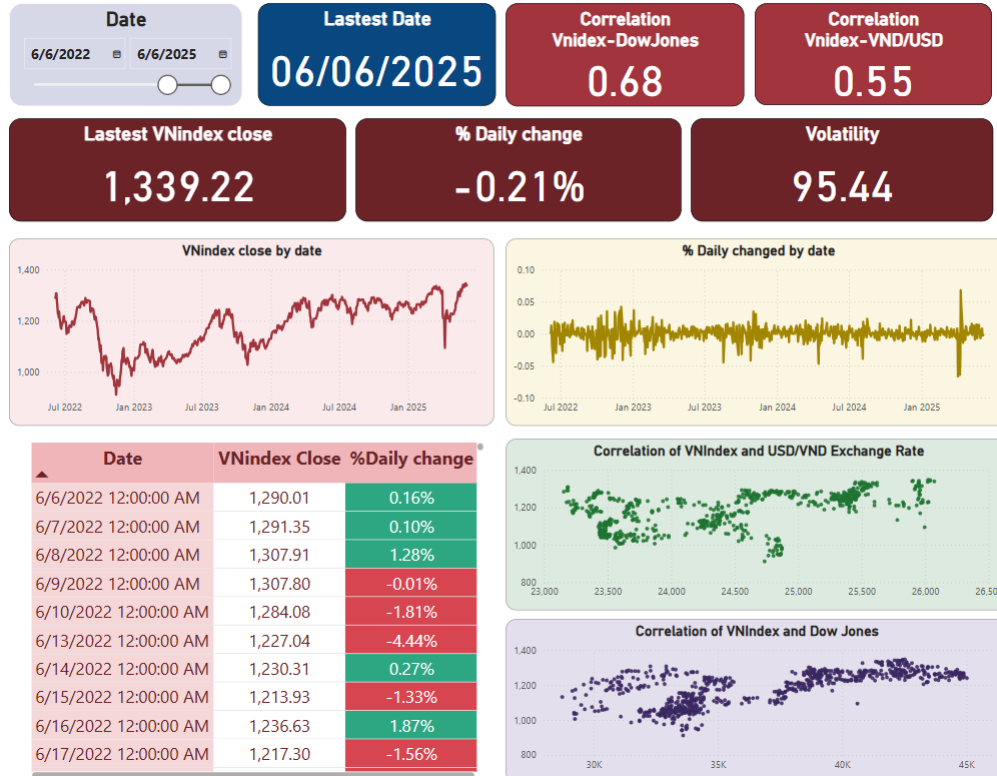
Hình 3.4: Thông tin hoạt động của pipeline được theo dõi

7. Cuối cùng schedule cho pipeline chạy tự động



Hình 3.5: Đặt lịch trình cho pipeline chạy hàng ngày

3.4 Xây dựng báo cáo



Hình 3.6: Dashboard phân tích tổng quan (6/6/2022-6/6/2025)

1. Phân tích biến động của VNindex

• Phân tích tổng quan

Trong giai đoạn từ 06/2022 đến 06/2025, biểu đồ đường (line chart) đã thể hiện rõ biến động của VNindex. Chỉ số VNindex trải qua các giai đoạn biến động tương đối phức tạp với các đợt tăng và giảm luân phiên:

- Giai đoạn tăng: Từ giữa năm 2022 đến đầu năm 2023, VNindex có xu hướng tăng nhẹ, phản ánh sự phục hồi nhất định của thị trường sau đại dịch COVID-19. Từ giữa năm 2024, chỉ số này tiếp tục có những đợt tăng, đồng thời cũng xuất hiện các đợt điều chỉnh ngắn hạn.
- Giai đoạn giảm: Đầu năm 2023 và cuối năm 2024 chứng kiến các đợt

điều chỉnh mạnh, VNindex giảm khá sâu, cho thấy sự phản ứng nhạy cảm của thị trường trước các biến động toàn cầu.

Biểu đồ % Daily change cho thấy mức độ biến động hàng ngày của VNindex là đáng kể, dao động từ -1% đến hơn 1% tại các thời điểm khác nhau. Trong giai đoạn tháng 4/2025, biến động mạnh với biên độ thay đổi từ -6% đến 6%, tình trạng này xảy ra liên quan đến quyết định áp thuế mới của tổng thống Mỹ Donald Trump.

Xem xét mức độ biến động (Volatility), đây là độ lệch chuẩn (standard deviation) của chỉ số VNindex trong toàn bộ giai đoạn phân tích. Nó đo lường mức độ biến động của VNindex – cho biết chỉ số này dao động nhiều hay ít quanh giá trị trung bình của chính nó. Kết quả trong giai đoạn 2022-2025 đạt 95.44, phản ánh VNindex có mức biến động khá cao trong giai đoạn này.

- **Phân tích dữ liệu mới nhất**

Dữ liệu hiện tại được cập nhật đến ngày 6/6/2025 với giá đóng cửa là 1339.22, giảm 0.21% so với ngày trước đó (5/6/2025), đây là một mức giảm nhẹ. Tuy nhiên khi xem xét bảng số liệu, kết quả cho thấy VNindex đã trải qua 3 phiên giảm liên tiếp, đây là hiện tượng cần lưu ý.

2. Phân tích mối tương quan giữa VNindex và các biến ngoại sinh

Để đánh giá mức độ gắn kết giữa VNIndex, tỷ giá VND/USD và chỉ số Dow Jones, hệ số tương quan Pearson được tính toán dựa trên dữ liệu hàng ngày. Công thức tính hệ số tương quan giữa hai biến X và Y là:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}}$$

trong đó \bar{X} và \bar{Y} lần lượt là giá trị trung bình của VNIndex và một trong hai biến còn lại (tỷ giá hoặc Dow Jones). Kết quả cho thấy hệ số tương quan giữa VNIndex và Dow Jones dao động từ 0,4 đến 0,7 trong các giai đoạn thị trường ổn định, nhưng giảm xuống dưới 0,3 trong các giai đoạn biến động mạnh, như

năm 2022. Mỗi tương quan giữa VNIndex và tỷ giá VND/USD dao động mạnh, cho thấy tác động của tỷ giá đến VNIndex không rõ rệt trong ngắn hạn. Kết quả cho thấy sự tương quan giữa VNIndex và Dow Jones mạnh hơn trong giai đoạn tăng trưởng, trong khi tỷ giá VND/USD có tác động lớn hơn trong giai đoạn suy thoái, do áp lực từ các yếu tố kinh tế vĩ mô. Đối với giai đoạn 2022-2025, hệ số tương quan 0.68 cho thấy VNIndex có mối tương quan thuận tương đối chặt với Dow Jones. Khi Dow Jones tăng, VNIndex có xu hướng tăng theo, phản ánh sự gắn kết của thị trường Việt Nam với biến động kinh tế Mỹ. Bên cạnh đó, hệ số tương quan giữa VNIndex và tỷ giá VND/USD là 0.55 phản ánh mối tương quan thuận vừa phải.

Khi biểu diễn mức độ tương quan bằng biểu đồ scatter: Các điểm dữ liệu tập trung thành một dải, cho thấy mối quan hệ tuyến tính thuận tương đối giữa VNIndex và 2 biến ngoại sinh, dù tồn tại sự phân tán nhất định, các điểm vẫn có xu hướng dịch chuyển cùng chiều. Biến ngoại sinh tỷ giá tăng thường có thể ảnh hưởng đến chi phí nhập khẩu hoặc triển vọng xuất khẩu, từ đó tác động đến lợi nhuận doanh nghiệp và thị trường chứng khoán. Trong khi đó biến ngoại sinh Dow Jones tác động đến thị trường Việt Nam do sự phản ứng nhanh chóng của tâm lý nhà đầu tư, dòng vốn nước ngoài và các thông tin vĩ mô quốc tế.

