

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - TIN HỌC



STOCK MARKET TREND ANALYSIS

ĐỒ ÁN TỐT NGHIỆP CỦ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Giảng viên hướng dẫn: ThS. Đoàn Thị Trâm

Sinh viên thực hiện:

- Nguyễn Thị Bích Ngọc
- Đoàn Thị Mẫn Nhi
- Nguyễn Thúy Vy

Tp. Hồ Chí Minh, tháng 07 năm 2025

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - TIN HỌC



STOCK MARKET TREND ANALYSIS

ĐỒ ÁN TỐT NGHIỆP CỦ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Giảng viên hướng dẫn: ThS. Đoàn Thị Trâm

Sinh viên thực hiện:

- Nguyễn Thị Bích Ngọc - 21280100
- Đoàn Thị Mẫn Nhi - 21280102
- Nguyễn Thúy Vy - 21280121

Tp. Hồ Chí Minh, tháng 07 năm 2025

Lời cam đoan

Chúng tôi xin cam đoan đề tài đồ án tốt nghiệp "**Stock Market Trend Analysis**" là công trình nghiên cứu của riêng chúng tôi. Các số liệu và kết quả nghiên cứu trong đồ án này là trung thực và không trùng lặp với các đề tài khác. Những số liệu, bảng biểu phục vụ cho việc phân tích và dẫn dắt đề tài này được thu thập từ các nguồn tài liệu khác nhau được ghi chú trong mục tài liệu tham khảo hoặc chú thích ngay bên dưới các bảng biểu.

Lời cảm ơn

Trải qua bốn năm học tập và rèn luyện tại Trường Đại học Khoa học Tự nhiên – ĐHQG TP.HCM, chúng tôi đã có cơ hội tiếp cận, tích lũy những kiến thức quý báu và phát triển tư duy khoa học. Đặc biệt, trong quá trình thực hiện đồ án tốt nghiệp – cột mốc quan trọng khép lại hành trình đại học, nhóm càng biết ơn các giá trị của sự đồng hành và hỗ trợ từ nhà trường và quý thầy cô.

Nhóm xin gửi lời cảm ơn chân thành đến **Ban Giám hiệu Trường Đại học Khoa học Tự nhiên – ĐHQG TP.HCM** và tập thể quý thầy cô **Khoa Toán – Tin học** đã tạo điều kiện học tập, cung cấp môi trường đào tạo chuyên nghiệp và truyền đạt những kiến thức nền tảng quý giá trong suốt khoảng thời gian được học tập tại trường.

Với lòng biết ơn sâu sắc, nhóm xin được gửi lời cảm ơn chân thành đến **ThS. Đoàn Thị Trâm**, giảng viên hướng dẫn đã đồng hành cùng nhóm trong suốt quá trình thực hiện đề tài "**Stock Market Trend Analysis**". Với sự hỗ trợ quý báu, những góp ý chuyên môn sâu sắc và sự tận tâm trong từng buổi trao đổi, cô đã giúp nhóm chúng tôi vượt qua nhiều khó khăn, định hướng đúng đắn cách tiếp cận đề tài và cung cấp những kiến thức bổ ích. Những chia sẻ từ cô không chỉ giúp nhóm phát triển kiến thức chuyên môn mà còn khơi dậy niềm đam mê nghiên cứu.

Tuy nhiên vì kiến thức chuyên môn còn hạn chế và còn thiếu nhiều kinh nghiệm nên nội dung của báo cáo không tránh khỏi những thiếu sót, nhóm rất mong nhận sự góp ý, chỉ bảo thêm của quý thầy cô để báo cáo này được hoàn thiện hơn.

Lời cuối cùng, nhóm chúng tôi xin gửi lời cảm ơn đến gia đình, người thân, bạn bè đã tạo điều kiện và khích lệ để nhóm chúng tôi có thể hoàn thành đồ án tốt nghiệp này.

Trân trọng,
Nhóm thực hiện

Mục lục

Lời cam đoan	i
Lời cảm ơn	ii
Mục lục	iii
Danh sách hình ảnh	vii
Danh sách bảng	viii
Danh sách từ viết tắt	x
Danh sách thuật ngữ	xiii
Tóm tắt đồ án	xiii
1 Tổng quan đề tài	1
1.1 Giới thiệu bài toán	1
1.2 Mục tiêu và nhiệm vụ	2
1.3 Đối tượng và phạm vi	4
2 Cơ sở lý thuyết	6
2.1 Thị trường chứng khoán và dữ liệu tài chính	6
2.1.1 Khái quát về thị trường chứng khoán	6
2.1.2 Vai trò và đặc điểm của dữ liệu tài chính	6
2.1.3 Các loại dữ liệu thường dùng trong dự đoán	7
2.1.4 Đặc điểm dữ liệu dạng chuỗi thời gian trong tài chính	8
2.2 Data pipeline	9
2.2.1 Khái niệm Data Pipeline	9
2.2.2 Kiến trúc Data Pipeline	9

2.2.3	Thách thức	10
2.3	Bài toán dự đoán chuỗi thời gian	11
2.3.1	Khái niệm	11
2.3.2	Phân loại bài toán	11
2.3.3	Đặc điểm và thách thức	12
2.3.4	Vai trò của Deep learning trong dự đoán chuỗi thời gian	13
2.4	Chuẩn hóa dữ liệu, Độ đo hiệu suất và Khoảng tin cậy dự báo	14
2.4.1	Chuẩn hóa dữ liệu	14
2.4.2	Các chỉ số đo lường hiệu suất mô hình	16
2.4.3	Dánh giá độ tin cậy qua Khoảng dự báo (Prediction Interval)	19
2.5	Các mô hình và thuật toán liên quan	20
2.5.1	Mạng nơ-ron tích chập (CNN)	20
2.5.2	Mạng LSTM và kiến trúc Stacked LSTM	22
2.5.3	Mạng Gated Recurrent Unit (GRU)	24
2.5.4	Cơ chế Chú ý (Attention Mechanism)	26
2.5.5	Học Đa nhiệm (Multi-task Learning)	29
2.5.6	Mô hình DA-RNN (Dual-Stage Attention-based RNN)	34
2.5.7	SeBlock(Squeeze and Excitation Block)	37
2.5.8	Mô hình 3D CNN-GRU	39
3	Kiến trúc hệ thống và phương pháp đề xuất	43
3.1	Thiết lập thực nghiệm chung	43
3.1.1	Môi trường và công cụ	43
3.1.2	Kiến trúc đường ống dữ liệu và triển khai	45
3.1.3	Mô tả tập dữ liệu	48
3.2	Mô hình Attention và học đa nhiệm	54
3.2.1	Cấu hình Mô hình Cơ sở: Stacked LSTM	54
3.2.2	Mô hình đề xuất: MT-DA-RNN	55
3.3	Mô hình kết hợp CNN – GRU và khối SEBlock (CGSE) .	60
3.3.1	Kiến trúc mô hình CGSE	60
3.3.2	Quy trình huấn luyện mô hình	64

4 Phân tích kết quả và trực quan trên Dashboard	67
4.1 Kết quả đánh giá - Mô hình Attention và học đa nhiệm	67
4.1.1 So sánh hiệu suất dự báo giá	67
4.1.2 Đánh giá hiệu suất dự báo xu hướng	71
4.1.3 Phân tích khả năng giải thích mô hình (Interpretability)	73
4.2 Phân tích hiệu quả chiến lược giao dịch (Backtest)	75
4.3 Kết quả đánh giá - Mô hình kết hợp CNN– GRU và khối SEBlock	77
4.3.1 Mục tiêu	77
4.3.2 Thiết lập thực nghiệm	77
4.3.3 Kết quả	77
4.3.4 Nhận xét	80
4.4 So sánh kết quả của các mô hình	82
4.5 Trực quan hóa trên Dashboard	83
4.5.1 Giới thiệu Dashboard	83
4.5.2 Tổng quan Thị trường - Bảng tổng quan thị trường thời gian thực	84
4.5.3 Biểu đồ Giá Tương tác với Phân tích Kỹ thuật	86
4.5.4 Dashboard Tiền mã hóa Thời gian thực	87
4.5.5 Tổng kết Kiến trúc Dashboard	88
5 Kết luận và định hướng	89
5.1 Kết luận	89
5.2 Hướng phát triển trong tương lai	90
Tài liệu tham khảo	92

Danh sách hình

Hình 1.1 Biểu đồ phân tích xu hướng giá cổ phiếu FPT.VN giai đoạn 01/2025 - 06/2025	1
Hình 2.1 Kiến trúc mô hình LSTM.	23
Hình 2.2 Kiến trúc mô hình GRU.	26
Hình 2.3 Kiến trúc Hard Parameter Sharing trong Học Da nhiệm.	30
Hình 2.4 Kiến trúc Soft Parameter Sharing trong Học Da nhiệm.	31
Hình 2.5 Sơ đồ tổng quát kiến trúc DA-RNN gồm hai cơ chế attention: (a) Cơ chế chú ý đầu vào (Input Attention); (b) Cơ chế chú ý theo thời gian (Temporal Attention) [1].	35
Hình 2.6 Kiến trúc mô hình SEBlock	37
Hình 2.7 Sơ đồ kiến trúc mô hình 3D CNN - GRU	40
Hình 3.1 Sơ đồ kiến trúc tổng thể của Đường ống Dữ liệu Chứng khoán Thời gian thực với hơn 15 dịch vụ được container hóa	47
Hình 3.2 Sơ đồ tổng quan quy trình thực nghiệm cho mô hình đề xuất MT-DA-RNN.	59
Hình 3.3 Kiến trúc mô hình CGSE	61
Hình 3.4 Quy trình huấn luyện mô hình CNN-GRU-SEBlock .	64
Hình 3.5 Minh họa kết quả đánh giá mô hình CGSE	66
Hình 3.6 Minh họa đồ thị dự đoán mô hình CGSE	66
Hình 4.1 Đồ thị so sánh đường giá thực tế với giá dự báo của MT-DA-RNN (màu đỏ) và Stacked LSTM (màu xanh lá) cho mã BID.VN.	70

Hình 4.2	Ma trận nhầm lẫn (Confusion Matrix) của mô hình MT-DA-RNN trên tập kiểm thử cho mã BID.VN.	72
Hình 4.3	Trọng số trung bình của Input Attention cho mã BID.VN. Cột cao hơn thể hiện mức độ tập trung cao hơn.	73
Hình 4.4	Phân bố trọng số trung bình của Temporal Attention trên cửa sổ 15 ngày quá khứ cho mã BID.VN.	74
Hình 4.5	Dự báo giá trên 2 mô hình baseline và 2 mô hình cải tiến so với giá thực	82
Hình 4.6	Giao diện tổng quan Dashboard với bố cục nhiều biểu đồ và cập nhật thời gian thực	84
Hình 4.7	Biểu đồ đường Giá Tương tác	86
Hình 4.8	Dashboard Tiền mã hóa Tần suất Siêu cao với cập nhật 1 giây	87

Danh sách bảng

Bảng 3.1 Cấu trúc bảng stock_daily - Dữ liệu OHLCV Lịch sử Hàng ngày	48
Bảng 3.2 Cấu trúc bảng stock_intraday_5m - Dữ liệu Khung 5 phút	49
Bảng 3.3 Cấu trúc bảng stock_realtime_1m - Báo giá Thời gian thực	49
Bảng 3.4 Cấu trúc bảng stock_metadata - Thông tin Công ty và Thị trường	50
Bảng 3.5 Các tham số chính của mô hình Stacked LSTM cơ sở.	55
Bảng 3.6 Các chỉ báo kỹ thuật được sử dụng trong mô hình đề xuất.	57
Bảng 4.1 Kết quả dự báo giá của mô hình cơ sở baseline Stacked LSTM trên 20 mã tài sản.	68
Bảng 4.2 Kết quả dự báo giá của mô hình đề xuất MT-DA-RNN trên 20 mã tài sản.	69
Bảng 4.3 Bảng so sánh các chỉ số đánh giá phân loại (Precision, Recall, F1-score, Accuracy, MCC) của mô hình MT-DA-RNN trên 20 mã tài sản.	71
Bảng 4.4 So sánh hiệu suất giữa chiến lược MT-DA-RNN và Buy & Hold.	76
Bảng 4.5 Kết quả đánh giá của mô hình baseline CNN-GRU trên 20 mã tài sản.	78
Bảng 4.6 Kết quả đánh giá của mô hình CGSE trên 20 mã tài sản.	79

Danh sách từ viết tắt

1. **1D**: One-Dimensional (Dữ liệu một chiều)
2. **AR**: Autoregressive
3. **ARIMA**: Autoregressive Integrated Moving Average
4. **EMA**: Exponential Moving Average
5. **FC**: Fully Connected Layer
6. **GARCH**: Generalized Autoregressive Conditional Heteroskedasticity
7. **LightGBM**: Light Gradient Boosting Machine
8. **MACD**: Moving Average Convergence Divergence
9. **MA**: Moving Average
10. **MTL**: Multi-Task Learning
11. **OHLCV**: Open - High - Low - Close - Volume
12. **RSI**: Relative Strength Index
13. **SARIMA**: Seasonal ARIMA (Autoregressive Integrated Moving Average with Seasonality)
14. **SVR**: Support Vector Regression
15. **Transformer**: Transformer Architecture (Mô hình học sâu sử dụng cơ chế Attention)
16. **XGBoost**: Extreme Gradient Boosting

Danh sách thuật ngữ

1. **Activation:** Hàm kích hoạt
2. **Application Programming Interface (API):** Giao diện lập trình ứng dụng
3. **Attention Mechanism:** Cơ chế chú ý
4. **Backtest:** Phương pháp kiểm tra lại hiệu quả của chiến lược giao dịch
5. **Baseline:** Mô hình cơ sở
6. **Buy:** Mua
7. **Dashboard:** Bảng điều khiển
8. **Data Lake:** Hồ dữ liệu
9. **Data Warehouse:** Kho dữ liệu
10. **Decoder:** Bộ giải mã
11. **Direct Forecasting:** Dự báo trực tiếp
12. **Dynamic context vector:** Vec-tơ ngữ cảnh động
13. **Encoder:** Bộ mã hóa
14. **Exponential Smoothing:** Làm mượt hàm mũ
15. **Feature Engineering:** Kỹ thuật tạo đặc trưng
16. **Hard Parameter Sharing:** Chia sẻ tham số cứng
17. **Hold:** Giữ
18. **Inductive transfer:** Chuyển giao suy luận

19. **Information Bottleneck**: Nút thắt thông tin
20. **Input Attention**: Cơ chế chú ý theo đầu vào
21. **Interpretability**: Khả năng diễn giải
22. **Loss Function**: Hàm mất mát
23. **Machine Learning**: Học máy
24. **Multivariate**: Đa biến
25. **Multivariate Time Series Forecasting**: Dự báo chuỗi thời gian đa biến
26. **Multi-step Forecasting**: Dự báo nhiều bước
27. **Multi-Task Learning**: Học đa nhiệm
28. **One-step Forecasting**: Dự báo một bước
29. **Overfitting**: Quá khớp
30. **Preprocessing**: Tiền xử lý
31. **Recursive Forecasting**: Dự báo đệ quy
32. **Regularization**: Quy chuẩn hóa
33. **Rescale**: Tái chuẩn hóa
34. **Scale**: Chuẩn hóa
35. **Sequence-to-Sequence**: Mô hình chuỗi sang chuỗi
36. **Shared Layers**: Tầng chia sẻ
37. **Shared representation**: Biểu diễn chia sẻ
38. **Snapshot**: Ảnh chụp trạng thái mô hình
39. **Soft Parameter Sharing**: Chia sẻ tham số mềm

40. **Stacked:** Xếp chồng (mô hình/lớp)
41. **Stock Market Trend Analysis:** Phân tích xu hướng thị trường chứng khoán
42. **Task-Specific Layers:** Tầng chuyên biệt
43. **Technical Indicators:** Chỉ báo kỹ thuật
44. **Temporal Attention:** Cơ chế chú ý theo thời gian
45. **Time series:** Chuỗi thời gian
46. **Univariate Time Series Forecasting:** Dự báo chuỗi thời gian đơn biến
47. **Weight:** Trọng số

Tóm tắt đồ án

Đề tài “**Stock Market Trend Analysis**” tập trung vào việc nghiên cứu và ứng dụng các phương pháp học sâu trong phân tích dữ liệu tài chính, cụ thể là dữ liệu chuỗi thời gian về giá cổ phiếu. Với sự phát triển nhanh chóng của công nghệ trí tuệ nhân tạo, việc ứng dụng các mô hình học sâu vào dự báo tài chính đang ngày càng trở nên phổ biến và mang lại nhiều triển vọng thực tiễn.

Trong khuôn khổ đề tài, nhóm sinh viên đã tiến hành thu thập, xử lý và chuẩn hóa dữ liệu lịch sử của nhiều mã cổ phiếu để xây dựng bộ dữ liệu đầu vào cho mô hình. Sau đó, nhóm đề xuất kiến trúc mô hình học sâu phù hợp nhằm khai thác các đặc trưng tiềm ẩn trong dữ liệu và đưa ra dự đoán về xu hướng biến động giá cổ phiếu trong ngắn hạn.

Quá trình thực hiện đề tài bao gồm các bước: phân tích dữ liệu, thiết kế mô hình, huấn luyện và đánh giá hiệu quả dự đoán thông qua các chỉ số đo lường độ sai số. Kết quả cho thấy mô hình có khả năng nắm bắt được xu hướng chung của thị trường, từ đó hỗ trợ nhà đầu tư trong việc ra quyết định đầu tư một cách hiệu quả hơn.

Bên cạnh việc xây dựng mô hình dự đoán, đề tài còn chú trọng đến khả năng trực quan hóa dữ liệu và kết quả đầu ra. Nhóm đã thiết kế một dashboard nhằm trình bày trực quan các thông tin liên quan đến xu hướng giá cổ phiếu, giúp người dùng dễ dàng theo dõi và đánh giá mô hình.

Đề tài không chỉ mang tính thực tiễn cao mà còn giúp sinh viên rèn luyện kỹ năng nghiên cứu, xử lý dữ liệu, lập trình, và tư duy phân tích – những yếu tố quan trọng trong lĩnh vực khoa học dữ liệu và trí tuệ nhân tạo hiện nay.

Nội dung báo cáo đồ án tốt nghiệp gồm 5 chương:

1. **Tổng quan đề tài:** Giới thiệu, lý do chọn đề tài, mục đích nhiệm vụ, phạm vi đề tài.
2. **Cơ sở lý thuyết:** Xác định bài toán và các kiến thức liên quan.
3. **Kiến trúc hệ thống và phương pháp đề xuất:** Mô tả quá trình thực hiện Data Pipeline và xây dựng mô hình.
4. **Phân tích kết quả trực quan hóa Dashboard:** Đánh giá các kết quả huấn luyện mô hình và trực quan hóa dữ liệu trên Dashboard.
5. **Kết luận và định hướng:** Dưa ra kết luận về mô hình đã thực hiện và trình bày hướng đi trong tương lai của bài toán.

Chương 1

Tổng quan đề tài

1.1 Giới thiệu bài toán

Trong bối cảnh thị trường tài chính ngày càng phát triển mạnh mẽ, việc đưa ra các quyết định đầu tư hiệu quả là một thách thức không nhỏ đối với các nhà đầu tư cá nhân lẫn tổ chức. Một trong những yếu tố then chốt để thành công trên thị trường là khả năng dự đoán biến động giá cổ phiếu trong tương lai. Tuy nhiên, thị trường chứng khoán là một hệ thống phức tạp, bị ảnh hưởng bởi nhiều yếu tố như tâm lý nhà đầu tư, tình hình kinh tế vĩ mô, tin tức, và đặc điểm riêng của từng doanh nghiệp. Do đó, việc xây dựng các mô hình dự báo chính xác là một bài toán khó, đòi hỏi sự kết hợp giữa hiểu biết tài chính và kỹ thuật phân tích dữ liệu hiện đại.



Hình 1.1: Biểu đồ phân tích xu hướng giá cổ phiếu FPT.VN giai đoạn 01/2025 - 06/2025

Các phương pháp thống kê truyền thống như ARIMA, GARCH hay mô hình hồi quy tuyến tính đã được áp dụng rộng rãi trong dự báo chuỗi

thời gian tài chính. Tuy nhiên, các mô hình này thường giả định tuyến tính và không thể hiện được mối quan hệ phi tuyến và phức tạp giữa các yếu tố tác động đến giá cổ phiếu. Trong những năm gần đây, sự phát triển nhanh chóng của các mô hình học sâu đã mở ra nhiều cơ hội mới trong phân tích và dự báo tài chính. Các kiến trúc như CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks), GRU (Gated Recurrent Unit), hay Transformer đã được chứng minh có hiệu quả cao trong xử lý dữ liệu chuỗi thời gian nhờ khả năng học các mẫu phức tạp và ghi nhớ dài hạn.

Trong đề tài này, nhóm tập trung xây dựng mô hình dự đoán giá cổ phiếu dựa trên dữ liệu chuỗi thời gian bao gồm các đặc trưng kỹ thuật như giá mở cửa, giá đóng cửa, giá cao nhất, giá thấp nhất và khối lượng giao dịch. Thay vì xây dựng mô hình riêng biệt cho từng mã cổ phiếu như các hướng tiếp cận truyền thống, đề tài lựa chọn hướng tiếp cận hiện đại hơn: huấn luyện một mô hình học sâu duy nhất trên toàn bộ tập dữ liệu gồm nhiều mã cổ phiếu. Mô hình được huấn luyện chung nhằm tận dụng mối quan hệ liên ngành, học các mẫu biến động mang tính toàn thị trường, từ đó nâng cao khả năng tổng quát hóa và giảm thiểu hiện tượng overfitting khi chỉ huấn luyện trên một mã duy nhất.

Thông qua việc huấn luyện chung trên toàn bộ dữ liệu nhưng đánh giá riêng cho từng mã, đề tài mong muốn đạt được mô hình có độ chính xác cao, ổn định và khả năng áp dụng thực tiễn trong phân tích tài chính.

1.2 Mục tiêu và nhiệm vụ

Trong quá trình thực hiện đề tài, nhóm nghiên cứu đã xác định rõ các mục tiêu cần đạt được cũng như các nhiệm vụ cụ thể cần thực hiện để đảm bảo tính nhất quán và hợp lý của toàn bộ quá trình nghiên cứu. Các mục tiêu và nhiệm vụ này là kim chỉ nam xuyên suốt đề tài, đảm bảo định hướng rõ ràng và kết quả khả thi.