3. Phân tích Mức độ Biến động

Biểu đồ “VNIndex close & Dow Jones close” cho thấy mối tương quan thuận rõ rệt. Hai chỉ số có xu hướng đồng biến, đặc biệt trong các giai đoạn phục hồi kinh tế toàn cầu. Hệ số tương quan đạt 0.68 phản ánh mối liên kết khá chặt chẽ. Khi Dow Jones tăng, VNIndex có khuynh hướng tăng theo, cho thấy tính nhạy cảm của thị trường Việt Nam với biến động kinh tế Mỹ. Đặc biệt, trong các giai đoạn Dow Jones biến động mạnh, VNIndex thường phản ứng chậm hơn, với độ trễ khoảng 1-2 ngày. Điều này có thể được giải thích bởi sự khác biệt về múi giờ và tâm lý nhà đầu tư trong nước.

Đối với tỷ giá USD/VND, biểu đồ “VNIndex close & VND/USD” phản ánh mối



Hình 3.7: Dashboard phân tích mức độ biến động (6/6/2022-6/6/2025)

tương quan thuận nhưng ở mức độ vừa phải (hệ số tương quan 0.55). Tỷ giá có ảnh hưởng nhất định đến VNIndex thông qua các yếu tố chi phí nhập khẩu và dòng vốn nước ngoài, song không mạnh bằng ảnh hưởng của chỉ số Dow Jones. Xem xét sự biến động theo ngày của 3 chỉ số, các biến động mạnh của VNIndex thường xuất hiện sau hoặc cùng thời điểm với biến động mạnh của Dow Jones, phản ánh tính “phụ thuộc” của VNIndex vào các tín hiệu quốc tế, đặc biệt là Mỹ. Biến động tỷ giá USD/VND nhỏ hơn, nhưng các thay đổi nhỏ vẫn có thể lan tỏa ảnh hưởng đến VNIndex, nhất là các doanh nghiệp phụ thuộc vào nhập khẩu – xuất khẩu.

Chương 4

ỨNG DỤNG MÔ HÌNH HỌC SÂU TRONG BÀI TOÁN DỰ BÁO CHỈ SỐ VNINDEX

4.1 Phát biểu bài toán

Bài toán được đặt ra là xây dựng mô hình dự báo chỉ số VNindex – là chỉ số thị trường chứng khoán đại diện cho Sở Giao dịch Chứng khoán TP. Hồ Chí Minh (HOSE). Các mô hình sẽ sử dụng dữ liệu lịch sử giá đóng cửa (close) của chỉ số VNindex để dự báo giá đóng cửa của VNindex trong tương lai. Dữ liệu lịch sử sẽ được xử lý và đưa vào bốn mô hình học sâu đã được đề cập ở chương 2, gồm: LSTM (Long Short-Term Memory), TimeMixer, TSMixer và TimeXer, các mô hình được cung cấp bởi thư viện `neuralforecast` – một công cụ mạnh mẽ cho các tác vụ dự báo tài chính và kinh tế. Mỗi mô hình sẽ được huấn luyện trên cùng một tập dữ liệu cơ sở, đảm bảo tính đồng nhất và công bằng trong quá trình so sánh.

Bài toán đặt ra theo 2 hướng tiếp cận. Đầu tiên, các mô hình chỉ sử dụng duy nhất dữ liệu giá đóng cửa của VNindex để dự báo. Đây là cách tiếp cận truyền thống, tập

trung khai thác thông tin nội sinh từ chính chỉ số VNindex nhằm học được quy luật tiềm ẩn của thị trường. Ngoài ra, trong đồ án có tiếp cận một hướng mới đó là sử dụng mô hình TimeXer với các biến ngoại sinh, bao gồm tỷ giá VND/USD và giá đóng cửa của chỉ số DowJones – đại diện cho tình hình kinh tế vĩ mô quốc tế. Việc đưa các biến ngoại sinh vào mô hình TimeXer cho phép đánh giá mức độ cải thiện khả năng dự báo của mô hình khi tận dụng thông tin ngoài thị trường chứng khoán Việt Nam. Đây là điểm khác biệt nổi bật so với ba mô hình còn lại và là trọng tâm phân tích của bài toán. Mục tiêu cuối cùng của bài toán là chứng minh tính ưu việt của TimeXer khi có thể tích hợp và khai thác hiệu quả các yếu tố ngoại sinh, giúp cải thiện chất lượng dự báo so với các mô hình chỉ sử dụng dữ liệu nội sinh.

4.2 Khám phá dữ liệu

Để thu thập tập dữ liệu phục vụ cho quá trình huấn luyện và đánh giá các mô hình dự báo, đồ án sử dụng hai thư viện nguồn mở: vnstock3 và yfinance.

- Thư viện vnstock3 là một công cụ tiện ích mạnh mẽ được thiết kế riêng cho việc truy xuất dữ liệu chứng khoán Việt Nam, bao gồm giá cổ phiếu, chỉ số thị trường, thông tin cơ bản doanh nghiệp và các dữ liệu lịch sử khác. Đây là một nguồn dữ liệu đáng tin cậy và phổ biến trong cộng đồng nghiên cứu tài chính tại Việt Nam, giúp tiết kiệm đáng kể thời gian và công sức trong khâu thu thập dữ liệu.
- Thư viện yfinance là thư viện Python phổ biến trong việc truy xuất dữ liệu tài chính quốc tế, bao gồm cổ phiếu, chỉ số thị trường (như DowJones), tỷ giá hối đoái và nhiều loại tài sản tài chính khác. Thư viện này kết nối trực tiếp với API của Yahoo Finance, cho phép người dùng dễ dàng lấy dữ liệu lịch sử và dữ liệu thời gian thực.

Các bộ dữ liệu được thu thập được lấy trong khoảng thời gian từ ngày 05/01/2015 đến

ngày 02/04/2025, đảm bảo có thể cung cấp cái nhìn tổng thể về lịch sử biến động, đồng thời độ dài đủ lớn để phục vụ cho việc huấn luyện các mô hình học sâu.

4.2.1 Mô tả dữ liệu

1. Dữ liệu VNindex

	time	open	high	low	close	volume
2	2015-01-05	544.86	549.22	543.78	544.45	91834620
3	2015-01-06	539.08	550.11	538.82	549.66	94081890
4	2015-01-07	548.44	555.83	548.44	552.05	109445780
5	2015-01-08	553.49	556.80	552.15	553.47	73883040
6	2015-01-09	553.49	570.52	552.15	569.73	104203680
...
2745	2025-03-27	1325.98	1328.82	1323.01	1323.81	522142993
2746	2025-03-28	1324.42	1325.34	1315.72	1317.46	601970197
2747	2025-03-31	1313.51	1314.09	1304.10	1306.86	706353460
2748	2025-04-01	1313.03	1317.46	1308.06	1317.33	516351102
2749	2025-04-02	1321.53	1324.46	1317.68	1317.83	668744030

2556 rows × 6 columns

Hình 4.1: Dữ liệu chỉ số VNindex

Bộ dữ liệu lịch sử của chỉ số VNindex 2556 dòng bao gồm các cột thông tin:

- 'time': Thời gian giao dịch, thể hiện theo dạng YYYY-MM-DD.
- 'open': Giá mở cửa của chỉ số trong ngày.
- 'high': Giá cao nhất của chỉ số trong ngày.
- 'low': Giá thấp nhất của chỉ số trong ngày.
- 'close': Giá đóng cửa của chỉ số trong ngày.
- 'volume': Tổng số cổ phiếu được giao dịch trong ngày.

Các giá trị được ghi nhận hàng ngày, trừ các ngày lễ, cuối tuần khi thị trường không hoạt động.

2. Dữ liệu biến ngoại sinh

Sử dụng thư viện yfinance để thu thập dữ liệu lịch sử tỷ giá hối đoái giữa USD và VND (Ticker 'USDVND=X').

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2015-01-05 00:00:00+00:00	21219.0	21375.0	21046.0	21046.0	0	0.0	0.0
2015-01-06 00:00:00+00:00	20987.0	21390.0	20987.0	21097.0	0	0.0	0.0
2015-01-07 00:00:00+00:00	21135.0	21415.0	21083.0	21083.0	0	0.0	0.0
2015-01-08 00:00:00+00:00	21074.0	21360.0	21061.0	21061.0	0	0.0	0.0
2015-01-09 00:00:00+00:00	21008.0	21335.0	21008.0	21021.0	0	0.0	0.0
...
2025-03-26 00:00:00+00:00	25600.0	25630.0	25550.0	25600.0	0	0.0	0.0
2025-03-27 00:00:00+00:00	25555.0	25610.0	25550.0	25555.0	0	0.0	0.0
2025-03-28 00:00:00+00:00	25560.0	25580.0	25540.0	25560.0	0	0.0	0.0
2025-03-31 00:00:00+01:00	25550.0	25575.0	25530.0	25550.0	0	0.0	0.0
2025-04-01 00:00:00+01:00	25565.0	25640.0	25565.0	25565.0	0	0.0	0.0

2670 rows × 7 columns

Hình 4.2: Dữ liệu biến ngoại sinh tỷ giá hối đoái VND/USD

Sử dụng thư viện yfinance để thu thập dữ liệu lịch sử chỉ số DowJones

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2015-01-05 00:00:00-05:00	17821.300781	17821.300781	17475.929688	17501.650391	116160000	0.0	0.0
2015-01-06 00:00:00-05:00	17504.179688	17581.050781	17262.369141	17371.640625	101870000	0.0	0.0
2015-01-07 00:00:00-05:00	17374.779297	17597.080078	17374.779297	17584.519531	91030000	0.0	0.0
2015-01-08 00:00:00-05:00	17591.970703	17916.039062	17591.970703	17907.869141	114890000	0.0	0.0
2015-01-09 00:00:00-05:00	17911.019531	17915.320312	17686.089844	17737.369141	93390000	0.0	0.0
...
2025-03-26 00:00:00-04:00	42655.851562	42821.828125	42326.671875	42454.789062	592650000	0.0	0.0
2025-03-27 00:00:00-04:00	42432.960938	42523.839844	42142.191406	42299.699219	484540000	0.0	0.0
2025-03-28 00:00:00-04:00	42245.820312	42258.148438	41530.000000	41583.898438	532360000	0.0	0.0
2025-03-31 00:00:00-04:00	41382.519531	42147.378906	41148.128906	42001.761719	732220000	0.0	0.0
2025-04-01 00:00:00-04:00	41879.750000	42140.660156	41519.898438	41989.960938	514610000	0.0	0.0

2576 rows × 7 columns

Hình 4.3: Dữ liệu biến ngoại sinh DowJones

Các cột 'Open' 'High' 'Low' 'Close' có cùng ý nghĩa với bộ dữ liệu Vnindex, ngoài ra các cột còn lại:

- 'Dividends': Cổ tức được trả (thường = 0 cho chỉ số).
- 'Stock Splits': Thông tin chia tách cổ phiếu (thường = 0 cho chỉ số). Tuy nhiên, trong bài toán đang xét sẽ không sử dụng các cột này.

Có thể thấy cả 3 bộ dữ liệu trong cùng khoảng thời gian (05-01-2015 đến 02-04-2025) nhưng lại không trùng khớp về số dòng dữ liệu, điều này do khác biệt thị trường và lịch nghỉ lễ.

4.2.2 Chuẩn bị dữ liệu

Các bộ dữ liệu được thu thập còn độc lập, bao gồm các cột chưa được chuẩn hóa để đưa vào mô hình, cũng như còn chứa các giá trị null, vì vậy cần làm sạch trước khi đưa vào mô hình. Để sử dụng thông tin tỷ giá hối đoái VND/USD và chỉ số DowJones làm biến ngoại sinh cho bài toán, cần thực hiện thao tác gộp ba bộ dữ liệu thành một

bảng dữ liệu duy nhất dựa trên cột date của chỉ số VNINDEX. Việc gộp dữ liệu được thực hiện bằng cách sử dụng phép nối (merge) theo cột date, với mục đích lấy toàn bộ ngày giao dịch của VNINDEX làm chuẩn. Sau khi gộp, chỉ giữ lại bốn cột: date và ba cột giá trị đóng cửa của VNINDEX, DowJones và tỷ giá USD/VND.

	date	vnindex_close	DJI_close	USD/VND
0	2015-01-05	544.45	17501.650391	21046.0
1	2015-01-06	549.66	17371.640625	21097.0
2	2015-01-07	552.05	17584.519531	21083.0
3	2015-01-08	553.47	17907.869141	21061.0
4	2015-01-09	569.73	17737.369141	21021.0
...
2552	2025-03-27	1323.81	42299.699219	25555.0
2553	2025-03-28	1317.46	41583.898438	25560.0
2554	2025-03-31	1306.86	42001.761719	25550.0
2555	2025-04-01	1317.33	41989.960938	25565.0
2556	2025-04-02	1317.83	41989.960938	25565.0