Để đạt được mục tiêu chung nêu trên, đề tài cần hoàn thành các mục tiêu cụ thể sau:

1. Tìm hiểu và phân tích các phương pháp học sâu phổ biến trong dự báo chuỗi thời gian tài chính, từ đó lựa chọn kiến trúc mô hình phù hợp với bài toán dự đoán giá cổ phiếu.
2. Xây dựng pipeline xử lý dữ liệu chứng khoán chứa nhiều mã cổ phiếu, bao gồm các bước như thu thập, chuẩn hóa dữ liệu, tao tập huấn luyện và kiểm thử sao cho đảm bảo độ tin cậy và chất lượng dữ liệu.
3. Triển khai mô hình baseline.
4. Cải tiến mô hình để tăng khả năng lọc đặc trưng ẩn quan trọng, giúp mô hình tập trung vào các chiều thông tin có giá trị cao trong quá trình dự đoán.
5. Dánh giá hiệu suất mô hình dựa trên các chỉ số như MSE, MAPE, RMSE, trong đó các chỉ số này được tính chung trên toàn bộ tập kiểm thử để đánh giá khả năng khái quát tổng thể của mô hình.
6. Vẽ biểu đồ so sánh kết quả dự đoán và thực tế trên từng mã cổ phiếu nhằm phân tích trực quan độ chính xác của mô hình trên từng đối tượng cụ thể, từ đó phát hiện các xu hướng hoặc bất thường trong kết quả dự đoán.

Dựa trên các mục tiêu đề ra, đề tài bao gồm các nhiệm vụ cụ thể sau đây nhằm đảm bảo quy trình thực hiện có hệ thống và hiệu quả:

1. Thu thập và xử lý dữ liệu chứng khoán từ nguồn tin cậy, đảm bảo dữ liệu đầu vào có chất lượng cao và đầy đủ thông tin cần thiết cho bài toán dự báo.
2. Thiết kế kiến trúc mô hình, từ đó xây dựng mô hình huấn luyện hiệu quả và linh hoạt trong việc xử lý chuỗi thời gian.
3. Huấn luyện mô hình trên tập dữ liệu tập hợp từ nhiều mã cổ phiếu, nhằm tận dụng các mối liên hệ tiềm ẩn giữa các mã và tăng khả năng học các mẫu mang tính tổng quát.

4. Chia tập kiểm thử riêng theo từng mã cổ phiếu nhưng tính toán chỉ số đánh giá tổng thể, từ đó vừa đảm bảo đánh giá khách quan về mô hình vừa có thể theo dõi chi tiết theo từng cổ phiếu cụ thể.
5. Trực quan hóa kết quả dự đoán, phân tích độ chính xác và đề xuất hướng cải tiến, giúp người dùng dễ dàng hiểu được hiệu suất mô hình và tiềm năng mở rộng trong tương lai.

1.3 Đối tượng và phạm vi

Việc xác định rõ đối tượng và phạm vi nghiên cứu giúp đề tài có trọng tâm và đảm bảo phù hợp với năng lực, thời gian và tài nguyên thực hiện. Đề tài tập trung vào các đối tượng sau:

1. Mô hình học sâu trong bài toán dự đoán chuỗi thời gian, đặc biệt là các kiến trúc phù hợp với dữ liệu tài chính như: CNN, GRU, LSTM, ...
2. Dữ liệu giá cổ phiếu từ thị trường chứng khoán, bao gồm các chỉ số kỹ thuật phổ biến như giá mở cửa, đóng cửa, giá cao nhất, giá thấp nhất và khối lượng giao dịch theo từng ngày.

Để đảm bảo tính khả thi và kiểm soát chất lượng kết quả, đề tài giới hạn phạm vi nghiên cứu như sau:

1. Mô hình thực hiện dự đoán một vài bước trong tương lai, nhằm kiểm nghiệm khả năng dự báo ngắn hạn và đánh giá độ chính xác trong điều kiện thực tế.
2. Các đặc trưng đầu vào được giới hạn ở thông tin kỹ thuật, bao gồm dữ liệu giá và khối lượng, không bao gồm yếu tố vĩ mô, tin tức thời sự hoặc cảm xúc thị trường, nhằm kiểm soát tốt tính đơn biến và giảm nhiễu cho mô hình.
3. Việc đánh giá mô hình được thực hiện trên toàn bộ tập kiểm thử gộp, nhưng kết quả vẫn được phân tích trực quan theo từng mã thông qua biểu đồ, giúp phát hiện các trường hợp dự đoán tốt hoặc kém.

Chương 1 đã trình bày tổng quan về đề tài, bao gồm bối cảnh, mục tiêu nghiên cứu, các nhiệm vụ cần thực hiện, cũng như đối tượng và phạm vi của đề tài. Việc lựa chọn mô hình học sâu để dự đoán giá cổ phiếu từ dữ liệu chuỗi thời gian nhiều mã là một hướng đi tiềm năng, vừa có tính ứng dụng thực tiễn cao, vừa đặt ra nhiều thử thách nghiên cứu. Những nội dung được trình bày trong chương này sẽ là nền tảng quan trọng cho các chương tiếp theo, đặc biệt là phần cơ sở lý thuyết và triển khai mô hình.

Chương 2

Cơ sở lý thuyết

2.1 Thị trường chứng khoán và dữ liệu tài chính

2.1.1 Khái quát về thị trường chứng khoán

Thị trường chứng khoán là nơi diễn ra hoạt động mua bán các loại chứng khoán như cổ phiếu, trái phiếu, chứng chỉ quỹ và các công cụ tài chính khác. Đây là một phần quan trọng của thị trường tài chính, đóng vai trò huy động vốn trung và dài hạn cho các doanh nghiệp cũng như Chính phủ, đồng thời tạo ra môi trường đầu tư cho các cá nhân và tổ chức.

Trên thị trường, giá trị của các loại chứng khoán được xác lập thông qua nguyên tắc cung – cầu, phản ánh kỳ vọng của nhà đầu tư về tình hình kinh doanh của doanh nghiệp và bối cảnh kinh tế vĩ mô. Thị trường chứng khoán thường có tính thanh khoản cao, biến động nhanh, và chịu ảnh hưởng bởi nhiều yếu tố như tâm lý đám đông, tin tức, chính sách điều hành, lãi suất, lạm phát, ...

Thị trường chứng khoán Việt Nam hiện bao gồm ba sàn giao dịch chính: HOSE (Sở Giao dịch Chứng khoán TP.HCM), HNX (Hà Nội), và UPCoM. Cùng với sự phát triển mạnh mẽ của công nghệ tài chính (fintech), lượng dữ liệu phát sinh từ thị trường ngày càng lớn, tạo điều kiện thuận lợi cho các bài toán phân tích và dự báo bằng mô hình học máy.

2.1.2 Vai trò và đặc điểm của dữ liệu tài chính

Dữ liệu tài chính – đặc biệt là dữ liệu giá cổ phiếu – là nguồn thông tin quan trọng giúp nhà đầu tư phân tích hành vi thị trường, đánh giá hiệu quả đầu tư và đưa ra quyết định mua bán. Việc khai thác dữ liệu tài chính không chỉ giúp mô hình hóa biến động giá mà còn góp phần phát hiện xu hướng, tín hiệu bất thường, hoặc điểm mua/bán tiềm năng.

Một số đặc điểm nổi bật của dữ liệu tài chính:

1. Có tính thời gian (temporal): Mỗi điểm dữ liệu đều gắn với mốc thời gian cụ thể, đòi hỏi phải xử lý như một chuỗi liên tục.
2. Đa biến (multivariate): Một cổ phiếu có thể có nhiều đặc trưng đi kèm (giá mở cửa, đóng cửa, khối lượng, ...).
3. Tính phi tuyến và nhiễu cao: Dữ liệu giá cổ phiếu thường rất khó dự đoán do không tuân theo phân bố chuẩn hoặc quy luật tuyến tính rõ ràng.
4. Dễ bị ảnh hưởng bởi yếu tố ngoại sinh: Tin tức, chính sách hoặc sự kiện bất ngờ có thể tác động mạnh tới giá cổ phiếu, làm cho việc dự báo trở nên không chắc chắn.

Chính vì những đặc điểm này, việc phân tích và dự báo dữ liệu tài chính đòi hỏi phải áp dụng các phương pháp có khả năng biểu diễn linh hoạt và học được mối quan hệ phức tạp trong dữ liệu – điều mà các mô hình học sâu hiện đại có thể đáp ứng.

2.1.3 Các loại dữ liệu thường dùng trong dự đoán

Trong các mô hình dự đoán giá cổ phiếu, dữ liệu sử dụng thường được lấy từ dữ liệu kỹ thuật (technical data), tức là các chỉ số trực tiếp từ thị trường chứng khoán mà không tính đến yếu tố vĩ mô hay tin tức. Các đặc trưng phổ biến bao gồm:

- Giá mở cửa (Open): Giá giao dịch đầu tiên trong phiên.
- Giá đóng cửa (Close): Giá giao dịch cuối cùng trong phiên – thường được xem là quan trọng nhất vì phản ánh tâm lý chốt phiên.
- Giá cao nhất (High): Mức giá cao nhất đạt được trong phiên.
- Giá thấp nhất (Low): Mức giá thấp nhất trong phiên.
- Khối lượng giao dịch (Volume): Tổng số cổ phiếu được giao dịch trong phiên – phản ánh mức độ quan tâm của thị trường.

Tùy vào chiến lược dự đoán và độ phức tạp của mô hình, có thể sử dụng tất cả các đặc trưng trên hoặc chỉ chọn một vài đặc trưng quan trọng như giá đóng cửa và khối lượng.

2.1.4 Đặc điểm dữ liệu dạng chuỗi thời gian trong tài chính

Dữ liệu tài chính thường được xem là một loại chuỗi thời gian đa biến (multivariate time series), nơi mà mỗi đặc trưng được ghi nhận tại một thời điểm nhất định. Điều này làm cho việc xử lý và dự đoán khác biệt so với các dữ liệu thông thường¹:

- Tính phụ thuộc theo thời gian: Giá hôm nay chịu ảnh hưởng bởi giá của những ngày trước đó. Việc mô hình hóa sự phụ thuộc này là mục tiêu chính của các mô hình chuỗi thời gian.
- Tính mùa vụ: Một số loại cổ phiếu hoặc ngành nghề có xu hướng biến động theo thời điểm (ví dụ: ngân hàng tăng trưởng mạnh vào cuối quý).
- Độ biến động cao và không ổn định: Dữ liệu thường biến động không theo quy luật rõ ràng, dễ bị ảnh hưởng bởi yếu tố ngẫu nhiên.
- Không tuyến tính: Mỗi quan hệ giữa các đặc trưng thường không tuân theo hàm tuyến tính đơn giản, do đó các mô hình tuyến tính như hồi quy thường không đạt hiệu quả cao.
- Có thể chứa nhiều hoặc không tin thừa: Các yếu tố như khối lượng tăng đột biến hoặc giá tăng mạnh có thể chỉ là nhiễu trong ngắn hạn, nhưng vẫn ảnh hưởng đến mô hình nếu không xử lý tốt.

Chính vì thế, trong bài toán dự đoán giá chứng khoán, việc lựa chọn mô hình có khả năng học được đặc trưng chuỗi, phát hiện mẫu ẩn và thích nghi với biến động là điều hết sức cần thiết.

¹<https://blog.tomorrowmarketers.org/time-series-analysis/>

2.2 Data pipeline

Trong các bài toán học máy và học sâu, việc thiết kế một quy trình xử lý dữ liệu – hay còn gọi là data pipeline – đóng vai trò hết sức quan trọng. Một mô hình dù hiện đại và phức tạp đến đâu cũng không thể đưa ra kết quả chính xác nếu dữ liệu đầu vào không được xử lý đúng cách, đầy đủ và phù hợp với đặc điểm của mô hình.

Đối với bài toán dự đoán giá cổ phiếu dựa trên chuỗi thời gian tài chính, data pipeline không chỉ đơn thuần là làm sạch dữ liệu mà còn bao gồm nhiều công đoạn chặt chẽ để chuyển đổi dữ liệu thô thành tập huấn luyện có cấu trúc, phục vụ cho mô hình học sâu. Các bước chính trong pipeline bao gồm: thu thập dữ liệu, tiền xử lý, chuẩn hóa, tạo mẫu theo cửa sổ thời gian, chia tập huấn luyện – kiểm thử, và tổ chức thành dạng input phù hợp cho mô hình.

2.2.1 Khái niệm Data Pipeline

Data Pipeline (đường ống dữ liệu) là một chuỗi các quy trình được tổ chức để thu thập, chuyển đổi, lưu trữ và xử lý dữ liệu từ nhiều nguồn khác nhau đến đích tự động và liên tục. Mục tiêu của Data Pipeline là tổ chức, xử lý dữ liệu và đảm bảo dữ liệu được truyền tải một cách liền mạch, chính xác và sẵn sàng cho các bước phân tích hoặc dự đoán tiếp theo.

2.2.2 Kiến trúc Data Pipeline

Một Data Pipeline thường có kiến trúc cơ bản:

- Thu thập dữ liệu:** Là quá trình thu thập dữ liệu từ nhiều nguồn khác nhau như API(Application Programming Interface) và chuyển vào luồng Data Pipeline để xử lý.
- Chuyển đổi dữ liệu:** Là quá trình chuyển đổi và xử lý dữ liệu về định dạng phù hợp với yêu cầu của bài toán phân tích và phù hợp với mô hình được xây dựng sau đó.

Quá trình gồm các bước:

- Chọn lọc dữ liệu
 - Chuẩn hóa dữ liệu
 - Xử lý các dữ liệu bị lỗi: dữ liệu bị thiếu, trùng lặp
 - Tính toán các thông số cần thiết
3. **Lưu trữ và quản lý dữ liệu:** Sau khi hoàn thành quá trình xử lý, dữ liệu có thể được lưu trữ trong các hệ thống phù hợp như Data Lake, Data Warehouse, ...

2.2.3 Thách thức

Khi thiết kế đường ống dữ liệu, cần phải giải quyết một số thách thức để duy trì hiệu quả của kiến trúc², bao gồm:

- Đảm bảo chất lượng dữ liệu: Việc duy trì chất lượng dữ liệu trong suốt quá trình xử lý là điều cần thiết, vì dữ liệu chất lượng kém có thể dẫn đến thông tin chi tiết không chính xác và dự đoán lỗi.
- Giảm độ phức tạp khi tích hợp: Việc tích hợp các nguồn dữ liệu và công nghệ khác nhau là một công việc phức tạp, đòi hỏi phải lập kế hoạch và phân tích chuyên sâu để tránh các vấn đề về khả năng tương thích và các xung đột tích hợp dữ liệu khác.
- Duy trì bảo mật và quyền riêng tư dữ liệu: Việc bảo vệ dữ liệu nhạy cảm trong suốt quá trình truyền dữ liệu, từ khi thu thập đến khi phân tích, thường là yêu cầu bắt buộc về mặt tuân thủ pháp lý, đòi hỏi phải phân tích và thực hiện cẩn thận liên tục.
- Giải quyết các hạn chế về khả năng mở rộng: Khi khối lượng dữ liệu tăng lên, một đường ống dữ liệu được thiết kế kém sẽ nhanh chóng trở nên không bền vững và khó quản lý. Đảm bảo chất lượng dữ liệu phù hợp và phân phối theo thời gian đòi hỏi phải cân nhắc cẩn thận về cơ sở hạ tầng dữ liệu cơ bản so với khối lượng dữ liệu hiện tại và dự kiến trong tương lai.

²<https://www.datamation.com/big-data/data-pipeline-architecture/>

2.3 Bài toán dự đoán chuỗi thời gian

2.3.1 Khái niệm

Chuỗi thời gian (time series) là một dãy các giá trị được quan sát và ghi nhận theo trình tự thời gian, thường cách nhau theo ngày, giờ, phút, hoặc các khoảng định kỳ khác. Mỗi giá trị trong chuỗi không độc lập mà có thể phụ thuộc vào các giá trị trước đó – đặc điểm khiến loại dữ liệu này khác biệt với các dạng dữ liệu thông thường khác.

Dự đoán chuỗi thời gian là một lĩnh vực nghiên cứu lâu đời trong thống kê và học máy, với mục tiêu chính là ước lượng giá trị tương lai dựa trên lịch sử đã biết. Các ứng dụng thực tiễn rất phong phú, trải dài từ dự báo thời tiết, nhu cầu tiêu dùng, lưu lượng mạng, cho đến dự đoán giá cả hàng hóa và chứng khoán.

Trong lĩnh vực tài chính, chuỗi thời gian thường là các dữ liệu liên quan đến giá cả, khối lượng giao dịch, hoặc các chỉ số thị trường theo thời gian. Đặc biệt, giá cổ phiếu là một dạng chuỗi thời gian phổ biến, được ghi nhận theo ngày giao dịch và chịu ảnh hưởng bởi nhiều yếu tố phức tạp như thông tin doanh nghiệp, biến động thị trường, chính sách vĩ mô, hoặc cảm lý nhà đầu tư.

Do đó, dự đoán giá cổ phiếu trong tương lai không chỉ đơn thuần là việc “kéo dài xu hướng” trong dữ liệu mà còn là quá trình trích xuất đặc trưng, phát hiện mẫu, và học được các mối quan hệ tiềm ẩn phức tạp trong chuỗi thời gian.

Bài toán trong đề tài có thể được mô hình hóa như sau:

- **Input:** một chuỗi giá trị liên tục của một hoặc nhiều đặc trưng tài chính trong khoảng thời gian $t - n$ đến t .
- **Output:** giá trị mục tiêu tại thời điểm tương lai $t + 1$ hoặc nhiều bước tiếp theo $(t + 1, t + 2, \dots, t + m)$.

2.3.2 Phân loại bài toán

Tùy thuộc vào đầu vào và mục tiêu dự báo, bài toán chuỗi thời gian có thể được phân loại như sau:

1. Theo số lượng biến đầu vào³:

- Univariate Time Series Forecasting: Dự báo dựa trên một biến duy nhất (ví dụ: chỉ dùng giá đóng cửa).
- Multivariate Time Series Forecasting: Sử dụng nhiều đặc trưng đầu vào đồng thời (giá mở cửa, cao nhất, thấp nhất, khối lượng, v.v.) để tăng độ chính xác.

2. Theo khoảng dự⁴:

- One-step Forecasting: Dự đoán giá trị ở một thời điểm duy nhất trong tương lai (ví dụ: ngày mai).
- Multi-step Forecasting: Dự đoán nhiều bước liên tiếp (ví dụ: 5 ngày tiếp theo), đòi hỏi mô hình có khả năng tích lũy và duy trì thông tin lâu dài.

3. Theo cách xây dựng mô hình⁵:

- Direct Forecasting: Dự đoán trực tiếp từng bước trong tương lai (dùng mô hình riêng cho từng bước).
- Recursive Forecasting: Dự đoán từng bước một, sau đó dùng kết quả đó làm đầu vào cho bước kế tiếp.
- Sequence-to-Sequence: Đầu vào và đầu ra đều là chuỗi (ví dụ: 30 ngày đầu vào → 5 ngày đầu ra).

Trong đề tài này, hướng tiếp cận chính là dự đoán đa biến, một bước hoặc nhiều bước, với mục tiêu tận dụng thông tin đa chiều và quá khứ gần để dự đoán chính xác giá trị tương lai.

2.3.3 Đặc điểm và thách thức

Bài toán dự đoán chuỗi thời gian tài chính có một số đặc điểm và thách thức riêng biệt:

³<https://doc.dataiku.com/dss/latest/time-series/understanding-time-series.html>

⁴<https://skforecast.org/0.16.0/introduction-forecasting/introduction-forecasting.html>

⁵<https://openforecast.org/2024/05/25/recursive-vs-direct-forecasting-strategy/>

1. Tính không tuyến tính cao: Giá chứng khoán thường biến động theo các quy luật phi tuyến, rất khó nắm bắt bằng mô hình đơn giản.
2. Độ nhiễu lớn và không ổn định: Chuỗi giá thường có nhiều biến động bất ngờ do tác động của tin tức, cảm xúc thị trường, ...
3. Tính phụ thuộc dài hạn: Một số xu hướng có thể phụ thuộc vào thông tin từ hàng tuần, thậm chí hàng tháng trước.
4. Mối quan hệ đa chiều phức tạp: Giữa các đặc trưng tài chính như giá, khối lượng, hay giữa các cổ phiếu khác nhau.
5. Vấn đề thiếu dữ liệu đồng bộ: Khi dùng dữ liệu từ nhiều mã cổ phiếu, có thể xảy ra tình trạng dữ liệu không đồng đều về thời gian và đặc trưng.

Những thách thức trên đặt ra yêu cầu cho mô hình phải có khả năng biểu diễn linh hoạt, học được mối quan hệ phi tuyến và chọn lọc thông tin quan trọng từ chuỗi thời gian.

2.3.4 Vai trò của Deep learning trong dự đoán chuỗi thời gian

Các phương pháp truyền thống như ARIMA, Exponential Smoothing hay GARCH được dùng trong dự báo chuỗi tài chính. Tuy nhiên, chúng có nhiều hạn chế như:

- Giả định tuyến tính hoặc phân phối chuẩn.
- Khó mở rộng cho dữ liệu đa biến.
- Thiếu khả năng tự động trích xuất đặc trưng.

Trong khi đó, các mô hình học sâu như RNN, LSTM, GRU, CNN, Transformer,... có khả năng:

- Học tự động từ dữ liệu thô, không cần đặc trưng thủ công.
- Mô hình hóa quan hệ phi tuyến phức tạp.
- Ghi nhớ và khai thác được thông tin quá khứ.

- Kết hợp được nhiều nguồn dữ liệu cùng lúc.

Chính vì vậy, học sâu ngày càng trở thành phương pháp chủ đạo trong các bài toán dự báo chuỗi thời gian tài chính, đặc biệt trong bối cảnh dữ liệu ngày càng nhiều và biến động ngày càng phức tạp.

2.4 Chuẩn hóa dữ liệu, Độ đo hiệu suất và Khoảng tin cậy dự báo

2.4.1 Chuẩn hóa dữ liệu

Trong các bài toán học sâu, đặc biệt là khi xử lý dữ liệu tài chính, các đặc trưng đầu vào như giá cổ phiếu và khối lượng giao dịch thường có đơn vị đo và độ lớn khác nhau. Sự chênh lệch này có thể làm chậm quá trình hội tụ của mô hình hoặc khiến các thuật toán tối ưu hoạt động kém hiệu quả. Do đó, việc chuẩn hóa dữ liệu là một bước tiền xử lý quan trọng giúp đưa các đặc trưng về cùng một thang đo, từ đó cải thiện khả năng học và sự ổn định trong quá trình huấn luyện.⁶

MinMax Scaling

MinMax Scaling (hay còn gọi là chuẩn hóa tuyến tính) là một kỹ thuật phổ biến giúp đưa toàn bộ dữ liệu về một khoảng giá trị nhất định, thường là [0, 1] hoặc [-1, 1]. Phương pháp này giữ nguyên hình dạng phân bố ban đầu của dữ liệu, nhưng nén các giá trị về một thang đo chuẩn.

Giả sử ta có một đặc trưng x , MinMax Scaling được định nghĩa như sau:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.1)$$

Trong đó:

- x : giá trị ban đầu,
- x_{min} : giá trị nhỏ nhất trong tập huấn luyện của đặc trưng đó,
- x_{max} : giá trị lớn nhất trong tập huấn luyện,

⁶<https://towardsdatascience.com/data-scaling-101-standardization-and-min-max-scaling-explained>

- x' : giá trị sau khi đã chuẩn hóa.

Ta cũng có thể mở rộng công thức nếu muốn đưa dữ liệu về một khoảng bất kỳ $[a, b]$:

$$x' = A + \frac{(x - x_{min})(b - a)}{x_{max} - x_{min}} \quad (2.2)$$

Trong bài toán dự báo giá chứng khoán, các đặc trưng như giá mở cửa, đóng cửa, khối lượng giao dịch,... có thể có giá trị chênh lệch rất lớn giữa các mã cổ phiếu hoặc giữa các giai đoạn thị trường. Do đó, MinMax Scaling giúp đưa các giá trị này về cùng một thang đo, giúp mô hình học tốt hơn và hội tụ nhanh hơn.

Standard Scaling (Z-score Normalization)

Phương pháp này chuẩn hóa các giá trị đầu vào sao cho phân bố kết quả có trung bình bằng 0 và độ lệch chuẩn bằng 1, giúp loại bỏ sự khác biệt về thang đo giữa các đặc trưng.

Cho một đặc trưng x , công thức chuẩn hóa theo Z-score được tính như sau:

$$x' = \frac{X - \mu}{\sigma} \quad (2.3)$$

Trong đó:

- x : giá trị ban đầu,
- μ : giá trị trung bình (mean) của đặc trưng trong tập huấn luyện,
- σ : độ lệch chuẩn của đặc trưng,
- x' : giá trị sau khi chuẩn hóa

Khi xử lý dữ liệu chuỗi thời gian như giá chứng khoán, đặc trưng đầu vào có thể thay đổi quy mô rất lớn theo thời gian và giữa các mã cổ phiếu. Việc chuẩn hóa bằng Z-score giúp ổn định quá trình huấn luyện, đặc biệt với các mô hình hồi quy hoặc khi dữ liệu gần với phân phối chuẩn. Tuy nhiên, cần đảm bảo rằng việc chuẩn hóa chỉ được thực hiện trên tập huấn luyện để tránh rò rỉ thông tin.

2.4.2 Các chỉ số đo lường hiệu suất mô hình

Để đánh giá mô hình một cách toàn diện, các chỉ số sau đây được sử dụng, phân loại theo từng nhiệm vụ cụ thể mà mô hình thực hiện.

2.4.2.1 Đối với nhiệm vụ dự báo giá (Hồi quy)

- **Sai số toàn phương trung bình (Root Mean Squared Error - RMSE):** Là căn bậc hai của trung bình các bình phương sai số giữa giá trị dự báo và giá trị thực tế. Chỉ số này đặc biệt nhạy cảm và phạt nặng các lỗi dự báo lớn.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.4)$$

- **Sai số tuyệt đối trung bình (Mean Absolute Error - MAE):** Là trung bình của các giá trị tuyệt đối của sai số. Chỉ số này cho biết mức độ lỗi trung bình của mô hình. Giá trị MAE càng thấp, mô hình càng tốt.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.5)$$

- **Hệ số xác định (R-squared - R²):** Đo lường tỷ lệ phần trăm sự biến thiên của biến mục tiêu (giá cổ phiếu) mà mô hình hồi quy có thể giải thích được.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{SSE}{TSS} \quad (2.6)$$

Trong đó, n là tổng số mẫu, y_i là giá trị thực tế, \hat{y}_i là giá trị dự báo, và \bar{y} là giá trị trung bình của các giá trị thực tế. Tử số (SSE - Sum of Squared Errors) là tổng bình phương sai số của mô hình. Mẫu số (TSS - Total Sum of Squares) là tổng bình phương toàn phần.

Giá trị R^2 càng gần 1, mô hình càng phù hợp với dữ liệu. Tuy nhiên, R^2 hoàn toàn có thể nhận giá trị âm. Một giá trị R^2 âm xuất hiện khi $SSE > TSS$, tức là tổng bình phương sai số của mô hình lớn hơn

tổng bình phương toàn phần. Điều này có nghĩa là mô hình dự báo còn tệ hơn một mô hình đơn giản chỉ luôn dự báo bằng giá trị trung bình của dữ liệu. Nguyên nhân có thể đến từ việc lựa chọn mô hình không phù hợp với bản chất của dữ liệu hoặc mô hình đã học phải các quy luật sai lệch (overfitting) từ tập huấn luyện.

2.4.2.2 Đối với nhiệm vụ dự báo xu hướng (Phân loại)

- **Độ chính xác (Accuracy):** Tỷ lệ các mẫu được dự báo đúng trên tổng số mẫu.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.7)$$

- **Độ chuẩn xác (Precision):** Trong số các mẫu được dự báo là một lớp cụ thể (ví dụ: "Up"), có bao nhiêu mẫu thực sự thuộc lớp đó.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.8)$$

- **Độ phủ (Recall / Sensitivity):** Trong số các mẫu thực sự thuộc một lớp cụ thể, có bao nhiêu mẫu được mô hình phát hiện đúng.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.9)$$

- **F1-Score:** Là trung bình điều hòa (harmonic mean) của Precision và Recall, cung cấp một chỉ số duy nhất để đánh giá sự cân bằng giữa hai yếu tố trên.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.10)$$

Trong đó, TP (True Positive), TN (True Negative), FP (False Positive), và FN (False Negative) là các thành phần của ma trận nhầm lẫn.

- **Hệ số tương quan Matthews (Matthews Correlation Coeffi-**

cient - MCC): Là một thước đo chất lượng phân loại đặc biệt hữu ích cho các tập dữ liệu mất cân bằng. Thay vì chỉ xem xét các dự đoán đúng như Accuracy, MCC tổng hợp hiệu suất từ cả bốn thành phần của ma trận nhầm lẫn (TP, TN, FP, FN) vào một giá trị duy nhất. Điều này giúp MCC đưa ra một đánh giá toàn diện và đáng tin cậy hơn, ngay cả khi số lượng mẫu giữa các lớp "Up", "Down" và "Neutral" không đồng đều.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (2.11)$$

Giá trị của MCC dao động từ -1 đến +1. Một giá trị gần +1 thể hiện hiệu suất phân loại rất tốt, trong khi giá trị gần 0 cho thấy mô hình hoạt động không tốt hơn đoán ngẫu nhiên và giá trị gần -1 chỉ ra một sự phân loại nghịch hoàn toàn (mô hình dự đoán ngược lại so với thực tế)

2.4.2.3 Đối với đánh giá chiến lược giao dịch (Backtest)

Đối với đánh giá chiến lược giao dịch (Backtest) Để đánh giá tính thực tiễn và hiệu quả kinh tế của các chiến lược giao dịch, các chỉ số tài chính sau được sử dụng:

- **Lợi nhuận ròng (Net Return):** Đo lường tổng lợi nhuận hoặc thua lỗ của chiến lược trên toàn bộ giai đoạn kiểm thử, được tính theo tỷ lệ phần trăm so với số vốn đầu tư ban đầu.

$$\text{Net Return} = \frac{\text{Vốn}_{\text{cuối kỳ}} - \text{Vốn}_{\text{ban đầu}}}{\text{Vốn}_{\text{ban đầu}}} \quad (2.12)$$

trong đó $\text{Vốn}_{\text{ban đầu}}$ và $\text{Vốn}_{\text{cuối kỳ}}$ lần lượt là tổng tài sản của chiến lược tại thời điểm bắt đầu và kết thúc giai đoạn kiểm thử.

- **Mức sụt giảm tối đa (Maximum Drawdown - MDD):** Là một chỉ số rủi ro quan trọng, đo lường mức lỗ lớn nhất từ một đỉnh (peak) xuống một đáy (trough) của tổng vốn. Chỉ số này cho biết mức tổn

thất tiềm năng tối đa mà chiến lược có thể gây ra.

$$\text{MDD} = \max_t \left(\frac{\text{Vốn tại Dĩnh}_t - \text{Vốn tại Dáy}_t}{\text{Vốn tại Dĩnh}_t} \right) \quad (2.13)$$

- **Tỷ lệ Sharpe (Sharpe Ratio):** Thước đo hiệu suất kinh điển, dùng để đánh giá lợi nhuận đã được điều chỉnh theo rủi ro. Tỷ lệ này cho biết chiến lược tạo ra bao nhiêu lợi nhuận cho mỗi đơn vị rủi ro mà nó gánh chịu (đo bằng độ lệch chuẩn của chuỗi lợi nhuận).

$$\text{Sharpe Ratio} = \frac{E[R_p - R_f]}{\sigma_p} \quad (2.14)$$

trong đó $E[R_p - R_f]$ là lợi nhuận vượt trội trung bình của chiến lược (với lợi nhuận phi rủi ro R_f được giả định bằng 0), và σ_p là độ lệch chuẩn của chuỗi lợi nhuận đó.

2.4.3 Đánh giá độ tin cậy qua Khoảng dự báo (Prediction Interval)

Một dự báo hoàn hảo sẽ chỉ đưa ra một con số duy nhất, ví dụ "giá ngày mai là 15,000 VNĐ". Tuy nhiên, trong thực tế, mọi mô hình đều có sai số và các yếu tố bất định. Vì vậy, một dự báo thực sự hữu ích cần phải đi kèm với một thước đo về độ tin cậy của nó.

Thay vì chỉ đưa ra một con số, Prediction Interval cung cấp một khoảng giá trị, ví dụ "giá ngày mai được dự báo sẽ nằm trong khoảng từ 14,500 đến 15,500 VNĐ với độ tin cậy 95%". Khoảng giá trị này giúp người dùng hình dung được mức độ dao động tiềm năng và đưa ra quyết định một cách thận trọng hơn.

Trong đồ án này, Prediction Interval được xây dựng dựa trên một nguyên tắc thực tế: "Mức độ sai sót của mô hình trong tương lai sẽ tương tự như mức độ sai sót của nó trong quá khứ". Quy trình được thực hiện một cách trực quan như sau:

1. **Đo lường sai sót lịch sử:** Đầu tiên, cho mô hình chạy thử

trên tập dữ liệu validation, một tập dữ liệu mà mô hình chưa từng thấy trong quá trình huấn luyện. Tại mỗi điểm, ghi lại sự chênh lệch giữa giá trị dự báo và giá trị thực tế.

2. **Tìm ra một con số đại diện:** Từ tập hợp các sai sót đã thu thập, tính toán *độ lệch chuẩn* của chúng, ký hiệu là $\hat{\sigma}$. Con số này đại diện cho "độ phân tán trung bình" của sai số. Nếu $\hat{\sigma}$ lớn, mô hình có xu hướng dự báo với sai số dao động mạnh; nếu nhỏ, các dự báo thường khá nhất quán.
3. **Xây dựng khoảng cho tương lai:** Cuối cùng, khi đưa ra một dự báo mới, lấy giá trị dự báo điểm và tạo ra một khoảng bao quanh nó. Dựa trên thuộc tính của phân phối chuẩn, việc cộng và trừ khoảng 2 lần độ lệch chuẩn ($1.96 \cdot \hat{\sigma}$) sẽ tạo ra một khoảng chứa tới 95% các khả năng có thể xảy ra. Công thức được áp dụng là:

$$\text{Khoảng Dự Báo } 95\% = \text{Giá trị dự báo} \pm 1.96 \cdot \hat{\sigma} \quad (2.15)$$

Bằng cách này, mỗi dự báo không chỉ là một con số cố định mà còn đi kèm một "vùng an toàn", cung cấp cho người dùng một cái nhìn sâu sắc hơn về độ tin cậy và rủi ro tiềm ẩn.

2.5 Các mô hình và thuật toán liên quan

2.5.1 Mạng nơ-ron tích chập (CNN)

CNN, hay Convolutional Neural Network (Mạng nơ-ron tích chập), là một loại mạng nơ-ron nhân tạo được sử dụng chủ yếu trong xử lý hình ảnh và thị giác máy tính. CNN có khả năng tự động học và trích xuất các đặc trưng quan trọng từ dữ liệu hình ảnh, giúp nhận dạng và phân loại đối tượng, khuôn mặt, và nhiều tác vụ khác. CNN được sử dụng rộng rãi trong nhiều lĩnh vực như:

- Nhận dạng hình ảnh: Phân loại và nhận diện các đối tượng trong hình ảnh, như nhận diện khuôn mặt, biển số xe, vật thể trong ảnh.

- Xử lý video: Phân tích và nhận diện các hành động, sự kiện trong video.
- Thị giác máy tính: Phát triển các hệ thống tự lái, robot, và các ứng dụng khác liên quan đến thị giác máy tính.

Tuy nhiên, với chuỗi thời gian, CNN 1 chiều (1D-CNN) được dùng để trích xuất các mẫu cục bộ trong chuỗi, chẳng hạn như xu hướng tăng hoặc giảm trong ngắn hạn. CNN đặc biệt hiệu quả trong việc phát hiện các “mẫu lặp” theo thời gian như sóng giá hoặc tín hiệu kỹ thuật. Cấu trúc của CNN bao gồm[6] ⁷:

- Lớp tích chập (Convolutional Layer): Đây là lớp cốt lõi của CNN, sử dụng các bộ lọc (filters) để trích xuất các đặc trưng từ hình ảnh đầu vào.
- Lớp Pooling: Giúp giảm kích thước của dữ liệu đầu ra, tăng hiệu quả tính toán và giảm thiểu overfitting.
- Lớp Fully Connected: Thực hiện phân loại dựa trên các đặc trưng đã được trích xuất.
- Ngoài ra còn các lớp khác như: lớp phi tuyến(activation), lớp chuẩn hóa, lớp dropout được thiết kế tùy theo thiết kế mô hình thuật toán.

Trong bài toán xử lý chuỗi thời gian đa biến, dữ liệu đầu vào có dạng:

$$X \in \mathbb{R}^{T \times d}$$

Trong đó:

- T : số bước thời gian
- d : số đặc trưng tại mỗi bước thời gian

Mỗi dòng $x_t \in \mathbb{R}^d$ biểu diễn tập hợp đặc trưng của ngày t

Mạng CNN sử dụng convolution 1 chiều (1D Convolution) để trích xuất các mẫu cục bộ theo trực thời gian. Kernel (bộ lọc) $w \in \mathbb{R}^{k \times d}$ trượt theo

⁷<https://www.deeplearningbook.org/contents/convnets.html>

chiều thời gian với bước stride, tại mỗi vị trí thực hiện tích hợp chập với đoạn chuỗi con $X_{t:t+k-1}$ (gồm k bước thời gian tiếp theo), tạo ra đầu ra:

$$s(t) = \sum_{i=0}^{k-1} w_i \cdot x_{t+i} \quad \text{với } t \in [0, T - k] \quad (2.16)$$

Sau đó, áp dụng các hàm activation(ReLU): $h(t) = \max(0, s(t) + b)$. Tiếp theo có thể áp dụng Max Pooling hoặc Average Pooling để giảm chiều và làm nổi bật đặc trưng chính.

CNN hoạt động như một công cụ trích đặc trưng mạnh mẽ, có thể phát hiện các xu hướng cục bộ. Khi kết hợp với GRU hoặc LSTM, CNN giúp mô hình học được các đặc điểm ngắn hạn, trong khi khối RNN học mối quan hệ dài hạn – đây là thiết kế thường thấy trong dự báo chuỗi tài chính.

2.5.2 Mạng LSTM và kiến trúc Stacked LSTM

2.5.2.1 Mạng Long Short-Term Memory (LSTM)

LSTM (Long Short-Term Memory) là một loại kiến trúc mạng nơ-ron hồi quy (RNN) đặc biệt được thiết kế để giải quyết vấn đề về phụ thuộc dài hạn trong dữ liệu chuỗi. Nó có khả năng ghi nhớ thông tin trong khoảng thời gian dài, giúp khắc phục hạn chế của RNN truyền thống trong việc xử lý các chuỗi dữ liệu có độ dài lớn.

Mỗi khối LSTM bao gồm ba “cổng” (gates) quan trọng giúp kiểm soát dòng thông tin: forget gate, input gate, và output gate. Ngoài ra, LSTM duy trì một trạng thái ô nhớ (cell state) C_t và một trạng thái ẩn h_t tại mỗi thời điểm t . Cơ chế hoạt động của LSTM là tại bước thời gian t với đầu vào x_t , trạng thái ẩn h_{t-1} và trạng thái ô nhớ c_{t-1} . Các bước của một đơn

vì LSTM gồm:[7][13]

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (\text{forget gate}) \quad (2.17)$$

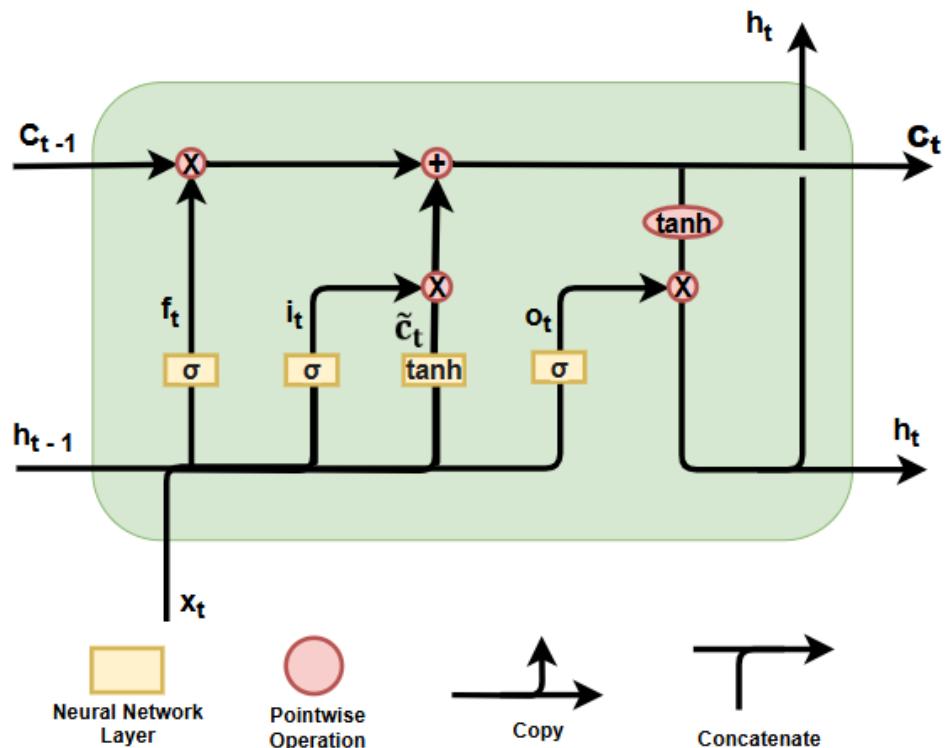
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (\text{input gate}) \quad (2.18)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (\text{candidate}) \quad (2.19)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (\text{new memory}) \quad (2.20)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (\text{output gate}) \quad (2.21)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (\text{new state}) \quad (2.22)$$



Hình 2.1: Kiến trúc mô hình LSTM.

Chức năng của các cổng:

- Forget gate f_t : quyết định thông tin nào từ trạng thái ô nhớ C_{t-1} nên bị loại bỏ.
- Input gate i_t : xác định thông tin mới nào sẽ được thêm vào trạng thái ô nhớ.
- Candidate state \tilde{C}_t : là thông tin ứng viên được đưa vào ô nhớ.

- Output gate o_t : điều khiển phần thông tin nào từ trạng thái ô nhớ sẽ được đưa ra làm đầu ra ẩn h_t .

Trong bài toán dự báo giá chứng khoán, LSTM đặc biệt hữu ích vì có thể học được các xu hướng dài hạn và chu kỳ giá, đồng thời tích hợp thông tin từ nhiều chỉ số kỹ thuật và dữ liệu đa biến. Khi được kết hợp với các tầng CNN (để trích xuất đặc trưng cục bộ) hoặc Attention (để học trọng số theo thời gian), LSTM giúp tăng đáng kể hiệu quả dự báo.

2.5.2.2 Kiến trúc Stacked LSTM

Để nắm bắt các mẫu phức tạp và các đặc trưng có tính phân cấp (hierarchical features) trong chuỗi thời gian, một kiến trúc phổ biến là xếp chồng nhiều lớp LSTM lên nhau, được gọi là **Stacked LSTM (hay Deep LSTM)**⁸. Trong kiến trúc này, chuỗi các trạng thái ẩn đầu ra của lớp LSTM thứ l , $\{h_t^{(l)}\}$, sẽ trở thành chuỗi đầu vào cho lớp LSTM thứ $l + 1$.

Việc xếp chồng các lớp cho phép mô hình học được các biểu diễn dữ liệu ở nhiều mức độ trừu tượng khác nhau. Các lớp thấp hơn có thể tập trung vào việc học các mẫu ngắn hạn, cục bộ, trong khi các lớp cao hơn có thể tổng hợp thông tin từ các lớp dưới để nhận diện các xu hướng dài hạn và phức tạp hơn. Do khả năng mô hình hóa mạnh mẽ và đã được kiểm chứng hiệu quả, Stacked LSTM thường được chọn làm một **mô hình cơ sở (baseline model)** mạnh mẽ trong các bài toán dự báo chuỗi thời gian tài chính để so sánh và đánh giá hiệu quả của các kiến trúc mới.

2.5.3 Mạng Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) là một trong những kiến trúc mạng nơ-ron hồi tiếp tiên tiến, đặc biệt hiệu quả trong việc xử lý dữ liệu tuần tự. Nhờ vào khả năng nhớ và quên thông minh thông qua các cổng điều khiển, GRU giúp giải quyết vấn đề độ dốc biến mất (vanishing gradient) mà các mạng nơ-ron truyền thống thường gặp phải.

Không giống như LSTM, có ba cổng riêng biệt (đầu vào, quên và đầu ra), nhờ đó GRU đơn giản hơn, học nhanh hơn nhưng vẫn đạt hiệu quả gần

⁸<https://machinelearningmastery.com/stacked-long-short-term-memory-networks/>

tương đương, đặc biệt khi dữ liệu không quá phức tạp. GRU chỉ sử dụng hai cổng: cổng Update và cổng Reset.

1. Cổng Update: Cổng này xác định lượng thông tin trong quá khứ (trạng thái ẩn trước đó) sẽ được chuyển tiếp đến trạng thái tương lai. Nó giúp mô hình quyết định lượng bộ nhớ hiện có cần giữ lại.
2. Cổng Reset: Cổng này quyết định lượng thông tin trong quá khứ cần quên trước khi tính toán trạng thái ẩn ứng viên mới. Nó kiểm soát cách đầu vào mới tương tác với bộ nhớ trước đó.

GRU bỏ đi trạng thái ô nhớ riêng biệt, thay vào đó chỉ duy trì trạng thái ẩn h_t – nhờ vậy giảm số lượng tham số đáng kể.