2557 rows × 4 columns

Hình 4.4: Gộp các file dữ liệu và xử lý null

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2556 entries, 0 to 2555
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   date             2556 non-null   datetime64[ns]
1   vnindex_close    2556 non-null   float64
2   DJI_close        2556 non-null   float64
3   USD/VND          2556 non-null   float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 80.0 KB
```

Hình 4.5: Thông tin về dữ liệu sau khi gộp

Các mô hình học sâu được xây dựng cho bài toán dự báo yêu cầu tập dữ liệu đầu vào phải chứa các cột thông tin:

- 'ds': Thời gian quan sát
- 'y': Giá trị quan sát (biến mục tiêu)
- 'unique_id': Cột chứa giá trị phân biệt các chuỗi thời gian được sử dụng. Với bài toán cụ thể trên, có thể đặt giá trị của cột là 'VNINDEX' theo tên biến cần dự đoán.

	unique_id	ds	y	USD/VND	DJI_close
0	VNINDEX	2015-01-05	544.45	21046.0	17501.650391
1	VNINDEX	2015-01-06	549.66	21097.0	17371.640625
2	VNINDEX	2015-01-07	552.05	21083.0	17584.519531
3	VNINDEX	2015-01-08	553.47	21061.0	17907.869141
4	VNINDEX	2015-01-09	569.73	21021.0	17737.369141
...
2551	VNINDEX	2025-03-27	1323.81	25555.0	42299.699219
2552	VNINDEX	2025-03-28	1317.46	25560.0	41583.898438
2553	VNINDEX	2025-03-31	1306.86	25550.0	42001.761719
2554	VNINDEX	2025-04-01	1317.33	25565.0	41989.960938
2555	VNINDEX	2025-04-02	1317.83	25565.0	41989.960938

2556 rows × 5 columns

Hình 4.6: Dữ liệu được chỉnh sửa phù hợp yêu cầu của các mô hình

Cuối cùng, dữ liệu được chia thành ba tập riêng biệt: tập huấn luyện (train), tập xác thực (validation), và tập kiểm tra (test). Tỷ lệ chia được xác định theo tỷ lệ 7:2:1, lần lượt cho ba tập train, validation và test. Phương pháp chia giữ nguyên thứ tự thời gian, tránh làm mất tính chất tuần tự vốn có của dữ liệu chuỗi thời gian.

4.3 Xây dựng mô hình LSTM dự báo chỉ số VNindex

Mô hình LSTM (Long Short-Term Memory) trong bài toán dự báo 20 bước chỉ số VNindex được triển khai qua thư viện NeuralForecast. Quy trình xây dựng mô hình bao gồm ba giai đoạn chính:

1. Tối ưu hóa siêu tham số: Áp dụng phương pháp Bayes Optimization (Gaussian Process và Gradient Boosted Trees) để tìm kiếm siêu tham số tối ưu trong một không gian tham số đa chiều.

Parameter	Value
input_size	[40; 80]
max_steps	[100; 500]
encoder_n_layers	[1; 3]
batch_size	{ 32; 64 }
decoder_layers	[1; 3]
learning_rate	[1e-5; 1e-3]

Bảng 4.1: Không gian tìm kiếm siêu tham số tối ưu cho mô hình LSTM

Mô hình sử dụng hàm mất mát được sử dụng là DistributionLoss với phân phối chuẩn (Normal) và các mức độ tin cậy 80% và 90%, kết hợp với kiểu chuẩn hóa **robust** để xử lý dữ liệu có biến động lớn. Hàm mục tiêu trong quá trình tối ưu hóa được thiết kế để huấn luyện mô hình LSTM trên tập dữ liệu huấn luyện, sử dụng kỹ thuật cross-validation với chân trời dự báo là 20 ngày và bước trượt bằng chân trời dự báo. Để đảm bảo tính hợp lệ của cross-validation, kích thước tập kiểm tra được điều chỉnh sao cho thỏa mãn điều kiện **(test_size - h) % step_size = 0**, trong đó **h** là bước dự báo và **step_size** là bước trượt.

Mỗi thuật toán thực hiện 25 lần gọi hàm mục tiêu, với 10 điểm khởi tạo ngẫu nhiên, và được đánh giá dựa trên sai số tuyệt đối trung bình (MAE) trên tập xác

thực. Kết quả của mỗi lần gọi được lưu trữ để theo dõi lịch sử tối ưu hóa, và cấu hình tốt nhất được chọn dựa trên MAE thấp nhất.

2. **Huấn luyện mô hình với siêu tham số tối ưu:** Sau khi xác định cấu hình tối ưu, mô hình LSTM cuối cùng được huấn luyện trên toàn bộ tập dữ liệu huấn luyện với các siêu tham số tốt nhất.
3. **Đánh giá hiệu suất thông qua kỹ thuật cross-validation:** Thực hiện cross-validation để đánh giá hiệu suất trên tập kiểm tra, sử dụng các chỉ số sai số tuyệt đối trung bình (MAE), sai số bình phương trung bình (RMSE), và sai số phần trăm tuyệt đối trung bình (MAPE).

Kết quả

Phương pháp	MAE tốt nhất	Số lần gọi
Gaussian Process	24.182323	25
Gradient Boosted Trees	23.673719	25

Bảng 4.2: Kết quả tối ưu hóa siêu tham số mô hình LSTM

Parameter	Value
input_size	41
max_steps	464
encoder_n_layers	2
batch_size	64
decoder_layers	2
learning_rate	0.00010968217207529519

Bảng 4.3: Siêu tham số tối ưu cho mô hình LSTM

4.4 Xây dựng mô hình TimeMixer và TSMixer dự báo chỉ số VNindex

Tương tự mô hình LSTM, hai mô hình TimeMixer và TSMixer trong bài toán dự báo 20 bước chỉ số VNindex cũng được triển khai qua thư viện NeuralForecast. Quy trình xây dựng mô hình bao gồm ba giai đoạn chính:

1. **Tối ưu hóa siêu tham số:** Hai mô hình đều áp dụng thuật toán Bayes Optimization: Gaussian Process (GP) và Gradient Boosted Regression Trees (GBRT). Mỗi thuật toán thực hiện 50 lần gọi hàm mục tiêu, với 10 điểm khởi tạo ngẫu nhiên. Hàm mục tiêu được thiết kế để huấn luyện mô hình trên tập dữ liệu huấn luyện, sử dụng kỹ thuật cross-validation với bước dự báo 20 ngày. Sai số MAE được tính toán dựa trên giá trị dự báo và giá trị thực tế trên các cửa sổ kiểm tra. Kết quả của mỗi lần gọi hàm được lưu trữ để theo dõi lịch sử tối ưu hóa, và cấu hình tốt nhất được chọn dựa trên MAE thấp nhất.

- **Mô hình TimeMixer**

Parameter	Value
input_size	[40; 80]
max_steps	[200; 500]
learning_rate	[1e-4; 1e-3]
batch_size	{32; 64}
d_ff	{256; 512}
e_layers	[1; 3]
drop_output	[0; 0.3]

Bảng 4.4: Không gian tìm kiếm siêu tham số cho mô hình TimeMixer

- **Mô hình TSMixer**

Parameter	Value
input_size	[40; 80]
max_steps	[200; 500]
learning_rate	[1e-4; 1e-3]
batch_size	{32; 64}
ff_dim	{32; 64}
drop_output	[0; 0.3]

Bảng 4.5: Không gian tìm kiếm siêu tham số cho mô hình TSMixer

2. **Huấn luyện mô hình với siêu tham số tối ưu** Sau khi xác định siêu tham số tối ưu từ quá trình tối ưu hóa Bayes, 2 mô hình được huấn luyện với các tham số tốt nhất và sử dụng hàm mất mát MAE. Quá trình huấn luyện sử dụng cơ chế early stopping với tham số early_stop_patience_steps là 5, giảm hiện tượng overfit và đảm bảo hiệu suất tối ưu trên tập dữ liệu huấn luyện.
3. **Đánh giá hiệu suất thông qua kỹ thuật cross-validation** Các chỉ số đánh giá bao gồm sai số tuyệt đối trung bình (MAE), sai số bình phương trung bình gốc (RMSE), và sai số phần trăm tuyệt đối trung bình (MAPE), được tính toán dựa trên giá trị dự báo và giá trị thực tế trên tập kiểm tra.

Kết quả:

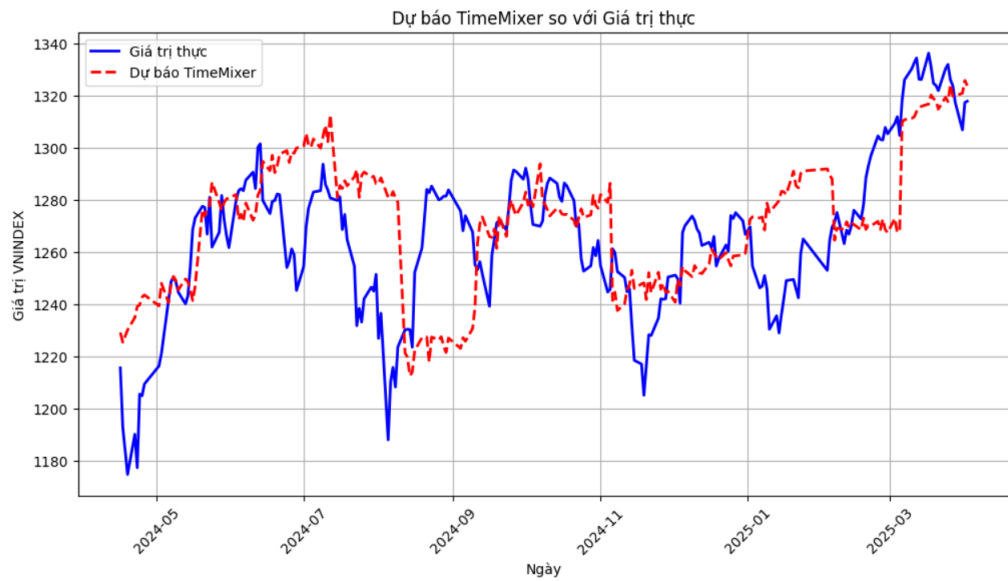
- **Mô hình TimeMixer**

Phương pháp	MAE tốt nhất	Số lần gọi
Gaussian Process	21.773080	25
Gradient Boosted Trees	22.996324	25

Bảng 4.6: Kết quả tối ưu hóa siêu tham số mô hình TimeMixer

Parameter	Value
input_size	72
max_steps	255
learning_rate	0.0006021310185147606
batch_size	64
d_ff	256
e_layers	2
drop_output	0.13777466758976017

Bảng 4.7: Siêu tham số tối ưu cho mô hình TimeMixer



Hình 4.7: Mô hình TimeMixer dự báo 20 bước

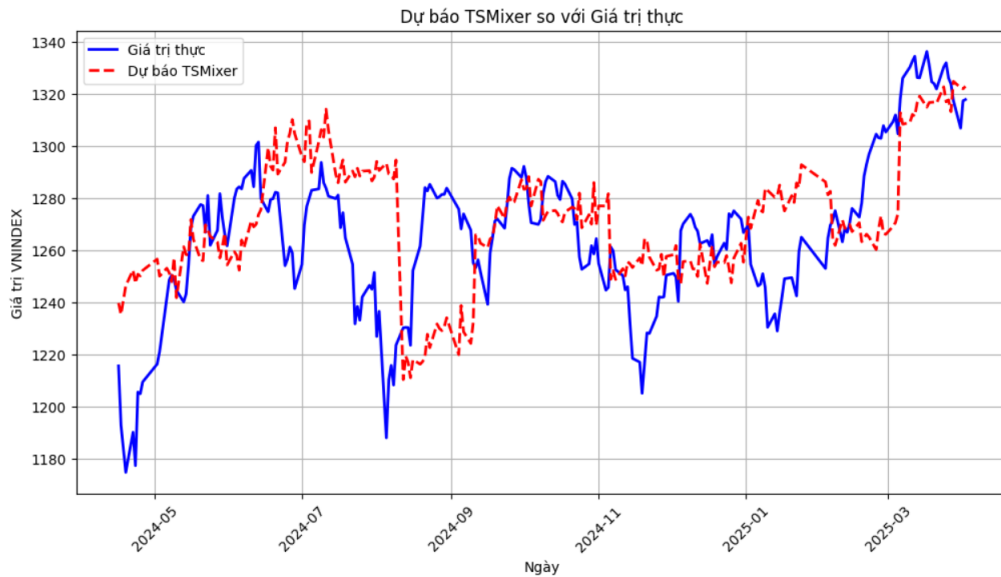
• Mô hình TSMixer

Phương pháp	MAE tốt nhất	Số lần gọi
Gaussian Process	24.793581	50
Gradient Boosted Trees	23.395328	50

Bảng 4.8: Kết quả tối ưu hóa siêu tham số mô hình TSMixer

Parameter	Value
input_size	41
max_steps	473
learning_rate	0.00045975057847321674
batch_size	32
ff_dim	32
drop_output	0.15602040635334327

Bảng 4.9: Siêu tham số tối ưu cho mô hình TSMixer



Hình 4.8: Mô hình TSMixer dự báo 20 bước

4.5 Xây dựng mô hình TimeXer dự báo chỉ số VNindex

Mô hình TimeXer được triển khai qua thư viện NeuralForecast và áp dụng với các kịch bản thực nghiệm:

- Dự báo 1 bước với input là VNindex Close
- Dự báo 20 bước với input là VNindex Close
- Dự báo 1 bước với input là VNindex Close kết hợp với biến ngoại sinh tỷ giá hối đoái VND/USD và DowJones Close
- Dự báo 20 bước với input là VNindex Close kết hợp với biến ngoại sinh tỷ giá hối đoái VND/USD và DowJones Close.

Với quy trình tương tự các mô hình đã trình bày ở trên, các bài toán đều gồm 3 giai đoạn:

1. **Tối ưu hóa siêu tham số:** Trong mỗi kịch bản thực nghiệm đều áp dụng thuật toán Bayes Optimization: Gaussian Process (GP) và Gradient Boosted Regres-

sion Trees (GBRT). Hàm mục tiêu được thiết kế để huấn luyện mô hình trên tập dữ liệu huấn luyện, sử dụng kỹ thuật cross-validation. Mỗi thuật toán thực hiện 50 lần gọi hàm mục tiêu, với 10 điểm khởi tạo ngẫu nhiên. Các kết quả được theo dõi và lựa chọn tốt nhất dựa trên giá trị MAE.

Parameter	Value (dự báo 1 bước)	Value (dự báo 20 bước)
input_size	[4; 36]	[40; 80]
hidden_size	{128, 256}	{128, 256}
n_heads	{4, 8, 16}	{8, 16}
e_layers	[1; 3]	[1; 3]
d_ff	{256, 512}	{256, 512}
dropout	[0; 0.3]	[0; 0.3]
learning_rate	[1e-5; 1e-3]	[1e-5; 1e-3]
batch_size	{32, 64, 128}	{32, 64, 128}
max_steps	[500; 1000]	[500; 1000]
scaler_type	{standard, robust, minmax}	{standard, robust, minmax}

Bảng 4.10: Không gian tìm kiếm siêu tham số cho mô hình TimeXer

- 2. Huấn luyện mô hình với siêu tham số tối ưu:** Sau khi xác định siêu tham số tối ưu từ quá trình trên, mô hình TimeXer được huấn luyện trên tập dữ liệu huấn luyện, sử dụng hàm mất mát MAE. Quá trình huấn luyện sử dụng cơ chế early stopping với độ kiên nhẫn 5 bước, nhằm ngăn chặn hiện tượng overfit và tối ưu hóa hiệu suất trên tập dữ liệu huấn luyện.

Với các kịch bản kết hợp biến ngoại sinh, mô hình sử dụng tham số **futr_exog_list** là 2 chuỗi: tỷ giá VND/USD và chỉ số DowJones.

- 3. Đánh giá hiệu suất thông qua kỹ thuật cross-validation:** Hiệu suất của mô hình được đánh giá thông qua kỹ thuật cross-validation trên tập kiểm tra. Các chỉ số đánh giá bao gồm sai số tuyệt đối trung bình (MAE), sai số bình phương

trung bình gốc (RMSE), và sai số phần trăm tuyệt đối trung bình (MAPE).

4.5.1 Mô hình TimeXer đơn biến

Kết quả tối ưu hóa và đánh giá mô hình:

Phương pháp	MAE tốt nhất	Số lần gọi
Gaussian Process	7.291542	50
Gradient Boosted Trees	7.151627	50

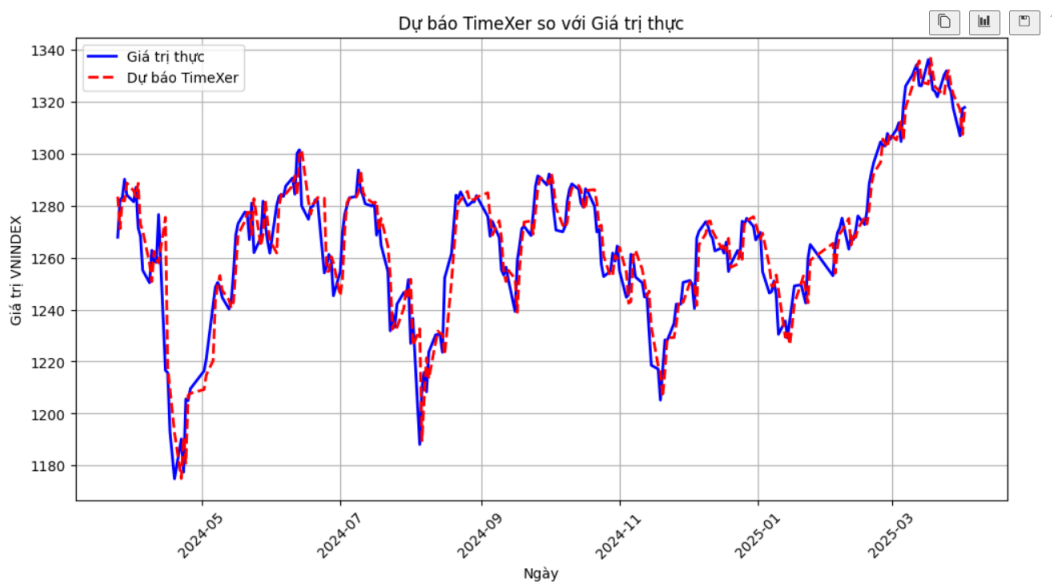
Bảng 4.11: Kết quả tối ưu hóa siêu tham số mô hình TimeXer đơn biến (dự báo 1 bước)

Phương pháp	MAE tốt nhất	Số lần gọi
Gaussian Process	21.409954	50
Gradient Boosted Trees	21.268049	50

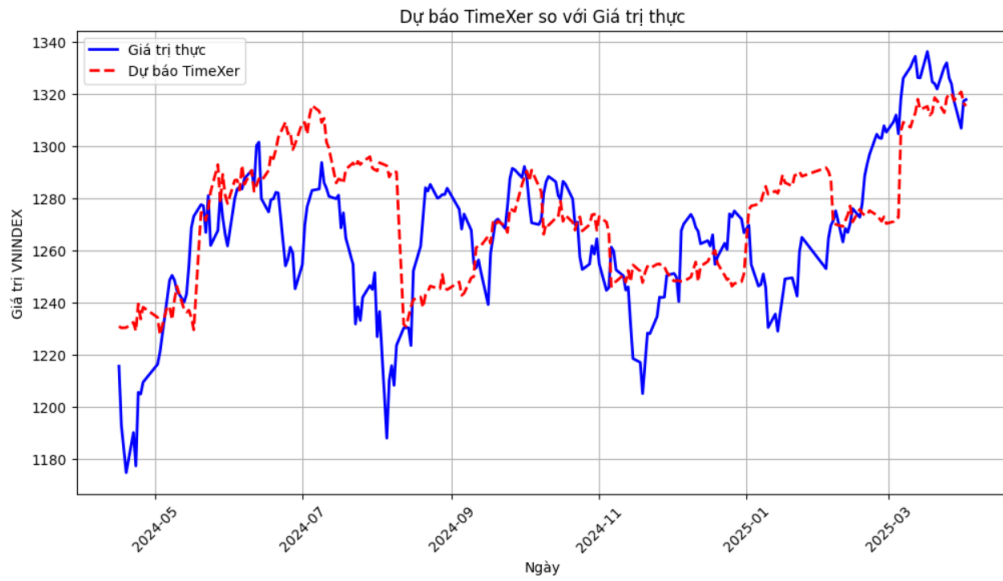
Bảng 4.12: Kết quả tối ưu hóa siêu tham số mô hình TimeXer đơn biến (dự báo 20 bước)

Parameter	Value (dự báo 1 bước)	Value (dự báo 20 bước)
input_size	16	45
hidden_size	256	256
n_heads	16	16
e_layers	2	2
d_ff	256	512
dropout	0.0467983561008608	0.14813867890931726
learning_rate	1.3066739238053285e-05	0.00011103647313054626
batch_size	128	64
max_steps	801	513
scaler_type	minmax	standard

Bảng 4.13: Siêu tham số tối ưu cho mô hình TimeXer



Hình 4.9: Mô hình TimeXer đơn biến dự báo 1 bước



Hình 4.10: Mô hình TimeXer đơn biến dự báo 20 bước

4.5.2 Mô hình TimeXer đa biến cho dự báo kết hợp một số biến ngoại sinh

Phương pháp	MAE tốt nhất	Số lần gọi
Gaussian Process	7.44751	50
Gradient Boosted Trees	7.115563	50

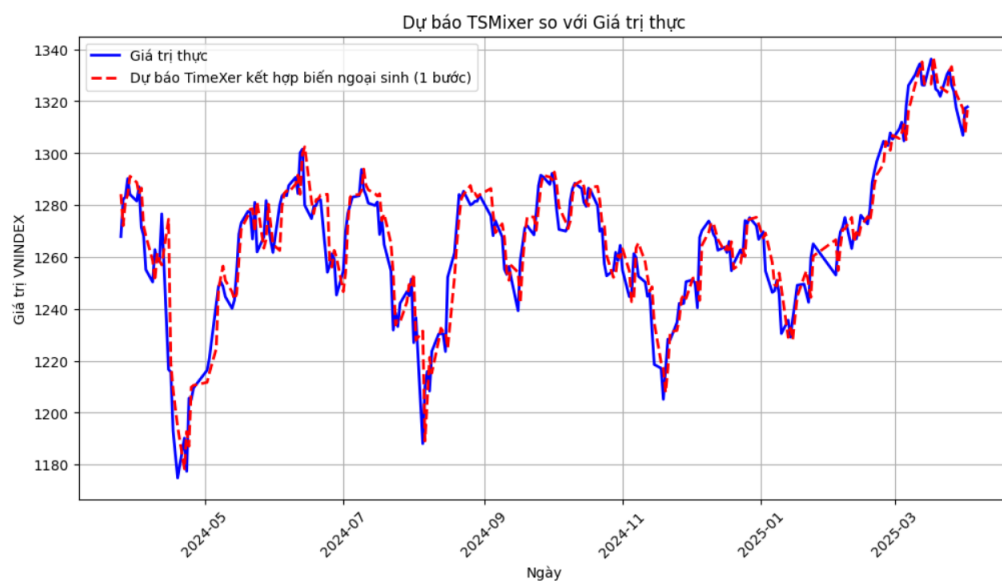
Bảng 4.14: Kết quả tối ưu hóa siêu tham số mô hình TimeXer kết hợp biến ngoại sinh (dự báo 1 bước)

Phương pháp	MAE tốt nhất	Số lần gọi
Gaussian Process	20.231151	50
Gradient Boosted Trees	19.889278	50

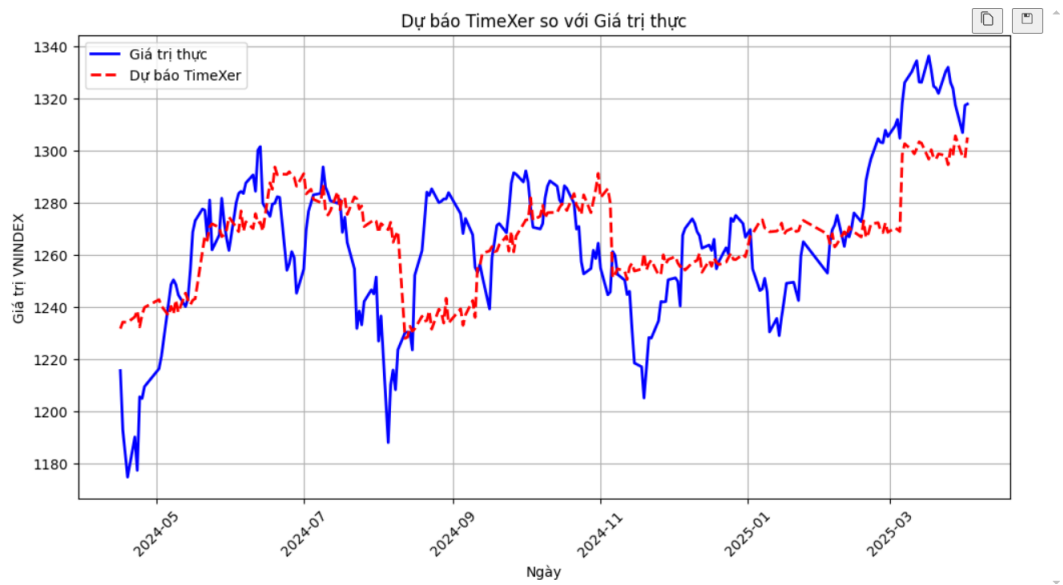
Bảng 4.15: Kết quả tối ưu hóa siêu tham số mô hình TimeXer kết hợp biến ngoại sinh (dự báo 20 bước)

Parameter	Value (dự báo 1 bước)	Value (dự báo 20 bước)
input_size	16	64
hidden_size	256	128
n_heads	16	8
e_layers	2	3
d_ff	256	512
dropout	0.0467983561008608	0.2425192044349384
learning_rate	1.3066739238053285e-05	4.066563313514792e-05
batch_size	64	32
max_steps	801	842
scaler_type	minmax	robust
futr_exog_list	['DJI_close']	['VND/USD', 'DJI_close']

Bảng 4.16: Siêu tham số tối ưu cho mô hình TimeXer kết hợp biến ngoại sinh



Hình 4.11: Mô hình TimeXer dự báo 1 bước kết hợp biến ngoại sinh



Hình 4.12: Mô hình TimeXer dự báo 20 bước kết hợp biến ngoại sinh

Chương 5

KẾT QUẢ CHẠY KIỂM THỬ MÔ HÌNH

5.1 So sánh kết quả của các mô hình

5.1.1 So sánh kết quả mô hình LSTM đơn biến và TimeXer đơn biến

Model	Horizon	Features used	MAE	RMSE	MAPE
LSTM đơn biến	20	Close	23.673719	31.572596	1.890625%
TimeXer đơn biến	20	Close	21.268049	27.641067	1.69447 %

Bảng 5.1: So sánh kết quả mô hình LSTM đơn biến- TimeXer đơn biến

Bảng 5.1 đánh giá hiệu suất dự báo chỉ số VNindex giữa mô hình LSTM và TimeXer đơn biến với horizon 20. Hai mô hình chỉ sử dụng dữ liệu lịch sử Close làm thông tin đầu vào.

- Với LSTM, MAE đạt 23.673719, trong khi TimeXer đạt 21.268049. Sự khác biệt khoảng 2.405670 cho thấy TimeXer có sai số tuyệt đối trung bình thấp hơn, tức là dự báo của TimeXer sát với giá trị thực tế hơn so với LSTM.
- TimeXer ghi nhận RMSE 27.641067, thấp hơn so với LSTM (31.572596). Giá trị RMSE thấp hơn của TimeXer cho thấy mô hình này không chỉ có sai số trung bình thấp mà còn kiểm soát tốt các sai số lớn, phản ánh tính ổn định cao hơn trong dự báo so với LSTM.
- TimeXer đạt MAPE 1.69447%, thấp hơn so với LSTM (1.890625%). MAPE thấp hơn của TimeXer cho thấy mô hình này có khả năng dự báo chính xác hơn về mặt tỷ lệ, đặc biệt quan trọng khi dự báo chỉ số VNINDEX với các biến động nhỏ.

5.1.2 So sánh kết quả mô hình TimeMixer, TSMixer và TimeXer

Model	Horizon	Features used	MAE	RMSE	MAPE
TimeMixer đơn biến	20	Close	21.54721	27.890251	1.715207%
TSMixer đơn biến	20	Close	23.395328	30.067503	1.864225%
TimeXer đơn biến	20	Close	21.268049	27.641067	1.69447 %

Bảng 5.2: So sánh kết quả mô hình TimeMixer đơn biến- TSMixer đơn biến- TimeXer đơn biến

Bảng 5.2 so sánh hiệu suất dự báo của các mô hình "mixing-based" TimeMixer, TSMixer và TimeXer trong kịch bản thực nghiệm dự báo 20 bước.

- TimeMixer đạt MAE là 21.54721, TSMixer là 23.395328, và TimeXer là 21.268049.

TimeXer có sai số tuyệt đối trung bình thấp nhất, với sự khác biệt lần lượt là 0.279161 so với TimeMixer và 2.127279 so với TSMixer. Điều này cho thấy TimeXer dự báo sát với giá trị thực tế hơn, trong khi TSMixer có hiệu suất kém nhất, có thể do thiếu sự linh hoạt trong việc xử lý các xu hướng dài hạn.