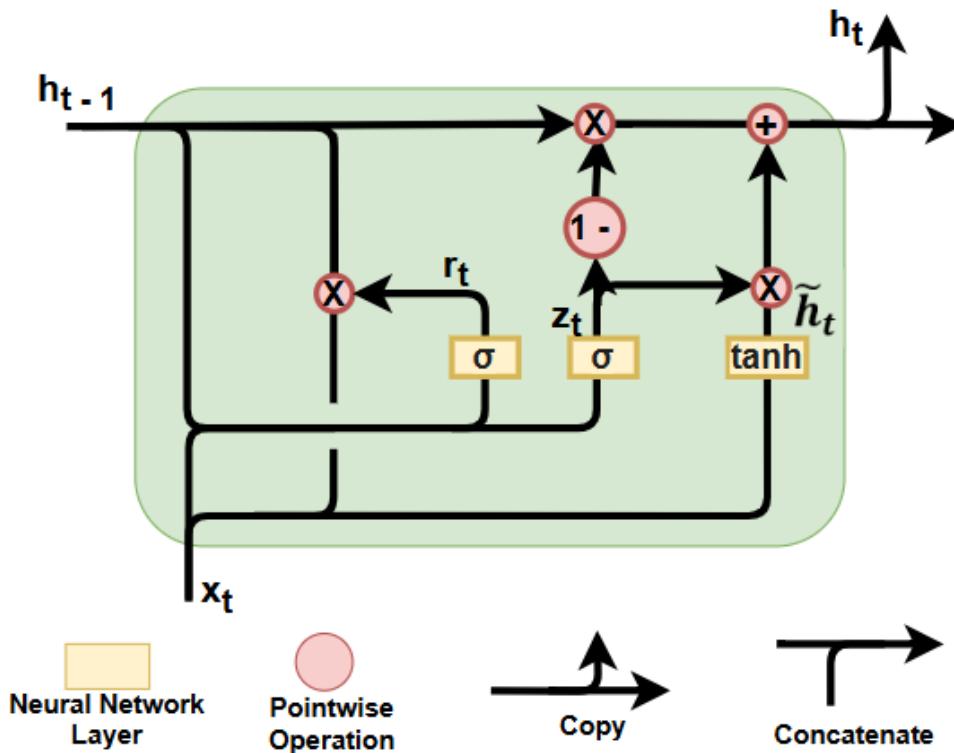
Tại bước thời gian t [6][13]:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (\text{update gate}) \quad (2.23)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (\text{reset gate}) \quad (2.24)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \quad (2.25)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (2.26)$$



Hình 2.2: Kiến trúc mô hình GRU.

GRU đóng vai trò như một bộ lọc thông minh trong mạng hồi tiếp, giúp mô hình học sâu nhớ điều nên nhớ và quên điều nên quên, từ đó nâng cao hiệu quả trong việc dự đoán các chuỗi phức tạp như giá chứng khoán, chuỗi cảm biến, hay chuỗi văn bản.

2.5.4 Cơ chế Chú ý (Attention Mechanism)

Cơ chế Chú ý (Attention Mechanism) là một kỹ thuật đột phá trong học sâu, có nguồn gốc từ bài báo "Attention Is All You Need" của Vaswani và các cộng sự (2017), và đã trở thành nền tảng cho các mô hình Transformer hiện đại [12]. Ý tưởng cốt lõi của nó là mô phỏng khả năng của con người trong việc tập trung vào những thông tin quan trọng và bỏ qua những chi tiết không liên quan. Thay vì ép một mô hình RNN phải nén toàn bộ thông tin của một chuỗi dài vào một vector trạng thái ẩn duy nhất—một quá trình có thể làm mất mát thông tin quan trọng ở quá khứ—cơ chế chú ý cho phép mô hình “nhìn lại” toàn bộ chuỗi đầu vào tại mỗi bước dự báo.

Nói một cách đơn giản, cơ chế này cho phép mô hình tự học cách gán

một trọng số chú ý cho mỗi phần tử trong chuỗi đầu vào. Các phần tử có trọng số cao hơn sẽ có ảnh hưởng lớn hơn đến kết quả đầu ra. Quá trình này giúp mô hình trở nên linh hoạt và hiệu quả hơn, đặc biệt với các chuỗi dài và phức tạp như dữ liệu tài chính.

Scaled Dot-Product Attention

Cốt lõi của cơ chế chú ý là công thức Scaled Dot-Product Attention, một cơ chế tính toán hiệu quả để xác định mối quan hệ giữa các phần tử trong một chuỗi.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) V \quad (2.27)$$

Trong đó, các đầu vào là ba ma trận:

- **Query (Q):** Một ma trận chứa một tập hợp các vector "truy vấn". Mỗi vector truy vấn đại diện cho một phần tử trong chuỗi đang cần tìm thông tin liên quan từ các phần tử khác.
- **Key (K):** Một ma trận chứa các vector "khóa". Mỗi vector khóa tương ứng với một phần tử trong chuỗi và hoạt động như một "nhãn" để có thể được so khớp với các truy vấn.
- **Value (V):** Một ma trận chứa các vector "giá trị". Mỗi vector giá trị chứa thông tin thực tế của phần tử tương ứng với một vector khóa.

Quá trình tính toán bao gồm:

1. **Tính điểm tương đồng (QK^\top):** Tích vô hướng được tính giữa mỗi vector Query và tất cả các vector Key để tạo ra một ma trận điểm số, đo lường mức độ tương hợp giữa chúng.
2. **Chuẩn hóa (Scaling):** Các điểm số được chia cho $\sqrt{d_k}$ (với d_k là chiều của vector khóa) để ổn định gradient trong quá trình huấn luyện.
3. **Tính trọng số (softmax):** Hàm softmax được áp dụng lên các điểm số đã chuẩn hóa để biến chúng thành một phân phối xác suất, hay còn gọi là các trọng số chú ý.

4. **Tổng hợp đầu ra:** Đầu ra cuối cùng là một tổng có trọng số của các vector Value, trong đó mỗi vector Value được nhân với trọng số chú ý tương ứng của nó.

Multi-Head Attention

Thay vì thực hiện một phép tính chú ý duy nhất, **Multi-Head Attention** cho phép mô hình đồng thời tập trung vào thông tin từ các không gian biểu diễn con khác nhau. Kiến trúc này bao gồm hai bước chính:

1. **Tính toán song song các “head”:** Trước tiên, mô hình tính toán song song h “đầu” chú ý (attention heads). Mỗi “đầu” là một phép tính Scaled Dot-Product Attention riêng, nhưng hoạt động trên một phiên bản đã được biến đổi tuyến tính của Q , K , và V :

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2.28)$$

trong đó (W_i^Q, W_i^K, W_i^V) là các ma trận trọng số riêng biệt cho mỗi “đầu” thứ i . Việc chiếu Q , K , V vào các không gian con khác nhau cho phép mỗi “đầu” học các kiểu phụ thuộc khác nhau của dữ liệu (ví dụ: các mẫu hình ngắn hạn, dài hạn, hoặc theo chu kỳ).

2. **Ghép và biến đổi đầu ra:** Sau khi có kết quả từ tất cả các “đầu”, chúng được ghép lại (concatenated) và đi qua một phép biến đổi tuyến tính cuối cùng để tạo ra biểu diễn cuối cùng:

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O \quad (2.29)$$

Bằng cách này, Multi-Head Attention giúp mô hình có được một cái nhìn đa chiều và phong phú hơn về dữ liệu so với việc chỉ sử dụng một cơ chế chú ý đơn lẻ.

2.5.5 Học Đa nhiệm (Multi-task Learning)

a. Nguyên lý và Động lực áp dụng trong Đồ án

Học Đa nhiệm (Multi-task Learning - MTL) là một phương pháp huấn luyện trong đó một mô hình duy nhất được thiết kế để giải quyết đồng thời nhiều nhiệm vụ có liên quan, thay vì xây dựng các mô hình riêng lẻ [3]. Nguyên lý nền tảng của MTL là ý tưởng về **chuyển giao học tập quy nạp (inductive transfer)** thông qua việc **chia sẻ biểu diễn (shared representation)**. Theo bài báo kinh điển của Caruana (1997), tín hiệu học được từ các **nhiệm vụ phụ (auxiliary tasks)** hoạt động như một dạng **điều chỉnh ngầm (implicit regularization)**. Điều này buộc mô hình phải học được các biểu diễn chung, mạnh mẽ và ít bị quá khớp (overfitting) hơn, đặc biệt hữu ích khi dữ liệu nhiều và hạn chế.

Trong lĩnh vực dự báo tài chính, dữ liệu giá cổ phiếu vốn dĩ rất nhiều và phi ổn định (non-stationary). Một mô hình chỉ tập trung vào nhiệm vụ hồi quy giá trị có thể dễ dàng học phải các quy luật sai lệch từ nhiều. Do đó, việc áp dụng MTL bằng cách thêm vào một nhiệm vụ phụ có liên quan là một chiến lược hiệu quả để cải thiện độ tin cậy của mô hình. Trong đồ án này, chúng tôi đề xuất một mô hình MTL giải quyết đồng thời hai nhiệm vụ có mối quan hệ nhân quả:

1. **Nhiệm vụ Hồi quy (Regression):** Dự báo giá cổ phiếu (được xem là nhiệm vụ chính).
2. **Nhiệm vụ Phân loại (Classification):** Dự báo xu hướng của giá cổ phiếu (Up, Down, Neutral) (được xem là nhiệm vụ phụ).

Hướng tiếp cận kết hợp hồi quy và phân loại này đã được triển khai trong nhiều nghiên cứu gần đây [2],[4] nơi các tác giả nhận thấy rằng việc học đồng thời giúp đạt được hiệu suất cao.

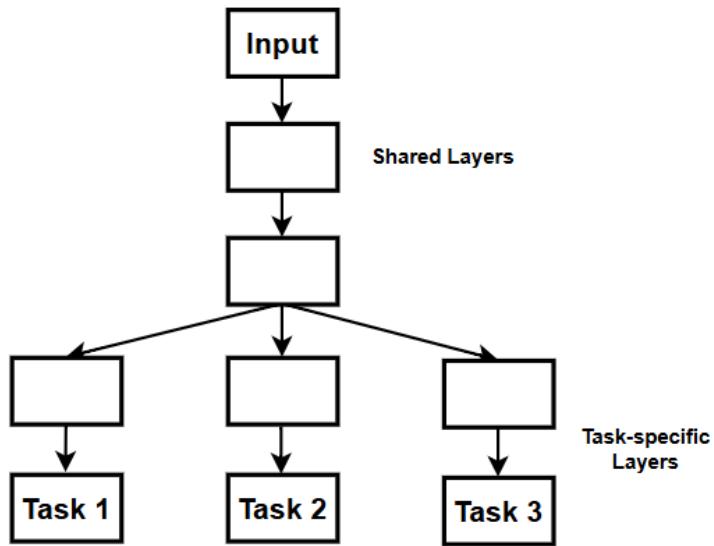
b. Các kiến trúc Học Đa nhiệm

Có hai kiến trúc chính để triển khai việc chia sẻ thông tin trong MTL: *Hard Parameter Sharing* và *Soft Parameter Sharing* [11]. Lựa chọn kiến

trúc phụ thuộc vào mức độ liên quan giữa các nhiệm vụ và sự linh hoạt mong muốn của mô hình.

Hard Parameter Sharing

Đây là kiến trúc phổ biến và được sử dụng rộng rãi nhất trong học sâu, đồng thời cũng là kiến trúc được áp dụng trong đồ án này. Trong kiến trúc này, mô hình được chia thành hai phần rõ rệt:



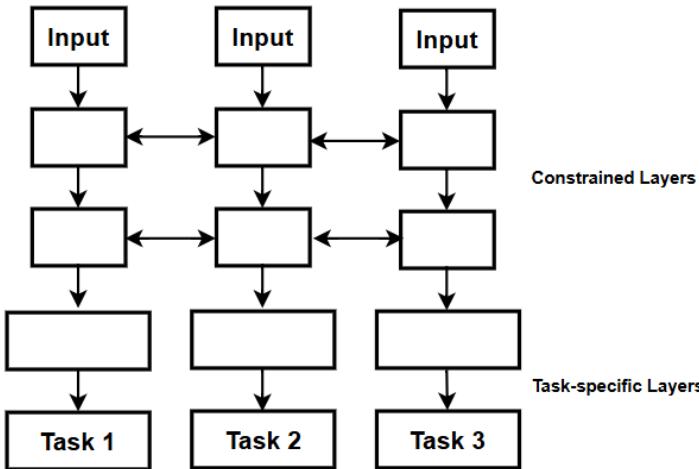
Hình 2.3: Kiến trúc Hard Parameter Sharing trong Học Đa nhiệm.

- **Các tầng chia sẻ (Shared Layers):** Một tập hợp các tầng ẩn được chia sẻ giữa tất cả các nhiệm vụ. Các tầng này thường nằm ở phần đầu của mạng (được gọi là backbone hoặc encoder), có nhiệm vụ học các đặc trưng chung từ dữ liệu đầu vào. Trong mô hình của chúng tôi, đây là toàn bộ phần mã hóa của DA-RNN.
- **Các tầng chuyên biệt (Task-Specific Layers):** Từ đầu ra của khối chia sẻ, mô hình sẽ nhánh thành nhiều “đầu ra” (output heads). Mỗi nhánh là một hoặc nhiều tầng chuyên biệt, có bộ tham số riêng và chỉ phục vụ cho một nhiệm vụ duy nhất.

Ưu điểm lớn của phương pháp này là nó giảm đáng kể nguy cơ quá khớp, vì các nhiệm vụ cùng nhau điều chỉnh một bộ tham số chung, buộc chúng phải tìm ra một không gian biểu diễn chung hữu ích. Tuy nhiên, hiệu quả của nó phụ thuộc nhiều vào mức độ liên quan giữa các tác vụ.

Soft Parameter Sharing

Khác với kiến trúc Hard Parameter Sharing, phương pháp Soft Parameter Sharing áp dụng một cách tiếp cận linh hoạt hơn. Trong kiến trúc này, mỗi nhiệm vụ sở hữu mô hình và bộ tham số của riêng mình. Thay vì ép buộc các tầng phải chia sẻ hoàn toàn tham số, cơ chế này khuyến khích các bộ tham số của các mô hình trở nên *tương tự* nhau.



Hình 2.4: Kiến trúc Soft Parameter Sharing trong Học Đa nhiệm.

Việc này thường được thực hiện bằng cách thêm một thành phần điều chuẩn (regularization term) vào hàm mất mát tổng thể, ví dụ như chuẩn L2 của sự khác biệt giữa các tham số của các mô hình: $\mathcal{L}_{\text{reg}} = \sum_{i,j} \|\theta_i - \theta_j\|_2^2$. Thành phần này sẽ “phạt” mô hình nếu các tham số của các tác vụ khác biệt quá nhiều.

c. Lý giải về cơ chế Học Đa nhiệm và Hàm mất mát

Cơ chế Tương tác giữa các Nhiệm vụ: Sự thành công của phương pháp MTL trong đồ án này nằm ở mối quan hệ tương tác giữa hai nhiệm vụ, dựa trên nguyên lý **chuyển giao học tập quy nạp** đã được giới thiệu.

1. **Nhiệm vụ Phân loại đóng vai trò Điều chuẩn ngầm (Implicit Regularization):** Như đã đề cập, nhiệm vụ phụ (auxiliary task) có thể hoạt động như một cơ chế điều chuẩn [11]. Trong trường hợp này, nhiệm vụ phân loại xu hướng (Up/Down/Neutral) buộc mô hình phải

học các biểu diễn có tính tổng quát cao hơn, có khả năng nắm bắt được “trạng thái” của thị trường thay vì chỉ tập trung vào các giá trị số cụ thể. Tín hiệu học được từ nhiệm vụ này giúp mô hình hồi quy chính tránh bị (overfitting) với các nhiễu ngẫu nhiên trong dữ liệu giá.

2. **Nhiệm vụ Hồi quy cung cấp thông tin về Độ lớn (Magnitude):** Ngược lại, nhiệm vụ hồi quy cung cấp một thông tin quan trọng mà nhiệm vụ phân loại không có: “độ lớn” của sự thay đổi giá. Thông tin định lượng này giúp mô hình phân loại đưa ra quyết định chắc chắn hơn. Ví dụ, một dự báo hồi quy cho thấy giá trị cổ phiếu thay đổi dương lớn sẽ là một tín hiệu mạnh mẽ cung cấp cho dự báo phân loại là “Up”.

Bằng cách này, hai nhiệm vụ không cạnh tranh mà bổ trợ lẫn nhau, giúp mô hình học được một biểu diễn chung mạnh mẽ và toàn diện hơn về động lực thị trường.

Hàm mất mát và Lựa chọn siêu tham số: Trong Học Đa nhiệm, hàm mất mát tổng thể thường được định nghĩa là một tổng có trọng số của các hàm mất mát riêng lẻ từ mỗi tác vụ. Công thức tổng quát cho một mô hình có T nhiệm vụ là:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^T \omega_i \cdot \mathcal{L}_i \quad (2.30)$$

Trong đó \mathcal{L}_i là hàm mất mát của nhiệm vụ thứ i và ω_i là trọng số tương ứng để cân bằng sự đóng góp của nhiệm vụ đó.

Áp dụng vào mô hình MT-DA-RNN trong đồ án này với hai nhiệm vụ là hồi quy và phân loại, hàm mất mát cụ thể được định nghĩa như sau, dựa trên mã nguồn thực nghiệm:

$$\mathcal{L}_{\text{total}} = \omega_{\text{reg}} \cdot \mathcal{L}_{\text{reg}} + \omega_{\text{cls}} \cdot \mathcal{L}_{\text{cls}} \quad (2.31)$$

Trong đó:

- \mathcal{L}_{reg} là hàm mất mát **Mean Squared Error (MSE)** cho nhiệm vụ

hồi quy.

- \mathcal{L}_{cls} là hàm mất mát **Cross-Entropy Loss** có trọng số để xử lý mất cân bằng lớp.
- ω_{reg} và ω_{cls} là các siêu tham số trọng số, tương ứng với các biến LOSS_WEIGHT_REGRESSION và LOSS_WEIGHT_CLASSIFICATION trong mã nguồn. Chúng đóng hai vai trò kỹ thuật quan trọng:
 1. **Cân bằng Thang đo (Scale Balancing):** Giá trị của hàm mất mát MSE và Cross-Entropy thường có thang đo (magnitude) rất khác nhau. Các trọng số ω được sử dụng để điều chỉnh và cân bằng tầm ảnh hưởng của mỗi nhiệm vụ lên quá trình cập nhật gradient tổng thể, tránh trường hợp một nhiệm vụ lấn át hoàn toàn nhiệm vụ còn lại.
 2. **Phản ánh Mức độ ưu tiên của Nhiệm vụ (Task Priority):** Các trọng số cũng cho phép chúng ta định hướng quá trình học theo mục tiêu của bài toán. Nếu mục tiêu là chiến lược giao dịch định tính, ta có thể tăng ω_{cls} . Ngược lại, nếu mục tiêu là định giá tài sản, ta sẽ tăng ω_{reg} .

Trong đồ án này, các giá trị ω được xác định thông qua thử nghiệm trên tập kiểm định (validation set). Chúng tôi đã thử nghiệm một số cặp giá trị và lựa chọn cặp mang lại sự cân bằng tốt nhất giữa chỉ số RMSE (cho hồi quy) và Accuracy (cho phân loại).

d. Lợi ích và Ứng dụng trong Đồ án

Việc áp dụng Học Đa nhiệm trong mô hình MT-DA-RNN của chúng tôi mang lại nhiều lợi ích thiết thực cho bài toán dự báo tài chính:

- **Cải thiện khả năng khái quát hóa và giảm quá khớp:** Đây là lợi ích cốt lõi. Bằng cách sử dụng nhiệm vụ phân loại xu hướng như một cơ chế điều chuẩn ngầm, mô hình được định hướng để học các biểu diễn mạnh mẽ, có khả năng chống lại nhiễu tốt hơn, giúp cải thiện hiệu suất trên dữ liệu chưa từng thấy .

- **Tăng cường hiệu quả sử dụng dữ liệu:** Mỗi mẫu dữ liệu được khai thác một cách hiệu quả hơn để cập nhật trọng số thông qua hai tín hiệu mắng mỉa khác nhau, giúp mô hình học nhanh hơn và tốt hơn.
- **Tạo ra kết quả dự báo toàn diện hơn:** Mô hình không chỉ đưa ra một con số dự báo (giá trị), mà còn cung cấp ngữ cảnh về xu hướng (định tính), mang lại một cái nhìn toàn diện và hữu ích hơn cho người dùng cuối.
- **Hiệu quả về mặt tính toán:** Việc huấn luyện một mô hình đa nhiệm duy nhất thay vì hai mô hình riêng biệt giúp tiết kiệm đáng kể tài nguyên tính toán và thời gian huấn luyện.

2.5.6 Mô hình DA-RNN (Dual-Stage Attention-based RNN)

a. Động lực và Ý tưởng cốt lõi

Dual-Stage Attention-based Recurrent Neural Network (DA-RNN) là một kiến trúc Encoder-Decoder tiên tiến do Qin và các cộng sự (2017) đề xuất, nhằm giải quyết các thách thức đặc thù trong bài toán dự báo chuỗi thời gian đa biến.