- TimeXer lại cho thấy RMSE thấp nhất (27.641067), với chênh lệch khoảng 0.249184 so với TimeMixer và 2.426436 so với TSMixer. Kết quả này cho thấy tính ổn định của TimeXer trong việc kiểm soát các sai số lớn, trong khi TSMixer thể hiện sự kém hiệu quả hơn trong việc giảm thiểu các sai lệch đáng kể.
- TimeMixer đạt MAPE là 1.71520%, thấp nhất trong 3 mô hình, với sự chênh lệch nhẹ khoảng 0.020737% so với TimeMixer và 0.169755% so với TSMixer. Điều này cho thấy TimeXer có độ chính xác tương đối cao hơn.

5.1.3 So sánh kết quả mô hình TimeXer đơn biến và mô hình TimeXer kết hợp biến ngoại sinh

Model	Horizon	Features used	MAE	RMSE	MAPE
TimeXer đơn biến	1	Close	7.151627	10.020534	0.569343%
TimeXer đa biến	1	Close, DowJones	7.115563	9.843223	0.566413 %

Bảng 5.3: So sánh kết quả mô hình TimeXer đơn biến- TimeXer có kết hợp biến ngoại sinh (h=1)

Model	Horizon	Features used	MAE	RMSE	MAPE
TimeXer đơn biến	20	Close	21.268049	27.641067	1.69447 %
TimeXer đa biến	20	Close VND/USD, DowJones	19.889278	24.876727	1.577967 %

Bảng 5.4: So sánh kết quả mô hình TimeXer đơn biến- TimeXer có kết hợp biến ngoại sinh (h=20)

Bảng 5.3 và bảng 5.4 so sánh hiệu suất dự báo của mô hình TimeXer khi chỉ sử dụng biến nội sinh dự báo chính nó so với mô hình TimeXer có kết hợp thêm biến ngoại sinh.

- Với horizon 1 ngày, TimeXer đơn biến (chỉ dùng biến nội sinh "Close") đạt MAE là 7.151627, trong khi TimeXer kết hợp biến ngoại sinh là chỉ số DowJones đạt 7.115563, giảm nhẹ khoảng 0.036064, cho thấy việc thêm biến "DowJones" cải thiện độ chính xác dự báo ngắn hạn. Với horizon 20 ngày, TimeXer đơn biến đạt 21.268049, trong khi TimeXer có thêm 2 biến ngoại sinh VND/USD, DowJones đạt 19.889278, giảm khoảng 1.378771, phản ánh lợi ích rõ rệt của các biến ngoại sinh trong dự báo dài hạn.
- Với horizon 1 ngày, TimeXer đơn biến đạt RMSE là 10.020534, trong khi TimeXer đa biến đạt 9.843223, giảm khoảng 0.177311, cho thấy sự cải thiện trong việc kiểm soát sai số lớn khi thêm "DowJones". Với horizon 20 ngày, TimeXer đơn biến đạt 27.641067, trong khi TimeXer đa biến đạt 24.876727, giảm khoảng 2.764340, củng cố tính ổn định của mô hình đa biến trong dự báo dài hạn.
- Với horizon 1 ngày, TimeXer đơn biến đạt MAPE là 0.569343%, trong khi TimeXer đa biến đạt 0.566413%, giảm nhẹ so với mô hình đơn biến, cho thấy cải thiện nhỏ trong độ chính xác tương đối. Với horizon 20 ngày, TimeXer đơn

biến đạt 1.69447%, trong khi TimeXer đa biến có MAPE thấp hơn khoảng 0.114503%, phản ánh lợi ích của biến ngoại sinh trong giảm sai số tương đối ở dự báo dài hạn.

5.2 Đánh giá và kết luận

Bốn mô hình học sâu đã được triển khai, bao gồm LSTM, TimeMixer, TSMixer và TimeXer, trong đó TimeXer được nhấn mạnh nhờ khả năng tích hợp các biến ngoại sinh tỷ giá hối đoái VND/USD và chỉ số DowJones. Dựa trên các kết quả thực nghiệm tốt nhất đã thực hiện trong đồ án, có thể đưa ra các nhận định sau:

1. Mô hình TimeXer hiệu quả hơn so với mô hình học sâu truyền thống LSTM:

Mô hình TimeXer tận dụng kiến trúc Transformer để học các mối quan hệ phức tạp, không tuyến tính giữa các yếu tố ảnh hưởng đến VNIndex, vượt trội hơn so với mô hình học sâu truyền thống như LSTM. Khả năng này giúp TimeXer nắm bắt tốt hơn các đặc trưng tiềm ẩn trong dữ liệu thị trường chứng khoán.

2. Hiệu quả vượt trội của cross-attention trong mô hình TimeXer so với các mô hình Transformer khác như TimeMixer và TSMixer: TimeXer đạt độ chính xác dự báo cao hơn nhờ cơ chế cross-attention, cho phép tích hợp hiệu quả giữa biến nội sinh (giá đóng cửa VNIndex) và các biến ngoại sinh.

3. Khả năng khai thác thông tin của biến ngoại sinh của mô hình TimeXer giúp cải thiện dự báo: Việc bổ sung các biến ngoại sinh (tỷ giá VND/USD và chỉ số DowJones) vào mô hình TimeXer mang lại sự cải thiện trong chất lượng dự báo so với trường hợp chỉ sử dụng biến nội sinh, đặc biệt trong dự báo dài hạn (20 ngày). Điều này chứng minh tiềm năng của TimeXer trong việc khai thác thông tin kinh tế vĩ mô bên ngoài thị trường chứng khoán Việt Nam.

4. Sai số phần trăm tuyệt đối trung bình (MAPE) của các kịch bản thực nghiệm đạt <1% : TimeXer thể hiện hiệu suất dự báo vượt trội với chỉ số

MAPE đạt khoảng 0.57% trong dự báo 1 bước và 1.58% trong dự báo dài hạn 20 bước so với giá trị thực tế.

5. **Áp dụng thành công mô hình TimeXer cho dữ liệu chứng khoán:** Đồ án đã áp dụng thành công mô hình TimeXer cho bài toán dự báo chỉ số VNIndex, mở rộng phạm vi ứng dụng của mô hình này so với bài báo gốc giới thiệu TimeXer, vốn chưa thực nghiệm trên dữ liệu chứng khoán. Điều này đánh dấu một đóng góp mới trong việc kiểm chứng hiệu quả của TimeXer trên các tập dữ liệu thực tế.