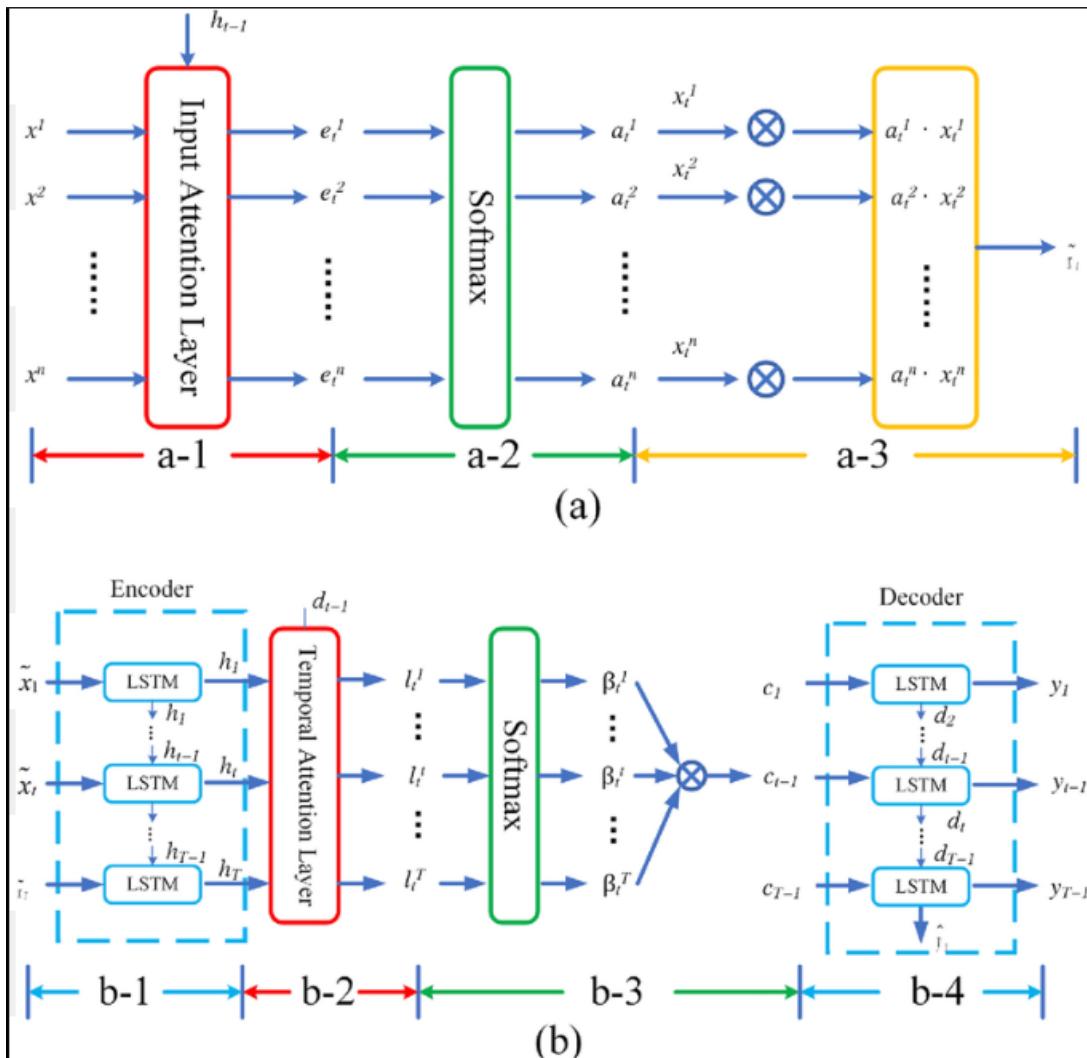
Các mô hình chuỗi thời gian truyền thống như LSTM hay GRU, mặc dù hiệu quả trong việc ghi nhớ tuần tự, thường đối mặt với hai hạn chế lớn khi xử lý dữ liệu phức tạp [10]:

1. **Đối xử bình đẳng với các đặc trưng:** Các mô hình này mặc định rằng tất cả các chuỗi đầu vào (ví dụ: giá đóng cửa, khối lượng, RSI) có tầm quan trọng như nhau tại mọi thời điểm.
2. **Nút thắt cổ chai thông tin (Information Bottleneck):** Toàn bộ thông tin của chuỗi đầu vào dài được nén vào một vector ngữ cảnh có kích thước cố định, khiến thông tin ở các bước thời gian xa trong quá khứ dễ bị lãng quên.

Để khắc phục các nhược điểm này, DA-RNN đưa ra một ý tưởng cốt lõi: mô hình cần có khả năng học cách tự động và linh hoạt lựa chọn thông tin quan trọng theo cả hai chiều: chiều đặc trưng và chiều thời gian.

b. Phân tích chi tiết Kiến trúc theo Hình 2.5

Kiến trúc DA-RNN bao gồm một bộ mã hóa (Encoder) và một bộ giải mã (Decoder), mỗi thành phần được trang bị một cơ chế chú ý riêng, tạo thành một cơ chế chú ý hai giai đoạn.



Hình 2.5: Sơ đồ tổng quát kiến trúc DA-RNN gồm hai cơ chế attention: (a) Cơ chế chú ý đầu vào (Input Attention); (b) Cơ chế chú ý theo thời gian (Temporal Attention) [1].

1. Giai đoạn 1: Cơ chế Chú ý Đầu vào (Input Attention - Hình 2.5a)

Giai đoạn này hoạt động như một “bộ lọc đặc trưng” động, cho phép mô hình tập trung vào các chỉ báo kỹ thuật quan trọng tại mỗi bước thời gian.

- **Đầu vào:** Tại mỗi bước thời gian t , đầu vào là vector đặc trưng thô

$x_t = (x_t^1, x_t^2, \dots, x_t^n)$ và trạng thái ẩn trước đó của bộ mã hóa h_{t-1} .

- **Bước a-1 (Input Attention Layer):** Một mạng nơ-ron nhỏ (thường là một lớp fully-connected) được sử dụng để tính toán một “điểm số quan trọng” e_t^k cho mỗi đặc trưng x_t^k . Quá trình này có tính đến ngữ cảnh thời gian thông qua trạng thái ẩn h_{t-1} .
- **Bước a-2 (Softmax):** Các điểm số e_t^k được đưa qua hàm softmax để chuẩn hóa thành các trọng số chú ý α_t^k . Các trọng số này không âm và có tổng bằng 1, thể hiện sự phân bổ “sự chú ý” cho các đặc trưng tại thời điểm t .
- **Bước a-3 (Tổng hợp có trọng số):** Một vector đặc trưng mới, \tilde{x}_t , được tạo ra bằng cách lấy tổng có trọng số của các đặc trưng ban đầu: $\tilde{x}_t = \sum_{k=1}^n \alpha_t^k x_t^k$.

Kết quả: Thay vì sử dụng x_t , mô hình sẽ sử dụng \tilde{x}_t – một phiên bản đã được “chắt lọc” và tập trung vào các thông tin quan trọng nhất – để đưa vào bộ mã hóa LSTM.

2. Giai đoạn 2: Cơ chế Chú ý theo Thời gian (Temporal Attention - Hình 2.5b)

Giai đoạn này được tích hợp vào bộ giải mã để phá vỡ “nút thắt cổ chai thông tin”, cho phép mô hình nhìn lại và chọn lọc thông tin từ toàn bộ chuỗi quá khứ.

- **Bước b-1 (Encoder LSTM):** Các vector đặc trưng đã được lọc $\tilde{x}_1, \dots, \tilde{x}_T$ được đưa qua một mạng LSTM để tạo ra một chuỗi các trạng thái ẩn của bộ mã hóa $\{h_1, h_2, \dots, h_T\}$. Mỗi h_i tóm tắt thông tin từ đầu chuỗi đến thời điểm i .
- **Bước b-2 (Temporal Attention Layer):** Tại bước dự báo t của bộ giải mã, một mạng nơ-ron khác sẽ tính toán điểm số quan trọng l_t^i cho mỗi trạng thái ẩn h_i của bộ mã hóa. Quá trình này có tính đến ngữ cảnh của bộ giải mã thông qua trạng thái ẩn trước đó của nó là d_{t-1} .
- **Bước b-3 (Softmax và Tổng hợp):** Tương tự, các điểm số l_t^i được chuẩn hóa thành trọng số chú ý theo thời gian β_t^i qua hàm softmax.

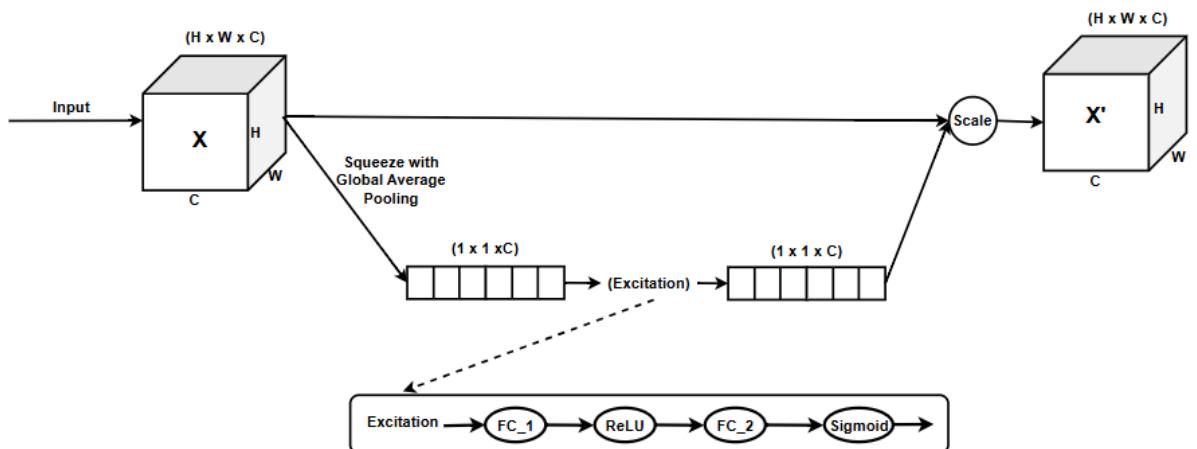
Các trọng số này được dùng để tính toán **vector ngữ cảnh động** (**dynamic context vector**) $c_t = \sum_{i=1}^T \beta_t^i h_i$. Vector c_t là một bản tóm tắt thông minh của toàn bộ chuỗi đầu vào, trong đó những thời điểm quan trọng nhất được nhấn mạnh.

- **Bước b-4 (Decoder LSTM):** Vector ngữ cảnh động c_t này được kết hợp với đầu vào của bộ giải mã và đưa qua mạng LSTM của bộ giải mã để tạo ra trạng thái ẩn mới d_t và đưa ra dự đoán cuối cùng \hat{y}_t .

Kết quả: Bằng cách tạo ra một vector ngữ cảnh c_t mới cho mỗi bước dự báo, mô hình có thể linh hoạt truy cập bất kỳ phần nào của chuỗi quá khứ, giải quyết hiệu quả vấn đề lãng quên thông tin dài hạn.

2.5.7 SeBlock(Squeeze and Excitation Block)

SEBlock là một mô-đun channel-wise attention dùng để tăng cường các đặc trưng có ích và làm giảm nhiễu trong mạng tích chập (CNN) và RNN. Không giống như attention theo không gian hay thời gian, SEBlock hoạt động trên từng kênh (channel) của đầu ra, giúp mô hình học cách gán trọng số động cho từng kênh đầu vào của một lớp convolution.



Hình 2.6: Kiến trúc mô hình SEBlock

Cơ chế này gồm 3 bước chính: [8]:

1. **Squeeze:** Mục tiêu là tổng hợp thông tin toàn cục từ từng kênh, bằng cách áp dụng Global Average Pooling trên mỗi kênh đầu ra của

feature map.

Giả sử đầu vào của SEBlock là một tensor 3D $X \in \mathbb{R}^{H \times W \times C}$, với:

- C: số lượng kênh (channels).
- H, W: chiều cao, chiều rộng của không gian.

Với mỗi kênh c , ta tính giá trị trung bình toàn bộ không gian

$$s_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_{i,j,c} \quad , c = 1 \dots C \quad (2.32)$$

Nói cách khác, ta rút gọn từng cột trong ma trận X thành một giá trị duy nhất, thu được một vector mới: $s_c \in \mathbb{R}^C$.

Vector này biểu diễn tầm quan trọng trung bình của từng kênh qua toàn bộ thời gian.

2. **Excitation:** Có vector đặc trưng $s_c \in \mathbb{R}^C$ ở trên. SeBlock sử dụng một mạng con gồm hai lớp fully-connected để học một hàm phi tuyến từ vector này nhằm sinh ra trọng số cho từng kênh. Quá trình này gồm:

- Giảm chiều từ C xuống C/r thông qua lớp FC1, sau đó áp dụng ReLU để tăng tính phi tuyến.
- Tăng chiều trở lại về C thông qua lớp FC2.
- Cuối cùng dùng sigmoid để giới hạn các giá trị đầu ra trong khoảng $(0, 1)$, tạo thành vector có trọng số:

$$e = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot s)) \quad (2.33)$$

với:

- r là hệ số giảm chiều (thường $r = 8$ hoặc 16) để giảm số lượng tham số cần học.
- $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}, W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$.
- σ là hàm sigmoid để đảm bảo các giá trị đầu ra nằm trong khoảng $(0, 1)$.

- Kết quả $e \in \mathbb{R}^C$ là vector trọng số cho từng kênh.
3. **Scale:** Mỗi kênh trong đầu vào ban đầu X được nhân (element-wise) với trọng số tương ứng từ vector e , tạo ra tensor đầu ra đã được "chú ý":

$$\hat{X}_c = e_c \cdot X_c, \quad \forall c = 1, \dots, C \quad (2.34)$$

Các kênh quan trọng ($e_c \approx 1$ được giữ lại, kênh kém quan trọng ($e_c \rightarrow 0$) sẽ bị làm mờ. Giá trị e_c nằm giữa 0 và 1 giúp “mềm hóa” việc chọn lọc, tức là không loại bỏ hoàn toàn, mà giảm tầm ảnh hưởng tương ứng. Kết quả cuối cùng là chuỗi đã được tăng cường các kênh quan trọng, và giảm nhiễu từ các đặc trưng ít liên quan.

Tóm lại, SEBlock cho phép mô hình học cách “chú ý” tới các đặc trưng quan trọng bằng cách tính toán trọng số cho từng kênh đầu vào. Qua đó, mô hình tập trung hơn vào các yếu tố ảnh hưởng mạnh đến kết quả dự đoán – ví dụ như giá đóng cửa hoặc khối lượng giao dịch – và giảm tác động của các đặc trưng nhiễu hoặc ít liên quan.

2.5.8 Mô hình 3D CNN-GRU

2.5.8.1 Ý tưởng cốt lõi

Trong bối cảnh dự báo chuỗi thời gian tài chính như giá cổ phiếu, mô hình học máy cần xử lý đồng thời nhiều thách thức đặc thù: dữ liệu thường có tính phi tuyến mạnh, biến động thất thường, mang theo yếu tố nhiễu cao, và thể hiện các chu kỳ ngắn hạn lồng ghép trong xu hướng dài hạn. Các phương pháp truyền thống như ARIMA, mô hình học sâu đơn tầng như LSTM, GRU, hay thậm chí các mô hình tự hồi tiếp (RNN) đơn thuần, thường chỉ xử lý tốt một khía cạnh – hoặc nắm bắt mối quan hệ tuần tự theo thời gian, hoặc trích xuất đặc trưng từ tập dữ liệu – nhưng chưa thật sự phát huy hiệu quả khi cả hai khía cạnh đều hiện diện và tương tác mạnh mẽ như trong thị trường chứng khoán.

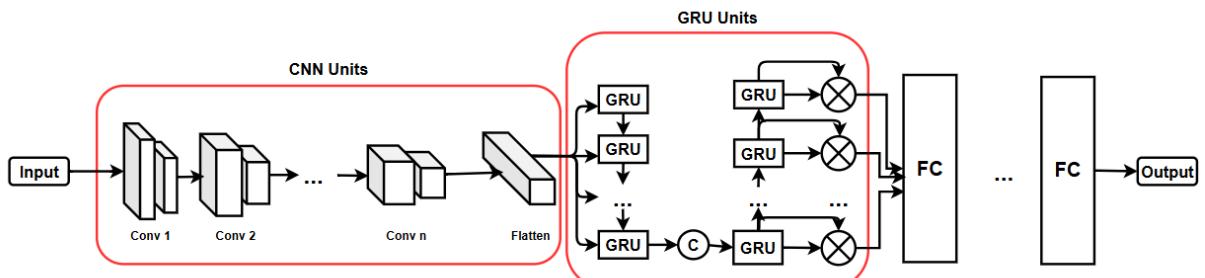
Ý tưởng cốt lõi của mô hình kết hợp CNN-GRU [5][9] là xây dựng một kiến trúc lai (hybrid architecture) có thể đồng thời:

- Trích xuất các đặc trưng cục bộ (local features) từ chuỗi thời gian thông qua các lớp tích chập 1 chiều (1D Convolution),
- Học được mối quan hệ phụ thuộc thời gian giữa các bước trong chuỗi thông qua mạng hồi tiếp GRU.

Sự kết hợp này mang lại lợi ích kép: CNN giúp lọc nhiễu, nhấn mạnh các biến động ngắn hạn có giá trị (ví dụ: xu hướng giá trong vài phiên gần đây), trong khi GRU cung cấp khả năng ghi nhớ và suy luận từ các xu hướng dài hạn (ví dụ: biến động giá trong vài tuần hoặc vài tháng trước). Điều này cho phép mô hình không những học tốt các mẫu dữ liệu phức tạp mà còn hiểu được ngữ cảnh thời gian sâu trong chuỗi dữ liệu đầu vào.

2.5.8.2 Kiến trúc mô hình

Hình 2.7 mô tả cấu trúc tổng thể của mô hình baseline 3D-CNN-GRU được sử dụng trong đề tài, gồm ba phần chính: khôi trích xuất đặc trưng CNN, khôi xử lý tuần tự GRU, và khôi suy luận đầu ra Fully Connected:



Hình 2.7: Sơ đồ kiến trúc mô hình 3D CNN - GRU

(1) **Khôi CNN – Trích xuất đặc trưng ngắn hạn.** Đầu vào của mô hình là chuỗi thời gian đa biến có kích thước $[T, D]$, trong đó:

- T : số bước thời gian trong cửa sổ trượt (sliding window).
- D : số đặc trưng tại mỗi thời điểm, chẳng hạn như giá đóng cửa, khối lượng giao dịch, chỉ báo kỹ thuật (RSI, MACD, v.v.)

Trước tiên, dữ liệu đầu vào được đưa qua một chuỗi các lớp tích chập 1 chiều (1D Convolution), ký hiệu là Conv1, Conv2, ..., Convn. Mỗi

lớp tích chập sử dụng các bộ lọc (filters) khác nhau nhằm tự động học các mẫu đặc trưng (pattern) xuất hiện trong chuỗi thời gian. Các bộ lọc này có khả năng phát hiện những xu hướng như tăng trưởng mạnh, sự đảo chiều, hoặc các giai đoạn tích lũy của giá.

Thông qua mỗi lớp tích chập, biểu diễn đầu ra dần được trừu tượng hóa, giúp làm nổi bật các tín hiệu quan trọng ngắn hạn và giảm nhiễu. Mục tiêu là thu được một biểu diễn giàu thông tin hơn phục vụ cho việc học sâu hơn ở các tầng sau.

Cuối cùng, một lớp Flatten được áp dụng để chuyển đổi đầu ra từ tensor 3 chiều sang dạng 2 chiều với kích thước $[T', D']$, tạo điều kiện phù hợp để đưa vào tầng GRU tiếp theo.

- (2) **Khối GRU – Học quan hệ theo thời gian.** Sau khi qua khối CNN, chuỗi đặc trưng đã được làm nổi bật sẽ được đưa vào một chuỗi các tầng GRU (Gated Recurrent Unit). GRU là một biến thể nhẹ hơn của LSTM, với cấu trúc đơn giản hơn nhưng vẫn giữ được khả năng ghi nhớ dài hạn, nhờ đó giúp giảm chi phí tính toán và thời gian huấn luyện.

Các tầng GRU được tổ chức theo dạng xếp chồng (stacked GRU), cho phép mô hình học được các biểu diễn phức tạp hơn theo chiều thời gian. Ở mỗi bước thời gian, GRU cập nhật trạng thái ẩn (hidden state) dựa trên thông tin đầu vào hiện tại và trạng thái trước đó.

GRU hoạt động dựa trên hai cổng chính:

- Cổng cập nhật (update gate): kiểm soát lượng thông tin cần lưu giữ từ bước trước.
- Cổng đặt lại (reset gate): quyết định bao nhiêu thông tin cũ nên bị lãng quên.

Sự linh hoạt này giúp mô hình thích ứng tốt với các chuỗi có tính chất dao động, chu kỳ khác nhau — đặc biệt quan trọng trong bài toán tài chính.

- (3) **Khối Fully Connected – Dự đoán đầu ra.** Trạng thái ẩn cuối cùng của GRU, hoặc toàn bộ chuỗi trạng thái (nếu sử dụng cơ chế

attention hoặc pooling), sẽ được đưa vào các lớp Fully Connected (FC) để tiến hành suy luận đầu ra.

Vai trò của khối FC là tổng hợp toàn bộ thông tin đã học từ các tầng trước nhằm đưa ra dự đoán cuối cùng, phù hợp với bài toán cụ thể:

- Đối với bài toán hồi quy: mô hình dự đoán giá cổ phiếu tại bước thời gian tiếp theo (next-step prediction).
- Đối với bài toán phân loại xu hướng: mô hình xuất ra xác suất thuộc các lớp tăng / giảm / giữ giá, sử dụng hàm kích hoạt Softmax hoặc Sigmoid để chuẩn hóa đầu ra.

Ngoài ra, các tầng FC còn có thể kết hợp với kỹ thuật Dropout nhằm giảm hiện tượng overfitting, hoặc Batch Normalization để tăng tốc độ hội tụ và ổn định quá trình huấn luyện.

Mô hình 3D-CNN-GRU sở hữu nhiều ưu điểm nổi bật trong việc xử lý dữ liệu chuỗi thời gian có cấu trúc không gian, chẳng hạn như dữ liệu môi trường đo tại nhiều trạm quan trắc. Nhờ sự kết hợp giữa mạng tích chập (CNN) và mạng hồi tiếp có cỗng (GRU), mô hình có thể đồng thời khai thác các đặc trưng không gian và thời gian, từ đó cải thiện chất lượng dự báo. GRU với cấu trúc đơn giản hơn LSTM giúp giảm thiểu độ phức tạp và thời gian huấn luyện trong khi vẫn giữ được khả năng ghi nhớ thông tin dài hạn. Đồng thời, CNN có khả năng tự động trích xuất đặc trưng từ dữ liệu đầu vào, giảm sự phụ thuộc vào các kỹ thuật xử lý thủ công và phù hợp với dữ liệu đa chiều

Tuy nhiên, mô hình cũng tồn tại một số nhược điểm. Do mang bản chất là một mô hình học sâu, 3D CNN-GRU hoạt động như một “hộp đen” nên khó giải thích rõ ràng quá trình ra quyết định, điều này gây hạn chế trong các lĩnh vực yêu cầu tính minh bạch cao. Ngoài ra, với số lượng tham số lớn, mô hình dễ bị quá khớp nếu không có đủ dữ liệu huấn luyện hoặc không áp dụng các kỹ thuật chính quy hóa thích hợp. Bên cạnh đó, mô hình trong kiến trúc cơ bản chưa thể hiện rõ khả năng khai thác tương tác giữa các trạm đo, điều này có thể giới hạn hiệu quả trong những bài toán mà mối liên hệ không gian giữa các trạm đóng vai trò quan trọng.

Chương 3

Kiến trúc hệ thống và phương pháp đề xuất

Chương này trình bày chi tiết về phương pháp luận và các bước thực thi của đồ án. Nội dung bao gồm các thiết lập nền tảng chung, sau đó đi sâu vào thiết kế và quy trình chuẩn bị dữ liệu riêng biệt cho từng hướng tiếp cận mô hình.

3.1 Thiết lập thực nghiệm chung

Phần này mô tả các yếu tố nền tảng được áp dụng chung cho cả hai hướng tiếp cận mô hình, bao gồm môi trường kỹ thuật, kiến trúc thu thập dữ liệu, mô tả chi tiết về tập dữ liệu và các chỉ số được sử dụng để đánh giá hiệu suất.

3.1.1 Môi trường và công cụ

Các thực nghiệm trong đồ án được xây dựng trên nền tảng ngôn ngữ lập trình **Python** (phiên bản 3.10), sử dụng một số thư viện mã nguồn mở phổ biến trong lĩnh vực Khoa học Dữ liệu, Học máy và xử lý dữ liệu thời gian thực. Việc sử dụng các thư viện này nhằm đảm bảo hiệu quả trong xử lý dữ liệu, xây dựng luồng dữ liệu (data pipeline) và phát triển ứng dụng web hiện đại.

3.1.1.1 Các thư viện Python chính

- **FastAPI (phiên bản 0.104.1)**: Nền tảng web hiện đại để xây dựng API với hiệu suất cao, hỗ trợ xử lý bất đồng bộ (asyncio) và tự động tạo tài liệu API.

- **Socket.IO for Python:** Thư viện WebSocket để triển khai giao tiếp thời gian thực giữa máy chủ và máy khách, có cơ chế dự phòng cho phương thức truyền tải.
- **Pandas (phiên bản 2.1.3):** Hỗ trợ xử lý và phân tích dữ liệu dạng bảng, đặc biệt hiệu quả với dữ liệu chuỗi thời gian tài chính.
- **NumPy (phiên bản 1.25.2):** Thư viện tính toán khoa học với cấu trúc mảng hiệu suất cao cho các tác vụ tính toán số.
- **AsyncPG:** Trình điều khiển (driver) PostgreSQL bắt đồng bộ cho Python, tối ưu cho các ứng dụng có độ tương tranh cao.
- **AioKafka:** Trình client Kafka bắt đồng bộ để xử lý luồng dữ liệu với thông lượng cao và độ trễ thấp.
- **yfinance:** Thư viện Python để truy cập dữ liệu tài chính từ API của Yahoo Finance.
- **ccxt:** Thư viện hỗ trợ kết nối với nhiều sàn giao dịch tiền điện tử.
- **TA-Lib:** Thư viện tính toán các chỉ báo kỹ thuật trong phân tích tài chính.

3.1.1.2 Công nghệ phía Giao diện (Frontend)

- **React 18 với TypeScript:** Nền tảng giao diện người dùng hiện đại, đảm bảo an toàn kiểu dữ liệu (type safety), sử dụng công cụ xây dựng Vite.
- **Socket.IO Client:** Thư viện phía máy khách để nhận dữ liệu thời gian thực từ máy chủ, hỗ trợ tự động kết nối lại.
- **Chart.js / Recharts:** Các thư viện trực quan hóa để tạo biểu đồ tài chính tương tác với khả năng cập nhật theo thời gian thực.
- **Tailwind CSS:** Framework CSS theo hướng tiện ích (utility-first) để xây dựng thiết kế đáp ứng (responsive) và định kiểu theo thành phần.

- **Nginx**: Máy chủ web và proxy ngược, dùng để phục vụ các tệp tĩnh và chuyển tiếp yêu cầu API.

3.1.1.3 HẠ TẦNG VÀ VẬN HÀNH (DevOps)

- **Docker và Docker Compose**: Công nghệ container hóa toàn bộ hệ thống với hơn 15 dịch vụ vi mô (microservices) được điều phối.
- **Apache Airflow (phiên bản 2.7.0)**: Nền tảng điều phối luồng công việc, dùng cho xử lý dữ liệu theo lô và thu nạp dữ liệu theo lịch trình.
- **Apache Kafka (phiên bản 3.5)**: Nền tảng truyền dữ liệu phân tán, phục vụ cho việc thu nạp dữ liệu thời gian thực và kiến trúc hướng sự kiện.
- **PostgreSQL (phiên bản 15) với TimescaleDB**: Cơ sở dữ liệu chuỗi thời gian được tối ưu hóa với các bảng đặc biệt (hypertables) và các truy vấn tổng hợp liên tục (continuous aggregates).
- **Redis**: Cơ sở dữ liệu trong bộ nhớ (in-memory) dùng cho quản lý phiên làm việc, bộ nhớ đệm (caching) và lưu trữ dữ liệu tạm thời.
- **Zookeeper**: Dịch vụ điều phối phân tán cho việc quản lý cụm Kafka.

Toàn bộ mã nguồn và tệp cấu hình được quản lý thông qua hệ thống quản lý phiên bản. Hệ thống được triển khai hoàn toàn bằng container để đảm bảo tính nhất quán của môi trường từ giai đoạn phát triển đến sản phẩm và khả năng tái lập cao.

3.1.2 Kiến trúc đường ống dữ liệu và triển khai

3.1.2.1 Giới thiệu quy trình

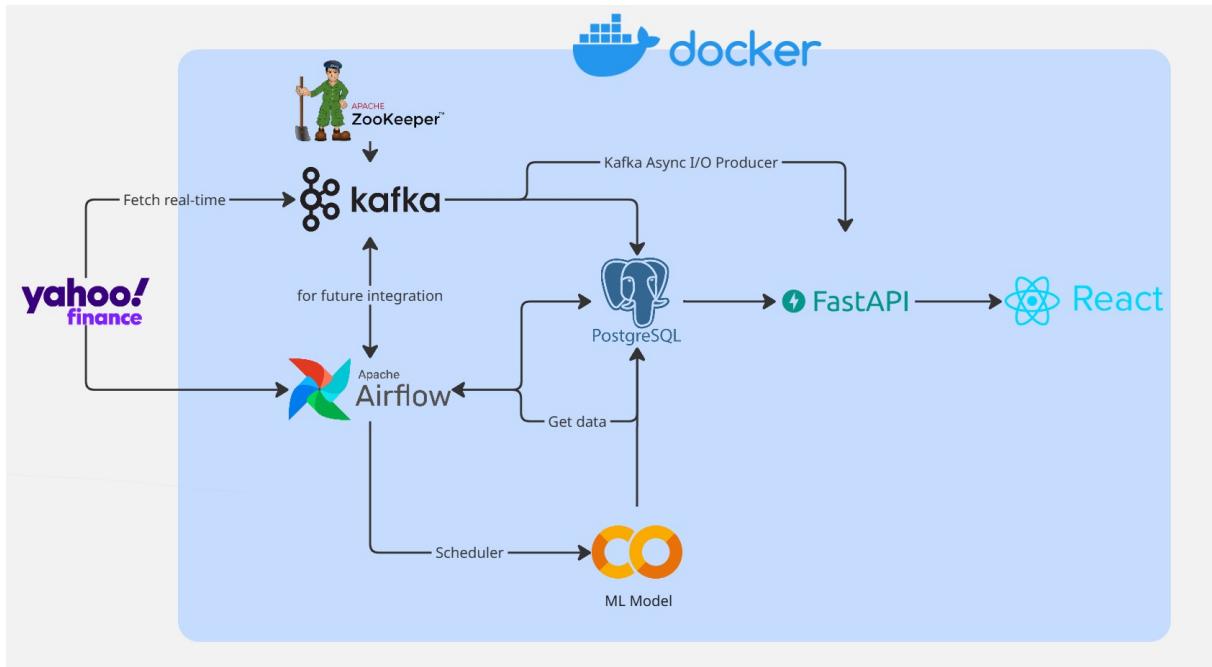
Trong bối cảnh thị trường tài chính với dữ liệu biến động liên tục và yêu cầu xử lý thời gian thực với độ trễ thấp, việc xây dựng một hệ thống đường ống dữ liệu (data pipeline) mạnh mẽ là vô cùng quan trọng. Hệ

thống này phải đảm bảo khả năng thu thập, xử lý và phân phối dữ liệu một cách hiệu quả và tin cậy.

Hệ thống **Đường ống Dữ liệu Chứng khoán Thời gian thực** của dự án được thiết kế theo kiến trúc dịch vụ vi mô (microservices) với chiến lược thu nạp dữ liệu kép, kết hợp xử lý theo lô và truyền dữ liệu thời gian thực để đáp ứng các mục tiêu chính:

- **Chiến lược thu nạp dữ liệu kép:** Kết hợp các luồng công việc (DAGs) của Airflow cho các tác vụ theo lịch trình và một Trình sản xuất Dữ liệu nâng cao (Enhanced Data Producer) để truyền dữ liệu liên tục theo thời gian thực.
- **Lưu trữ dữ liệu đa mức độ chi tiết:** Lưu trữ dữ liệu ở các mức khác nhau (hàng ngày, 5 phút, 1 phút) để phục vụ các nhu cầu phân tích đa dạng.
- **Xử lý thời gian thực:** Xử lý và phân phối dữ liệu thời gian thực với độ trễ thấp thông qua luồng tin nhắn (message streaming) và Socket.IO.
- **Kiến trúc dịch vụ vi mô có khả năng mở rộng:** Thiết kế hơn 15 dịch vụ độc lập, được container hóa để dễ dàng mở rộng theo chiều ngang và bảo trì.
- **Phối hợp và tránh xung đột dữ liệu:** Tránh xung đột dữ liệu giữa quá trình xử lý theo lô và truyền dữ liệu thời gian thực thông qua logic điều phối.
- **Cấu hình nhận biết môi trường:** Tự động phát hiện và thích ứng giữa môi trường Docker và môi trường phát triển cục bộ.

3.1.2.2 Sơ đồ kiến trúc tổng thể



Hình 3.1: Sơ đồ kiến trúc tổng thể của Đường ống Dữ liệu Chứng khoán Thời gian thực với hơn 15 dịch vụ được container hóa

Hệ thống bao gồm hơn 15 dịch vụ được điều phối hoàn toàn bởi Docker Compose với luồng dữ liệu chính như sau:

Yahoo Finance → Trình sản xuất Dữ liệu → Kafka → Trình tiêu thụ API → Socket.IO → Bảng điều khiển React TypeScript

Các mẫu luồng dữ liệu chính

- **Dữ liệu lịch sử theo lô:** Luồng công việc Airflow (DAGs) → PostgreSQL (kích hoạt thủ công để lấy dữ liệu lịch sử, lên lịch để cập nhật).
- **Truyền dữ liệu thời gian thực:** Trình sản xuất Dữ liệu → Chủ đề Kafka `stock_prices` → Trình tiêu thụ API bất đồng bộ → Phát sóng qua Socket.IO.
- **Tích hợp giao diện người dùng:** React TypeScript với khả năng nhận diện môi trường (Docker/Cục bộ) → Máy khách Socket.IO → Cập nhật bảng điều khiển theo thời gian thực.

- **Chiến lược cơ sở dữ liệu:** Sử dụng phương pháp đa bảng của TimescaleDB với cơ chế tự động chọn bảng dựa trên khung thời gian được yêu cầu.

3.1.3 Mô tả tập dữ liệu

Tập dữ liệu được thiết kế theo kiến trúc lưu trữ đa mức độ chi tiết để phục vụ đa dạng các nhu cầu, từ phân tích lịch sử dài hạn đến theo dõi và cảnh báo thời gian thực. Dữ liệu được thu thập từ nhiều nguồn thông qua chiến lược thu nạp kép và được tối ưu hóa cho phân tích chuỗi thời gian.

3.1.3.1 Kiến trúc Lược đồ Cơ sở dữ liệu

Dữ liệu Lịch sử Hàng ngày (finance.stock_daily) Bảng chính cho dữ liệu OHLCV hàng ngày với lịch sử sâu, phục vụ việc huấn luyện mô hình học máy và phân tích kỹ thuật dài hạn:

Bảng 3.1: Cấu trúc bảng stock_daily - Dữ liệu OHLCV Lịch sử Hàng ngày

Cột	Kiểu dữ liệu	Mô tả
symbol	VARCHAR(10)	Mã chứng khoán (khóa chính)
date	DATE	Ngày giao dịch (khóa chính)
open	DECIMAL(10,4)	Giá mở cửa
high	DECIMAL(10,4)	Giá cao nhất trong ngày
low	DECIMAL(10,4)	Giá thấp nhất trong ngày
close	DECIMAL(10,4)	Giá đóng cửa
volume	BIGINT	Khối lượng giao dịch
adjusted_close	DECIMAL(10,4)	Giá đóng cửa đã điều chỉnh (chia tách, cổ tức)
dividend_amount	DECIMAL(8,4)	Lượng cổ tức được trả trong ngày
split_coefficient	DECIMAL(8,4)	Hệ số chia tách cổ phiếu

Dữ liệu trong ngày khung 5 phút (finance.stock_intraday_5m)
Dữ liệu chi tiết cho phân tích trong ngày với chính sách lưu trữ 60 ngày:

Bảng 3.2: Cấu trúc bảng stock_intraday_5m - Dữ liệu Khung 5 phút

Cột	Kiểu dữ liệu	Mô tả
symbol	VARCHAR(10)	Mã chứng khoán
timestamp	TIMESTAMPTZ	Dấu thời gian (múi giờ UTC) của nến 5 phút
open	DECIMAL(10,4)	Giá mở cửa của nến 5 phút
high	DECIMAL(10,4)	Giá cao nhất trong 5 phút
low	DECIMAL(10,4)	Giá thấp nhất trong 5 phút
close	DECIMAL(10,4)	Giá đóng cửa của nến 5 phút
volume	BIGINT	Tổng khối lượng trong 5 phút

Dữ liệu thời gian thực khung 1 phút (finance.stock_realtime_1m)

Bảng thời gian thực để theo dõi trực tiếp với thông tin trạng thái thị trường:

Bảng 3.3: Cấu trúc bảng stock_realtime_1m - Báo giá Thời gian thực

Cột	Kiểu dữ liệu	Mô tả
symbol	VARCHAR(10)	Mã chứng khoán
timestamp	TIMESTAMPTZ	Dấu thời gian cập nhật (UTC)
price	DECIMAL(10,4)	Giá giao dịch gần nhất
change_amount	DECIMAL(10,4)	Mức thay đổi tuyệt đối so với giá đóng cửa trước
change_percent	DECIMAL(5,2)	Phần trăm thay đổi
volume	BIGINT	Khối lượng giao dịch lũy kế
bid	DECIMAL(10,4)	Giá chào mua cao nhất
ask	DECIMAL(10,4)	Giá chào bán thấp nhất
market_state	VARCHAR(20)	Trạng thái thị trường (Mở cửa, Đóng cửa, Trước giờ, Sau giờ)
source	VARCHAR(20)	Nguồn dữ liệu (yahoo, binance, v.v.)

Siêu dữ liệu Chứng khoán (finance.stock_metadata)

Dữ liệu chủ về thông tin công ty và phân tích cơ bản:

Bảng 3.4: Cấu trúc bảng stock_metadata - Thông tin Công ty và Thị trường

Cột	Kiểu dữ liệu	Mô tả
symbol	VARCHAR(10)	Mã chứng khoán (Khóa chính)
company_name	VARCHAR(255)	Tên đầy đủ của công ty
sector	VARCHAR(100)	Ngành kinh tế (Công nghệ, Y tế, v.v.)
industry	VARCHAR(100)	Lĩnh vực cụ thể (Phần mềm, Công nghệ sinh học, v.v.)
market_cap	BIGINT	Vốn hóa thị trường (USD)
pe_ratio	DECIMAL(8,2)	Tỷ số P/E
beta	DECIMAL(6,3)	Hệ số beta (đo lường biến động so với thị trường)
dividend_yield	DECIMAL(5,2)	Tỷ suất cổ tức (%)
is_active	BOOLEAN	Trạng thái giao dịch hiện tại
last_updated	TIMESTAMPTZ	Thời gian cập nhật siêu dữ liệu lần cuối

3.1.3.2 Mô hình sử dụng dữ liệu và Tích hợp Backend

Luồng logic lựa chọn bảng ở phía Backend Hệ thống tự động chọn bảng phù hợp dựa trên tham số yêu cầu API và khung thời gian:

- Truy vấn giá hiện tại:
 - Điểm cuối: /api/stocks/current/{symbol}
 - Nguồn: stock_realtimel_1m để lấy báo giá mới nhất.
 - Mục đích: Hiển thị giá thời gian thực, cảnh báo, theo dõi thị trường.
- Dữ liệu biểu đồ lịch sử:
 - Điểm cuối: /api/stocks/historical/{symbol}
 - Lựa chọn nguồn: stock_daily cho khung thời gian > 6 tháng, stock_intraday_5m cho phân tích ngắn hạn chi tiết.
 - Tham số: Ngày bắt đầu, ngày kết thúc, mức độ chi tiết.
- Cập nhật truyền trực tiếp thời gian thực:

- Luồng: Chủ đề Kafka `stock_prices` → Trình tiêu thụ FastAPI bất đồng bộ → Phát sóng qua Socket.IO.
- Bảng đích: `stock_realtime_1m` để lưu trữ.
- Sự kiện: `all_stocks`, `crypto_data`, `real_time_update`.
- Phân tích thị trường và Bảng điều khiển:
 - Điểm cuối: `/api/dashboard/overview`
 - Kết hợp đa bảng: Dữ liệu giá với `stock_metadata` để làm giàu thông tin phân tích.
 - Sử dụng `continuous aggregates` của TimescaleDB để tối ưu hiệu suất.

Lược đồ sự kiện thời gian thực của Socket.IO

- Sự kiện: `all_stocks`
 - Tải trọng (Payload): Mảng các đối tượng chứng khoán với giá hiện tại.
 - Tần suất: Bất cứ khi nào có cập nhật từ trình tiêu thụ Kafka.
 - Định dạng: `{symbol, price, change_percent, volume, timestamp}`
- Sự kiện: `crypto_data`
 - Tải trọng: Giá tiền điện tử với độ chi tiết 1 giây.
 - Nguồn: Tích hợp CCXT trong một luồng riêng.
 - Định dạng: `{symbol, price, change_24h, volume_24h, timestamp}`
- Sự kiện: `real_time_update`
 - Tải trọng: Cập nhật cho từng mã riêng lẻ với đầy đủ dữ liệu thị trường.
 - Kích hoạt: Khi có thay đổi ở một mã cụ thể hoặc theo đăng ký của máy khách.
 - Định dạng: Cấu trúc đầy đủ của bản ghi `stock_realtime_1m`.

3.1.3.3 Đảm bảo và xác thực chất lượng dữ liệu

Các biện pháp toàn vẹn dữ liệu

- Chuẩn hóa dấu thời gian:

- Tất cả dấu thời gian được lưu theo múi giờ UTC trong PostgreSQL (kiểu TIMESTAMPTZ).
- Việc chuyển đổi múi giờ được xử lý ở tầng ứng dụng.

- Xác thực và chuẩn hóa mã chứng khoán:

- Chuẩn hóa sang chữ hoa cho các mã.
- Xác thực dựa trên danh sách các sàn và thị trường đã biết.
- Ngăn chặn trùng lặp bằng khóa chính tổng hợp.

- Xác thực dữ liệu giá:

- Kiểm tra phạm vi hợp lệ cho các giá trị giá.
- Phát hiện và lọc các giá trị ngoại lai (biến động bất thường).
- Xác thực logic OHLC (Mở ≤ Cao, Thấp ≤ Đóng, v.v.).

- Toàn vẹn dữ liệu khối lượng:

- Xác thực khối lượng không âm.
- Phát hiện và cảnh báo các đột biến khối lượng đáng ngờ.

Cơ chế xử lý lỗi và phục hồi

- Phục hồi lỗi API:

- Logic thử lại với thời gian chờ tăng dần cho các lệnh gọi API Yahoo Finance.
- Sử dụng nguồn dữ liệu dự phòng khi API chính không khả dụng.
- Sử dụng dữ liệu từ bộ nhớ đệm khi các nguồn thời gian thực gặp lỗi.

- Độ bền của đường ống dữ liệu:

- Khả năng phát lại tin nhắn Kafka để phục hồi dữ liệu.
- Hoàn tác giao dịch cơ sở dữ liệu khi các tác vụ xử lý lô thất bại.
- Cơ chế thử lại tác vụ của Airflow với chính sách có thể cấu hình.

- Phục hồi luồng thời gian thực:

- Socket.IO tự động kết nối lại với bộ đệm phía máy khách.
- Quản lý vị trí (offset) của trình tiêu thụ Kafka để đảm bảo tin nhắn được xử lý.
- Nhóm kết nối cơ sở dữ liệu (connection pooling) với khả năng phục hồi tự động.

- Giám sát và kiểm tra tình trạng:

- Điểm cuối /health của FastAPI để giám sát trạng thái dịch vụ.
- Cảnh báo thành công/thất bại của luồng công việc Airflow.
- Các chỉ số tùy chỉnh về độ tươi mới của dữ liệu và hiệu suất đường ống.

Các chiến lược tối ưu hóa hiệu suất

- Tối ưu hóa TimescaleDB:

- Phân vùng hypertable theo khoảng thời gian để tăng hiệu suất truy vấn.
- Sử dụng continuous aggregates cho các phân tích phổ biến đã được tính toán trước.
- Chiến lược đánh chỉ mục (indexing) cho các truy vấn dựa trên mã và thời gian.
- Chính sách lưu giữ dữ liệu để tự động dọn dẹp dữ liệu cũ.

- Tầng bộ nhớ đệm (Caching):

- Sử dụng Redis để lưu vào bộ nhớ đệm các siêu dữ liệu thường xuyên truy cập.
 - Bộ nhớ đệm ở cấp ứng dụng cho dữ liệu tham chiếu tĩnh.
 - Chiến lược bộ nhớ đệm của trình duyệt cho dữ liệu biểu đồ lịch sử.
- **Tối ưu hóa xử lý bất đồng bộ:**
 - Nhóm kết nối AsyncPG cho các hoạt động cơ sở dữ liệu.
 - Xử lý tin nhắn theo lô với Aiokafka để đạt thông lượng cao.
 - Các trình xử lý tuyến đường (route handler) bất đồng bộ của FastAPI để phản hồi API không bị chặn.

Tổng thể, kiến trúc cơ sở dữ liệu đa bảng được thiết kế để hỗ trợ các trường hợp sử dụng đa dạng, từ phân tích giao dịch thời gian thực đến nghiên cứu lịch sử, với khả năng mở rộng theo chiều ngang và duy trì chất lượng

3.2 Mô hình Attention và học đa nhiệm

Hướng tiếp cận này tập trung vào việc cải thiện khả năng dự báo bằng cách kết hợp nhiều nhiệm vụ liên quan và sử dụng một kiến trúc chú ý hai giai đoạn để tự động chọn lọc thông tin quan trọng từ dữ liệu đầu vào.

3.2.1 Cấu hình Mô hình Cơ sở: Stacked LSTM

Để đánh giá hiệu quả của mô hình đề xuất, chúng tôi đã triển khai một mô hình cơ sở là Stacked LSTM, với nền tảng lý thuyết đã được trình bày ở mục 2.5.2. Mô hình này được xây dựng bao gồm hai lớp LSTM xếp chồng, sau đó là các lớp Dropout để điều chỉnh và các lớp Dense để đưa ra dự báo cuối cùng. Cấu hình cụ thể cho mô hình cơ sở trong thực nghiệm này được tóm tắt trong Bảng 3.5.

Bảng 3.5: Các tham số chính của mô hình Stacked LSTM cơ sở.

Tham số	Giá trị
Sequence Length	30
Số lớp LSTM	2
Số units mỗi lớp LSTM	32
Tỷ lệ Dropout	0.2
Số units lớp Dense	25
Epochs	50
Batch Size	32
Optimizer	Adam
Hàm Loss	Mean Squared Error (MSE)

Mô hình cơ sở được huấn luyện trên dữ liệu đầu vào là chuỗi đa biến gồm 5 đặc trưng OHLCV. Hiệu suất của mô hình được đánh giá trên tập kiểm thử (20% dữ liệu) bằng các chỉ số hồi quy phổ biến là RMSE, MAE và R^2 .