Những kết luận trên khẳng định tính ưu việt của mô hình TimeXer trong dự báo chỉ số VNIndex, đặc biệt khi tích hợp các biến ngoại sinh. Kết quả này không chỉ đóng góp vào lĩnh vực dự báo tài chính mà còn mở ra hướng nghiên cứu mới trong việc áp dụng các mô hình Transformer tiên tiến cho các bài toán kinh tế và tài chính phức tạp.

KẾT LUẬN

Đồ án đã đạt được mục tiêu đề ra

Kết quả của đồ án

Đồ án đã trình bày được các khái niệm, tính chất của bài toán chuỗi thời gian và cũng như xây dựng được mô hình dự báo chuỗi thời gian cho giá cổ phiếu trong thị trường chứng khoán và đưa ra những nhận định quan trọng

Cụ thể:

1. Đồ án đã hệ thống hóa các kiến thức nền tảng về chuỗi thời gian và bài toán dự báo chuỗi thời gian. Quá trình nghiên cứu làm quen các mô hình học sâu, đặc biệt tập trung vào các mô hình có cấu trúc "mixing-based" mới được công bố những năm gần đây.
2. Nghiên cứu lý thuyết các mô hình học sâu trong dự báo chuỗi thời gian, tập trung đặc biệt vào mô hình TimeXer mới được công bố có khả năng khai thác hiệu quả thông tin từ các biến ngoại sinh. Việc này bao gồm phân tích kiến trúc của TimeXer, so sánh với mô hình mạng nơ-ron như LSTM và mô hình cùng cấu trúc "mixing-based" nhưng chưa tích hợp biến ngoại sinh như TimeMixer và TSMixer.
3. Tìm hiểu và áp dụng các công cụ được tích hợp trong nền tảng SaaS Microsoft Fabric, nhằm thu thập, lưu trữ, xử lý và phân tích dữ liệu. Xây dựng thành công data pipeline cập nhật dữ liệu tự động hàng ngày.

4. Xây dựng nhiều kịch bản thử nghiệm nhằm phân tích hiệu suất của mô hình từ các góc độ khác nhau, bao gồm cấu hình mô hình, và việc tích hợp các biến ngoại sinh. Các thử nghiệm này giải quyết bài toán dự báo chỉ số thị trường VNindex ngắn hạn và trung hạn, được thiết kế để kiểm tra khả năng tổng quát hóa và độ nhạy của mô hình trước các thay đổi trong dữ liệu hoặc cấu hình.
5. Đưa ra 5 nhận định ý nghĩa từ các kết quả thực nghiệm. Những nhận định này có ý nghĩa thực tiễn cao, góp phần định hướng cho việc áp dụng các mô hình vào dự báo thị trường chứng khoán. Đồng thời, đây cũng là nền tảng quan trọng để phát triển các nghiên cứu tiếp theo trong cùng lĩnh vực.

Kỹ năng đạt được

1. Thực hiện tra cứu và tổng hợp các tài liệu chuyên ngành liên quan tới đề tài
2. Biết tổng hợp các kiến thức đã học và kiến thức trong tài liệu tham khảo để viết báo cáo đồ án.
3. Chế bản đồ án bằng LATEX , viết chương trình mô hình mô phỏng cho ví dụ minh họa bằng sử dụng ngôn ngữ PYTHON.

Hướng phát triển của đồ án trong tương lai

1. Tiếp tục cải tiến và thử nghiệm thêm nhiều mô hình khác nhau, tối ưu hóa các tham số bằng phương pháp hiệu quả hơn để nâng cao độ chính xác của dự báo.
2. Nghiên cứu thêm các yếu tố như: máy học trong việc nghiên cứu tâm lý đám đông, các chính sách của Chính phủ, các vấn đề khác trên thế giới ảnh hưởng đến thị trường chứng khoán của Việt Nam hiện nay.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] David R.Anderson, Dennis J.Sweeney, Thomas A.Williams, *Thống kê trong Kinh tế và Kinh doanh*, Nhà xuất bản Kinh tế, Tp Hồ Chí Minh.
- [2] Bạch Đức Hiển, *Giáo trình thị trường chứng khoán*, Nhà xuất bản Tài Chính, 2008

Tiếng Anh

- [3] Ratnadip Adhikari & R.K.Agrawal *An Introductory Study on Time Series Modeling and Forecasting* . Axioms,2013.
- [4] Hochreiter, Sepp & Schmidhuber, Jürgen. *Long Short-term Memory*, 1997
- [5] Wang, Shiyu & Wu, Haixu & Shi, Xiaoming & Hu, Tengge & Luo, Huakun & Ma, Lintao & Zhang, James & Zhou, Jun. *TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting*, arXiv, 2024.
- [6] Yang, C., Li, H., Ma, Y., Huang, Y., & Chu, X., *Enhanced TSMixer Model for the Prediction and Control of Particulate Matter*, Sustainability, 2025.
- [7] Chen, S.-A., Li, C.-L., Yoder, N., Arik, S., & Pfister, T., *TSMixer: An all-MLP Architecture for Time Series Forecasting*, arXiv, 2023.

-
- [8] Wang, Y., Wu, H., Dong, J., Qin, G., Zhang, H., Liu, Y., Qiu, Y., Wang, J., & Long, M., *TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables*, arXiv, 2024.
- [9] Krasser, M., *Bayesian Optimization*, <https://krasserm.github.io/>, 2018.
- [10] Gondu, M., *A Gentle Introduction to Bayesian Optimization* <https://www.miguelgondou.com/>, 2023.
- [11] Google Developers, *Introduction to Gradient Boosted Decision Trees (GBDT)*, <https://developers.google.com/>.
- [12] Microsoft, *Overview of Microsoft Fabric*, <https://learn.microsoft.com/>, 2025.