3.2.2 Mô hình đề xuất: MT-DA-RNN

Mô hình đề xuất MT-DA-RNN được thiết kế nhằm giải quyết các hạn chế của baseline, với các cải tiến cốt lõi về cả phương pháp chuẩn bị dữ liệu và kiến trúc mô hình.

Chuẩn bị dữ liệu và Kỹ thuật Đặc trưng (Preprocessing and Feature Engineering). Đây là cải tiến quan trọng đầu tiên và là một phần không thể tách rời của phương pháp đề xuất. Để cung cấp cho MT-DA-RNN một cái nhìn đa chiều về thị trường, một quy trình chuẩn bị dữ liệu chuyên biệt đã được thực hiện.

1. Làm sạch dữ liệu (Data Cleaning) Dữ liệu tài chính thô sau khi tính toán các chỉ báo kỹ thuật có thể chứa các giá trị không hợp lệ. Quy trình làm sạch được thực hiện tuần tự để đảm bảo tính toàn vẹn của dữ liệu:

- **Xử lý giá trị vô cực:** Các giá trị vô cực (\inf , $-\inf$), thường xuất hiện do các phép chia cho 0 trong quá trình tính toán, được thay thế bằng giá trị không xác định (NaN – Not a Number).

- **Loại bỏ giá trị thiếu:** Sau khi xử lý giá trị vô cực, các hàng dữ liệu chứa bất kỳ giá trị NaN nào sẽ bị loại bỏ hoàn toàn. Các giá trị NaN này chủ yếu phát sinh ở các mẫu dữ liệu đầu tiên của chuỗi do các chỉ báo kỹ thuật cần một "cửa sổ trượt" (rolling window) dữ liệu quá khứ để có thể tính toán (ví dụ: SMA(10) sẽ không có giá trị cho 9 ngày đầu tiên). Việc loại bỏ các hàng này đảm bảo mọi mẫu dữ liệu đưa vào mô hình đều đầy đủ và hợp lệ.

2. Làm giàu dữ liệu với các chỉ báo kỹ thuật (Feature Engineering with Technical Indicators) Để cung cấp cho mô hình một cái nhìn đa chiều vượt ra ngoài dữ liệu Giá Mở cửa, Cao nhất, Thấp nhất, Đóng cửa và Khối lượng (OHLCV) gốc, một bộ các Chỉ báo kỹ thuật (Technical Indicators - TIs) đã được tính toán. Các chỉ báo này được lựa chọn cẩn thận để nắm bắt các khía cạnh khác nhau của động lực thị trường như xu hướng, động lượng và mức độ biến động. Việc làm giàu dữ liệu này giúp mô hình có thêm ngữ cảnh để đưa ra các dự báo chính xác hơn. Bảng 3.6 tóm tắt các chỉ báo được sử dụng và vai trò của chúng.

3. Xây dựng bộ đặc trưng cuối cùng và chuẩn hóa Dữ liệu chuỗi thời gian tài chính có tính chất không dừng (non-stationarity), nghĩa là các thuộc tính thống kê của chúng thay đổi theo thời gian. Để giải quyết vấn đề này và đưa các đặc trưng về cùng một thang đo, một bộ đặc trưng cuối cùng gồm **14 features** được xây dựng và chuẩn hóa:

- **Biến đổi các đặc trưng có tính xu hướng:** Các đặc trưng có giá trị biến động cùng với xu hướng chung của thị trường được chuyển đổi sang dạng **phần trăm thay đổi (%) change**. Việc này giúp ổn định chuỗi dữ liệu, loại bỏ ảnh hưởng của xu hướng và giúp mô hình tập trung vào động lượng thay đổi. Nhóm này bao gồm dữ liệu OHLCV, Volume, các đường SMA, EMA và các dải Bollinger. Công thức tính được định nghĩa như sau:

$$\text{PctChange}_t = \frac{\text{value}_t - \text{value}_{t-1}}{\text{value}_{t-1}} \quad (3.1)$$

Bảng 3.6: Các chỉ báo kỹ thuật được sử dụng trong mô hình đề xuất.

Chỉ báo (Tham số)	Mô tả và Vai trò
SMA (10) & EMA (10)	Trung bình trượt đơn giản và hàm mũ với chu kỳ 10 ngày, dùng để làm mượt giá và xác định xu hướng ngắn hạn.
RSI (14)	Chỉ báo sức mạnh tương đối (Relative Strength Index) , một chỉ báo dao động đo lường tốc độ thay đổi của giá để xác định các vùng quá mua/quá bán.
MACD	Chỉ báo trung bình động hội tụ phân kỳ (Moving Average Convergence Divergence) , đo lường động lượng và mối quan hệ giữa hai đường EMA của giá.
Bollinger Bands (20)	Các Dải Bollinger đo lường sự biến động của thị trường, bao gồm một đường SMA(20) và các dải trên/dưới dựa trên độ lệch chuẩn.
ATR (14)	Phạm vi thực trung bình (Average True Range) , một chỉ báo đo lường mức độ biến động (volatility) của thị trường.
ADX (14)	Chỉ số định hướng trung bình (Average Directional Index) , đo lường sức mạnh của một xu hướng hiện tại, không phân biệt xu hướng tăng hay giảm.
Stochastic Oscillator (14, 3, 3)	Một chỉ báo dao động động lượng, so sánh giá đóng cửa với một khoảng giá trong một giai đoạn nhất định để xác định vị thế trong xu hướng.

- **Giữ nguyên các đặc trưng dao động:** Các chỉ báo dao động (oscillators) như RSI, ADX, Stochastic Oscillator và các chỉ báo đo lường biến động như MACD, ATR đã được thiết kế để dao động quanh một mức trung tâm hoặc đo lường sự thay đổi tương đối. Do đó, chúng được giữ nguyên giá trị gốc để bảo toàn ý nghĩa về các ngưỡng hoặc mức độ dao động.
- **Chuẩn hóa bằng StandardScaler:** Sau khi tạo ra bộ 14 đặc trưng cuối cùng, toàn bộ dữ liệu được đưa qua một bộ chuẩn hóa StandardScaler. Kỹ thuật này biến đổi dữ liệu sao cho chúng có

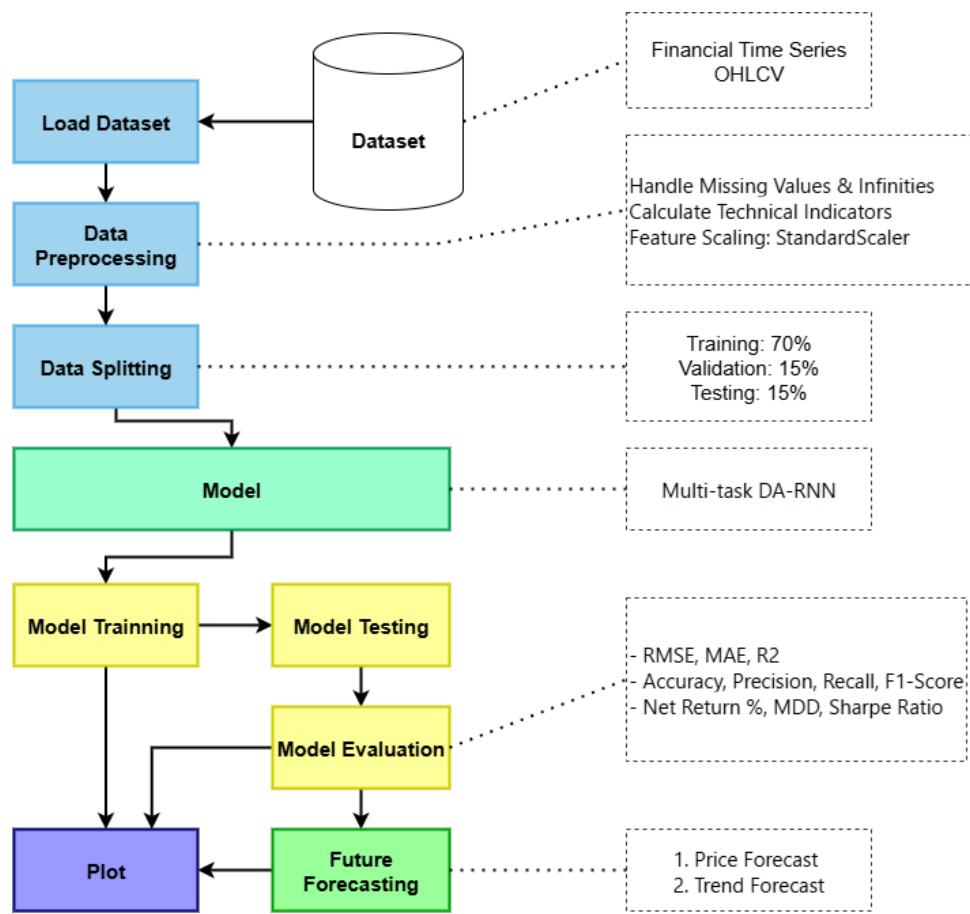
giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1. Việc chuẩn hóa là một bước bắt buộc đối với các mô hình học sâu, giúp quá trình tối ưu hóa bằng thuật toán lan truyền ngược và gradient descent hội tụ nhanh và ổn định hơn. Để tránh **rò rỉ dữ liệu (data leakage)**, bộ `StandardScaler` được "khớp" (*fit*) duy nhất trên tập dữ liệu huấn luyện và sau đó được dùng để biến đổi (*transform*) cho cả tập validation và tập test.

Nền tảng và các cải tiến chính Mô hình MT-DA-RNN được xây dựng nhằm giải quyết các hạn chế của mô hình Stacked LSTM cơ sở bằng cách kết hợp ba chiến lược chính:

1. **Làm giàu dữ liệu đầu vào (Data Enrichment):** Khác với mô hình cơ sở chỉ sử dụng dữ liệu OHLCV, mô hình đề xuất được cung cấp một bộ đặc trưng đa biến mở rộng, bao gồm 8 nhóm chỉ báo kỹ thuật. Điều này cung cấp cho mô hình một cái nhìn sâu sắc và đa chiều hơn về động lực thị trường.
2. **Kế thừa kiến trúc Attention tiên tiến (Leveraging Advanced Architecture):** Để xử lý hiệu quả bộ dữ liệu phức tạp này, mô hình sử dụng kiến trúc DA-RNN (Dual-Stage Attention-based RNN) làm nền tảng (backbone). Cơ chế attention hai giai đoạn vốn có của DA-RNN cho phép mô hình tự động tập trung vào các đặc trưng và các thời điểm quan trọng nhất, vượt qua hạn chế của kiến trúc LSTM thông thường.
3. **Tích hợp học đa nhiệm:** Điểm mới và là đóng góp chính của mô hình đề xuất là việc mở rộng kiến trúc DA-RNN thành một mô hình học đa nhiệm. Thay vì chỉ thực hiện một nhiệm vụ dự báo giá, MT-DA-RNN được huấn luyện để đồng thời thực hiện hai nhiệm vụ có liên quan:
 - Dự báo giá trị giá đóng cửa (Hồi quy).
 - Dự báo xu hướng "Up/Down/Neutral" (Phân loại).

Để xác định xu hướng, thay vì dùng một ngưỡng cố định (ví dụ: tăng/giảm 5%), chúng tôi chia các biến động giá trên tập huấn luyện thành ba nhóm có số lượng gần bằng nhau: 33% các ngày giảm mạnh nhất được gọi là "Down", 33% các ngày tăng mạnh nhất là "Up", và khoảng 34% còn lại ở giữa được xem là "Neutral". Cách chia này đảm bảo mô hình học được một cách công bằng từ cả ba loại xu hướng cho từng mã cổ phiếu, thay vì bị thiên vị bởi một loại xu hướng nào đó chiếm đa số. Cải tiến học đa nhiệm này giúp mô hình học được các biểu diễn chung mạnh mẽ hơn và tăng khả năng khái quát hóa.

Quy trình thực nghiệm tổng thể cho mô hình đề xuất được mô tả trong Hình 3.2.



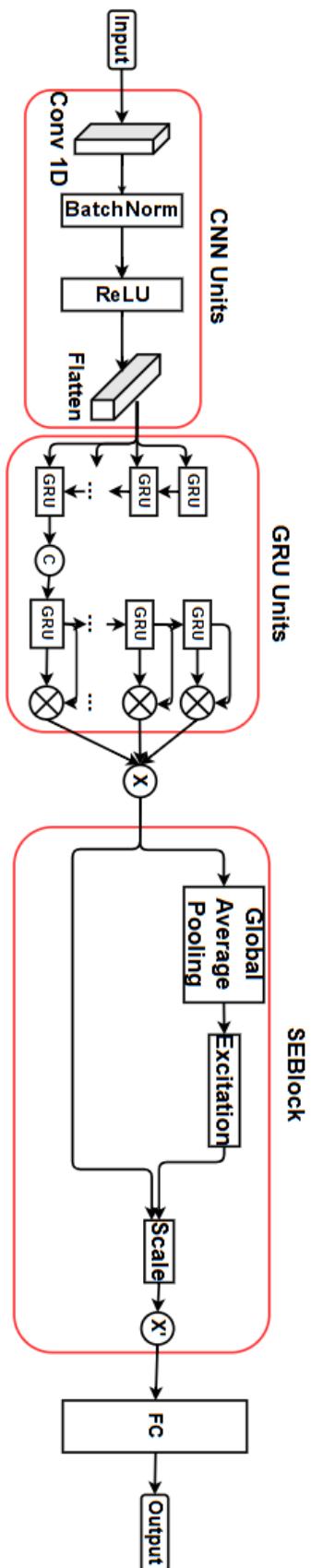
Hình 3.2: Sơ đồ tổng quan quy trình thực nghiệm cho mô hình đề xuất MT-DA-RNN.

3.3 Mô hình kết hợp CNN – GRU và khối SEBlock (CGSE)

Hướng tiếp cận thứ hai nhằm mục tiêu khai thác cả đặc trưng cục bộ và động lực thời gian trong chuỗi dữ liệu. Để thực hiện điều này, nhóm sử dụng sự kết hợp giữa Mạng nơ-ron Tích chập (CNN) và Mạng hồi tiếp GRU (Gated Recurrent Unit). Mô hình baseline sử dụng kiến trúc CNN-GRU, sau đó được cải tiến thành mô hình CGSE bằng cách tích hợp khối Squeeze-and-Excitation nhằm tối ưu hiệu suất trích xuất đặc trưng.

3.3.1 Kiến trúc mô hình CGSE

Mô hình CGSE được đề xuất nhằm nâng cao hiệu quả dự báo chuỗi thời gian thông qua việc kết hợp ba thành phần chính: mạng tích chập (CNN), mạng hồi tiếp GRU và khối chú ý kênh Squeeze-and-Excitation (SEBlock). Ý tưởng cốt lõi của mô hình là tận dụng khả năng học đặc trưng cục bộ theo thời gian của CNN, khả năng ghi nhớ chuỗi dài hạn của GRU, đồng thời tăng cường khả năng làm nổi bật các đặc trưng quan trọng thông qua SEBlock – một kỹ thuật chú ý giúp mô hình tự động học được trọng số quan trọng giữa các đặc trưng đầu vào.[5][6][8][9]



Hình 3.3: Kiến trúc mô hình CGSE

Cụ thể, đầu vào của mô hình là một chuỗi thời gian nhiều chiều, gồm các chỉ số giá mỏ của, giá đóng cửa, giá cao nhất, giá thấp nhất, ... Dữ liệu đầu vào có dạng:

$$X \in \mathbb{R}^{T \times d}$$

Trong đó:

- T là bước thời gian(Seq_len)
- d là số đặc trưng tại mỗi bước.

Trước tiên, Dữ liệu này được chuyển vị để phù hợp với CNN:

$$X_{CNN} = \text{permute}(X) \in \mathbb{R}^{d \times T} \quad (3.2)$$

Sau đó đưa vào các lớp CNN 1D, cho phép trích xuất các mẫu (pattern) cục bộ có tính thời gian từ mỗi đặc trưng riêng biệt.

$$F = \text{ReLU}(\text{Conv1D}(X_{CNN})) \in \mathbb{E}^{c \times T} \quad (3.3)$$

với c là số lượng kênh đầu ra Khác với các mô hình truyền thống xử lý từng đặc trưng độc lập hoặc đơn giản kết hợp tất cả, CNN giúp mô hình tự động học được các khuôn mẫu quan trọng thông qua việc sử dụng nhiều kernel khác nhau.

Sau khi trích xuất đặc trưng từ CNN, đầu ra F sẽ được đưa vào GRU. GRU có khả năng ghi nhớ thông tin trong chuỗi mà vẫn giảm thiểu số lượng tham số. GRU giúp mô hình học được mối quan hệ thời gian dài hạn giữa các bước trong chuỗi, từ đó cải thiện độ chính xác trong dự báo. Dữ liệu đầu vào từ CNN sẽ được chuyển đổi kích thước về dạng chuẩn:

$$F_{GRU} \in \mathbb{R}^{T \times c} \quad (3.4)$$

Lúc này, chuỗi đặc trưng theo thời gian được đưa vào GRU để học thuộc dài hạn theo thời gian:

$$h_t = \text{GRU}(F_{GRU,t}, h_{t-1}) \quad (3.5)$$

Sau đó, chọn đầu ra tại thời điểm cuối - tức là thời điểm gần thời gian cần dự đoán nhất, do đó ta lấy vector ẩn cuối cùng của chuỗi:

$$h_{final} = h_T \in \mathbb{R}^H \quad (3.6)$$

Vector đặc trưng h_{final} chứa thông tin tổng hợp của toàn bộ chuỗi. Tuy nhiên, không phải chiều nào của vector này cũng quan trọng như nhau đối với dự báo. Do đó, SEBlock được áp dụng để tái chuẩn hóa trọng số giữa các chiều đầu ra. SEBlock hoạt động gồm hai bước chính: "Squeeze" và "Excitation".

Ở bước Squeeze, thông tin theo chiều thời gian được nén lại bằng cách tính trung bình toàn cục, giúp tổng hợp thông tin khái quát về mỗi kênh đặc trưng.

$$z = h_{final} \in \mathbb{R}^H \quad (3.7)$$

Sau đó, bước Excitation sử dụng một mạng con gồm hai lớp dense (một giảm chiều và một tăng chiều) để sinh ra trọng số điều chỉnh tương ứng cho mỗi đặc trưng.

$$s = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot z)) \in \mathbb{R}^H \quad (3.8)$$

Trong đó:

- $W_1 \in \mathbb{R}^{\frac{H}{r} \times H}, W_2 \in \mathbb{R}^{H \times \frac{H}{r}}$: trọng số hai lớp fully-connected.
- r là hệ số giảm chiều
- σ : hàm sigmoid

Trọng số này được nhân lại vào tensor đặc trưng đầu ra, từ đó làm nổi bật những đặc trưng quan trọng và giảm tác động của các đặc trưng kém liên quan.

$$\mathbf{h}_{SE} = \mathbf{h}_{final} \odot \mathbf{s} \quad (3.9)$$

Sau khi làm nổi bật các đặc trưng quan trọng nhờ SEBlock, vector đầu ra được đưa qua một lớp tuyến tính để dự đoán giá trị mục tiêu:

$$\hat{y} = W \cdot \mathbf{h}_{SE} + b \quad (3.10)$$

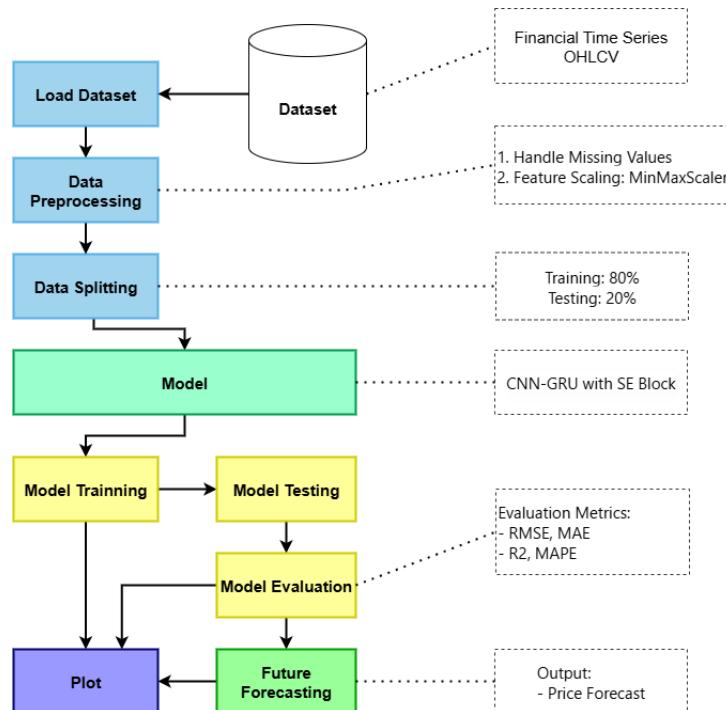
trong đó $\hat{y} \in \mathbb{R}^{d_{out}}$ là đầu ra cuối cùng của mô hình.

Điều này đặc biệt có ích trong các bài toán mà không phải đặc trưng nào cũng có ảnh hưởng như nhau đến đầu ra dự đoán.

Tổng thể, kiến trúc CGSE vừa tận dụng được sức mạnh học đặc trưng của CNN, khả năng xử lý chuỗi của GRU, vừa bổ sung tính năng lọc chú ý đặc trưng nhờ SEBlock. Điều này giúp mô hình có thể học được mối quan hệ phức tạp giữa các yếu tố đầu vào theo cả không gian và thời gian, đồng thời tránh việc đối xử các đặc trưng một cách đồng đều như trong các kiến trúc truyền thống.

3.3.2 Quy trình huấn luyện mô hình

Quy trình huấn luyện mô hình dự báo giá cổ phiếu được xây dựng một cách hệ thống nhằm đảm bảo chất lượng đầu ra cũng như khả năng tổng quát hóa của mô hình. Các bước chính bao gồm: thu thập dữ liệu, tiền xử lý dữ liệu, chia tập dữ liệu, xây dựng mô hình, huấn luyện, đánh giá và dự báo. Sơ đồ tổng quát quy trình được minh họa ở hình 3.4 .



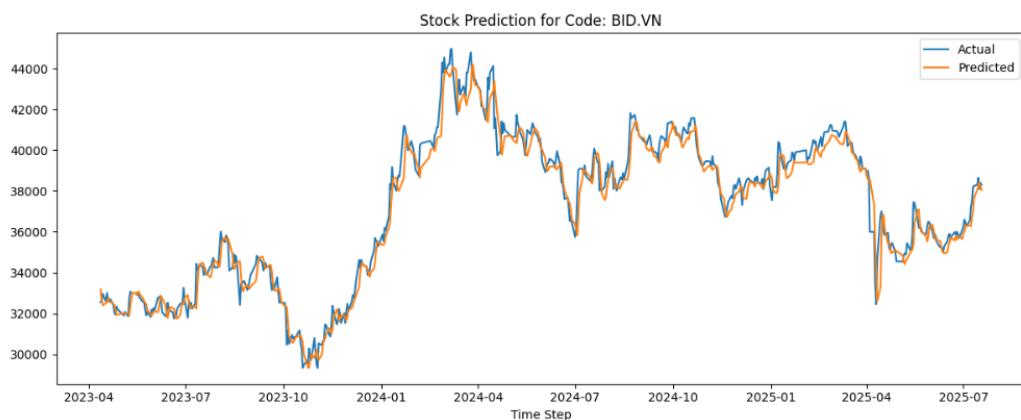
Hình 3.4: Quy trình huấn luyện mô hình CNN-GRU-SEBlock

1. **Thu thập dữ liệu:** Dữ liệu đầu vào của mô hình là chuỗi thời gian tài chính, thường được thể hiện dưới dạng các chỉ số OHLCV, bao gồm: giá mở cửa (Open), giá cao nhất (High), giá thấp nhất (Low), giá đóng cửa (Close) và khối lượng giao dịch (Volume). Đây là những đặc trưng quan trọng phản ánh biến động của thị trường trong một khoảng thời gian cụ thể.
2. **Tiền xử lý dữ liệu:** Dữ liệu thực tế thường chứa các giá trị thiếu, không đồng nhất hoặc có thang đo khác nhau giữa các thuộc tính. Do đó, cần thực hiện xử lý giá trị thiếu (bằng phương pháp nội suy hoặc loại bỏ), và chuẩn hóa dữ liệu về một khoảng xác định bằng kỹ thuật MinMax Scaling. Bước này giúp cải thiện hiệu suất huấn luyện và giảm thiểu sự thiên lệch giữa các đặc trưng đầu vào.
3. **Chia tập dữ liệu:** Dữ liệu sau khi được tiền xử lý sẽ được chia thành hai phần: 80% cho tập huấn luyện và 20% cho tập kiểm tra. Tỷ lệ này được lựa chọn nhằm đảm bảo mô hình có đủ dữ liệu để học trong khi vẫn giữ lại một phần đủ lớn để đánh giá khả năng tổng quát hóa.
4. **Xây dựng mô hình:** Mô hình đề xuất trong báo cáo là CNN-GRU kết hợp với khối Squeeze-and-Excitation (SE Block) theo các bước ở mục 3.3.1.
5. **Huấn luyện và đánh giá mô hình:** Mô hình được huấn luyện trên tập dữ liệu train bằng thuật toán tối ưu hóa (như Adam), với hàm mất mát phản ánh sai lệch giữa dự đoán và thực tế (ví dụ: RMSE hoặc MAE). Sau khi hoàn tất huấn luyện, mô hình được kiểm tra trên tập test nhằm đánh giá hiệu quả và khả năng dự đoán. Mô hình được đánh giá bằng nhiều chỉ số khác nhau nhằm phản ánh toàn diện độ chính xác:

	Code	MAE	RMSE	MAPE (%)	R ²
0	^GSPC	0.015577	0.019347	2.110742	0.978671
1	^DJI	0.010869	0.013636	1.482314	0.983190
2	NQ=F	0.016467	0.020682	2.382447	0.981983
3	^RUT	0.016286	0.020599	2.258968	0.952260
4	^NYA	0.012049	0.014925	1.706425	0.983427
5	VCB.VN	0.013832	0.017788	1.726737	0.979295
6	BID.VN	0.012363	0.017044	1.661662	0.978319
7	FPT.VN	0.032114	0.052016	4.520358	0.932837
8	CTG.VN	0.015778	0.020928	2.543407	0.981008
9	VHM.VN	0.029965	0.042569	10.412688	0.594376
10	HPG.VN	0.011495	0.015808	2.124662	0.981664

Hình 3.5: Minh họa kết quả đánh giá mô hình CGSE

6. **Dự đoán và trực quan:** Sau khi mô hình được đánh giá và xác nhận độ tin cậy, nó sẽ được sử dụng để dự báo giá cổ phiếu trong các bước thời gian tiếp theo. Kết quả dự báo giúp đưa ra những nhận định chiến lược cho các quyết định đầu tư tài chính. Cuối cùng, các kết quả đầu ra được trực quan hóa dưới dạng biểu đồ đường để so sánh giữa giá trị thực tế và giá trị dự đoán. Việc này không chỉ giúp đánh giá trực quan hiệu quả mô hình mà còn hỗ trợ người dùng cuối trong việc hiểu và áp dụng kết quả dự báo.



Hình 3.6: Minh họa đồ thị dự đoán mô hình CGSE

Chương 4

Phân tích kết quả và trực quan trên Dashboard

4.1 Kết quả đánh giá - Mô hình Attention và học đa nhiệm

Trong phần này, hiệu suất của mô hình cơ sở Stacked LSTM và mô hình đề xuất MT-DA-RNN được đánh giá một cách định lượng trên tập dữ liệu kiểm thử. Việc đánh giá được chia thành hai phần, tương ứng với hai nhiệm vụ chính: dự báo giá (hồi quy) và dự báo xu hướng (phân loại).

4.1.1 So sánh hiệu suất dự báo giá

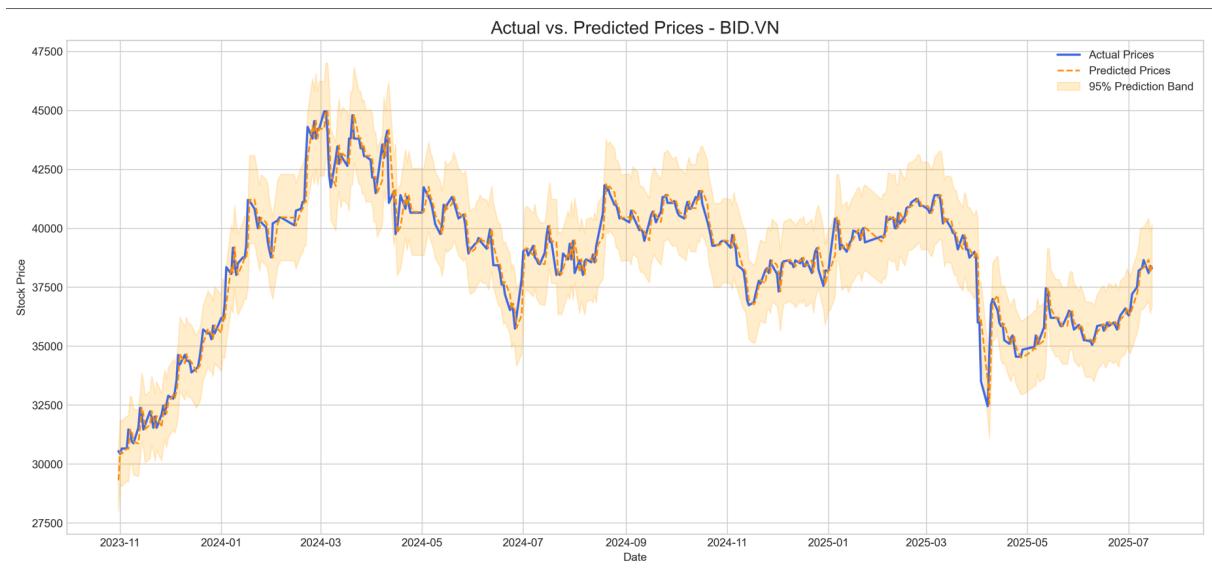
Hiệu suất dự báo giá trị của giá đóng cửa được so sánh giữa hai mô hình thông qua ba chỉ số hồi quy phổ biến: RMSE, MAE và R2. Để có một cái nhìn tổng quan, Bảng 4.1 và 4.2 trình bày kết quả trên 20 mã tài sản khác nhau, trong khi Hình 4.1 minh họa trực quan sự khác biệt cho một mã cụ thể là BID.VN.

Bảng 4.1: Kết quả dự báo giá của mô hình cơ sở baseline Stacked LSTM trên 20 mã tài sản.

STT	Ticker	RMSE	MAE	R2
1	BID.VN	1133.6	966.976	0.9031
2	BNB-USD	20.0035	14.5697	0.9615
3	BTC-USD	3668.75	2626.83	0.9815
4	CTG.VN	1169.89	919.58	0.9641
5	ETH-USD	133.504	99.661	0.9491
6	FPT.VN	11329.7	6999.7	0.8703
7	GAS.VN	1260.75	854.749	0.9259
8	HPG.VN	742.41	550.441	0.959
9	NQ=F	1068.31	705.529	0.8964
10	SOL-USD	11.1357	8.5929	0.8975
11	TCB.VN	815.256	612.805	0.9532
12	USDT-USD	0.0006	0.0004	-0.3048
13	VCB.VN	1979.19	1672.33	0.9255
14	VHM.VN	7214.02	5914.46	0.7175
15	VIC.VN	6864.62	5543.77	0.8903
16	VPB.VN	527.908	371.346	0.8182
17	^DJI	1941.08	1368	0.8845
18	^GSPC	1683.27	1074.8	-0.4781
19	^NYA	2692.95	1960.18	0.1474
20	^RUT	124.699	90.6915	0.8215

Bảng 4.2: Kết quả dự báo giá của mô hình đề xuất MT-DA-RNN trên 20 mã tài sản.

STT	Ticker	RMSE	MAE	R2
1	BID.VN	575.451	399.608	0.9608
2	BNB-USD	16.1324	11.5331	0.9214
3	BTC-USD	1966.54	1400.45	0.9911
4	CTG.VN	523.865	346.981	0.9907
5	ETH-USD	99.2805	71.51	0.9727
6	FPT.VN	1656.17	1099.82	0.9973
7	GAS.VN	958.375	630.804	0.9566
8	HPG.VN	349.453	251.574	0.984
9	NQ=F	243.101	175.22	0.9949
10	SOL-USD	8.0162	5.6786	0.9536
11	TCB.VN	418.041	278.126	0.9864
12	USDT-USD	0.0004	0.0003	0.4715
13	VCB.VN	735.034	492.111	0.9569
14	VHM.VN	2170.26	1634.01	0.9546
15	VIC.VN	2921.93	2125.78	0.9603
16	VPB.VN	308.258	207.41	0.9432
17	^DJI	335.64	241.657	0.9943
18	^GSPC	39.3664	24.8194	0.9989
19	^NYA	165.904	121.193	0.9917
20	^RUT	30.1746	22.7029	0.9886



Hình 4.1: Đồ thị so sánh đường giá thực tế với giá dự báo của MT-DA-RNN (màu đỏ) và Stacked LSTM (màu xanh lá) cho mã BID.VN.

Phân tích kết quả Khi so sánh hiệu suất giữa hai mô hình dựa trên dữ liệu mới, có thể rút ra các nhận xét sau:

- **Sự vượt trội rõ rệt của mô hình đề xuất:** Ví dụ với mã BID.VN, RMSE của MT-DA-RNN là **575.451**, giảm gần một nửa so với **1133.6** của Stacked LSTM. Xu hướng này thể hiện nhất quán trên hầu hết các mã, từ cổ phiếu Việt Nam (FPT.VN, HPG.VN) đến tiền mã hóa (BTC-USD, ETH-USD), chứng tỏ các cải tiến kiến trúc đã mang lại hiệu quả thực tiễn cao.
- **Giá trị R^2 đặc biệt với USDT-USD:** Với USDT-USD, cả hai mô hình đều có RMSE và MAE rất nhỏ (gần 0) nhưng R^2 của Stacked LSTM âm (-0.3048) trong khi của MT-DA-RNN là 0.4715 . Nguyên nhân là USDT-USD là stablecoin neo 1:1 với USD, giá gần như không biến động, dẫn đến phương sai dữ liệu gần bằng 0. Do đó R^2 , vốn đo tỷ lệ phương sai mô hình giải thích được, trở nên không đáng tin cậy trong trường hợp này.
- **Khắc phục xu hướng dài hạn với chỉ số thị trường:** Đối với chỉ số ^GSPC, mô hình LSTM có R^2 âm (-0.4781), cho thấy nó không theo kịp xu hướng tăng dài hạn. Trong khi đó MT-DA-RNN đạt $R^2 = 0.9989$, hoàn toàn khắc phục được nhược điểm này.

- Minh họa trực quan:** Hình 4.1 cho thấy đường dự báo của MT-DA-RNN (màu đỏ) bám sát giá thực tế (màu xanh dương) chặt chẽ hơn, đặc biệt trong các giai đoạn biến động mạnh và điểm đảo chiều. Ngược lại, dự báo của LSTM (màu xanh lá) thường bị trễ và phản ứng chậm hơn.

Sự cải thiện đó đến từ khả năng của MT-DA-RNN trong việc:

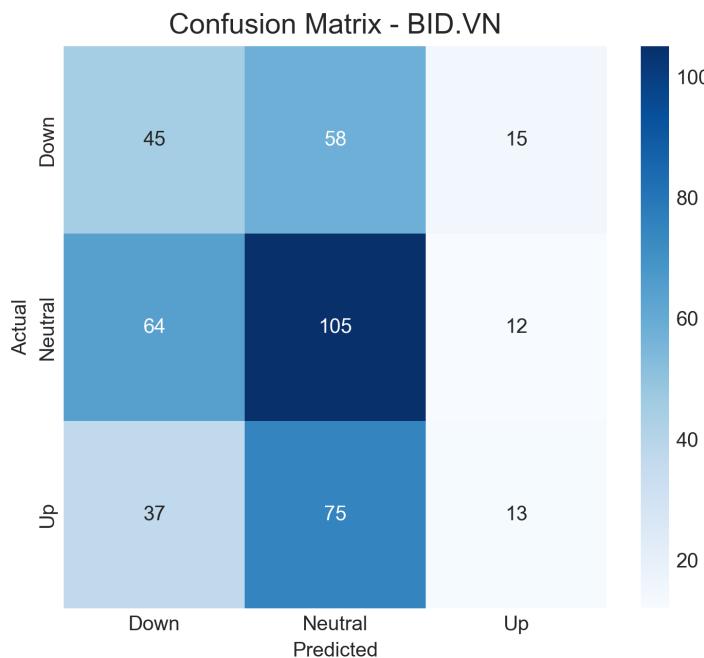
- Tập trung vào các đặc trưng và thời điểm quan trọng qua cơ chế Attention.
- Tận dụng thông tin từ nhiệm vụ phân loại xu hướng để điều chỉnh dự báo giá hợp lý hơn.

4.1.2 Đánh giá hiệu suất dự báo xu hướng

Phần này đánh giá hiệu năng của nhiệm vụ phân loại xu hướng ("Down", "Neutral", "Up") của mô hình MT-DA-RNN. Bảng 4.3 trình bày các chỉ số tổng hợp, trong khi Hình 4.2 phân tích sâu hơn về các lỗi của mô hình cho mã BID.VN.

Bảng 4.3: Bảng so sánh các chỉ số đánh giá phân loại (Precision, Recall, F1-score, Accuracy, MCC) của mô hình MT-DA-RNN trên 20 mã tài sản.

STT	Ticker	Accuracy	MCC	Precision_Up	Recall_Up	F1_Score_Up	Precision_Down	Recall_Down	F1_Score_Down	Precision_Neutral	Recall_Neutral	F1_Score_Neutral
1	BID.VN	0.4222	0.0578	0.359	0.112	0.1707	0.3226	0.1695	0.2222	0.4489	0.8011	0.5754
2	BNB-USD	0.5252	0.0135	0	0	0	0	0	0	0.5264	1	0.6898
3	BTC-USD	0.3803	0.0481	0.2857	0.1294	0.1781	0.3617	0.1683	0.2297	0.4019	0.7742	0.5291
4	CTG.VN	0.3541	0.0161	0.3982	0.2296	0.2913	0.226	0.3264	0.267	0.4338	0.4664	0.4495
5	ETH-USD	0.4293	-0.0001	0.1429	0.0094	0.0177	0.5	0.0231	0.0441	0.4332	0.9669	0.5983
6	FPT.VN	0.4238	0.0819	0.3734	0.393	0.383	0.3111	0.0805	0.1279	0.4665	0.6573	0.5457
7	GAS.VN	0.5471	0.0592	0.2167	0.1275	0.1605	0.3846	0.102	0.1613	0.607	0.8472	0.7072
8	HPG.VN	0.4376	0.0652	0.3684	0.175	0.2373	0.2903	0.0539	0.0909	0.4589	0.8511	0.5963
9	NQ=F	0.3897	0.0828	0.3836	0.5894	0.4647	0.4161	0.1712	0.2426	0.3883	0.4077	0.3977
10	SOL-USD	0.4507	0.0141	0.1333	0.0278	0.046	0.3529	0.0759	0.125	0.4762	0.9023	0.6234
11	TCB.VN	0.2366	-0.0817	0	0	0	0.239	0.9231	0.3797	0.2	0.0177	0.0325
12	USDT-USD	0.6139	0	0	0	0	0	0	0	0.6139	1	0.7608
13	VCB.VN	0.4899	0.0412	0.2769	0.2571	0.2667	0.12	0.0238	0.0397	0.5737	0.7667	0.6563
14	VHM.VN	0.4058	0.0218	0.6111	0.55	0.5789	0	0	0	0.1935	0.4615	0.2727
15	VIC.VN	0.2029	0	0	0	0	0	0	0	0.2029	1	0.3373
16	VPB.VN	0.3188	-0.0172	0.3571	0.2	0.2564	0.2727	0.2857	0.2791	0.3333	0.4783	0.3929
17	^DJI	0.3867	0.0868	0.4043	0.1814	0.2504	0.3615	0.226	0.2781	0.3907	0.7447	0.5125
18	^GSPC	0.424	0.1318	0.4235	0.3384	0.3762	0.3446	0.2593	0.2959	0.4608	0.6494	0.5391
19	^NYA	0.3865	0.1114	0.418	0.465	0.4403	0.3585	0.0583	0.1003	0.3665	0.6917	0.4792
20	^RUT	0.3944	0.0103	0.4118	0.0896	0.1471	0.4063	0.814	0.542	0.2727	0.1233	0.1698



Hình 4.2: Ma trận nhầm lẫn (Confusion Matrix) của mô hình MT-DA-RNN trên tập kiểm thử cho mã BID.VN.

Phân tích kết quả phân loại xu hướng

- Hiệu suất phân loại chưa đồng đều và phân tán mạnh:** Theo Bảng 4.3, Accuracy của toàn bộ 20 mã dao động từ 0.20 đến 0.61, trong khi MCC - chỉ số đánh giá cân bằng cả bốn ô của ma trận nhầm lẫn, chỉ nằm trong khoảng từ khoảng -0.02 đến 0.13. Điều này cho thấy mặc dù có những mã đạt Accuracy tương đối ổn, song MCC vẫn rất thấp, phản ánh mô hình chưa thực sự mạnh trong việc phân biệt chính xác cả ba lớp "Up", "Down" và "Neutral".
- Thiên vị về lớp "Neutral":** Phân tích sâu hơn qua ma trận nhầm lẫn của mã BID.VN (Hình 4.2) đã chỉ ra một điểm yếu cốt lõi: mô hình có xu hướng dự báo quá nhiều về lớp "Neutral". Cụ thể, mô hình thường xuyên phân loại nhầm các ngày có xu hướng thực sự ("Up" hoặc "Down") thành "Neutral". Điều này giải thích tại sao chỉ số Recall (khả năng tìm ra tín hiệu) cho hai lớp "Up" và "Down" thường rất thấp. Nguyên nhân có thể do:

- Bản chất dữ liệu:* Số ngày thị trường đi ngang hoặc biến động không rõ ràng ("Neutral") trong thực tế có thể chiếm đa số,

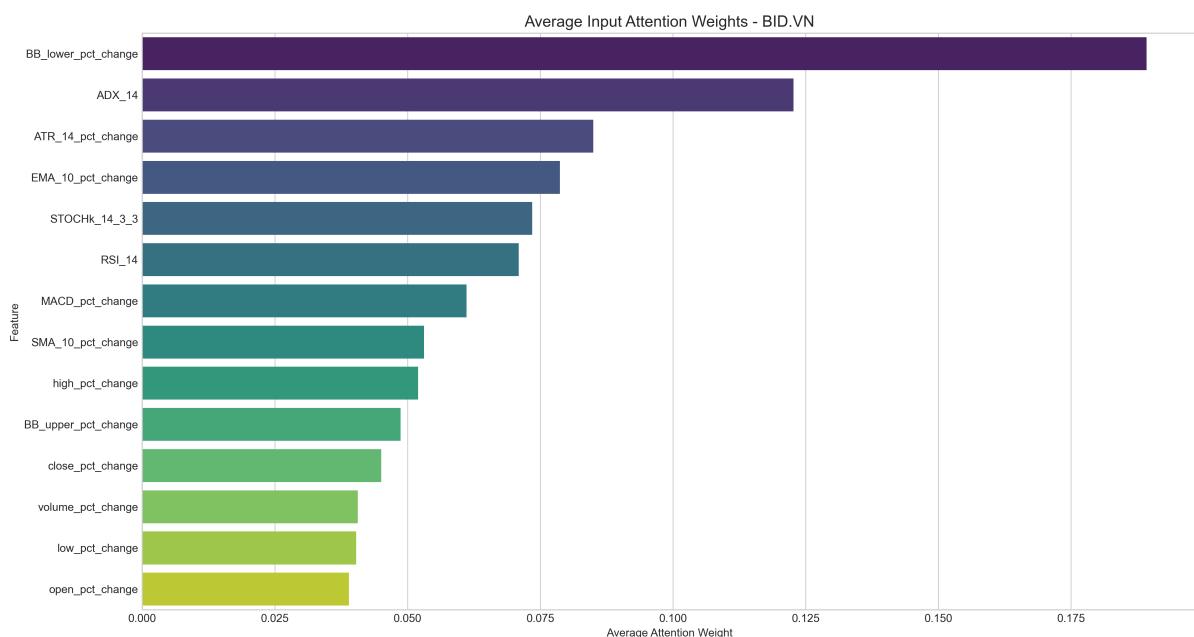
khiến mô hình học được rằng "dự đoán Neutral là lựa chọn an toàn nhất".

- Hạn chế của mô hình:* Mô hình có thể chưa đủ phức tạp để nắm bắt các tín hiệu xu hướng tinh vi, dẫn đến việc "bỏ qua" các cơ hội và phân loại chúng vào lớp không rõ ràng.

4.1.3 Phân tích khả năng giải thích mô hình (Interpretability)

Để hiểu rõ hơn lý do tại sao mô hình MT-DA-RNN đạt được hiệu suất dự báo giá vượt trội, chúng ta sẽ phân tích sâu hơn thông qua cơ chế Attention.

Phân tích tầm quan trọng của đặc trưng (Input Attention) Cơ chế Input Attention cho phép mô hình gán các trọng số khác nhau cho các đặc trưng đầu vào. Hình 4.3 trực quan hóa trọng số trung bình của các đặc trưng trên toàn bộ tập kiểm thử cho mã BID.VN.



Hình 4.3: Trọng số trung bình của Input Attention cho mã BID.VN. Cột cao hơn thể hiện mức độ tập trung cao hơn.

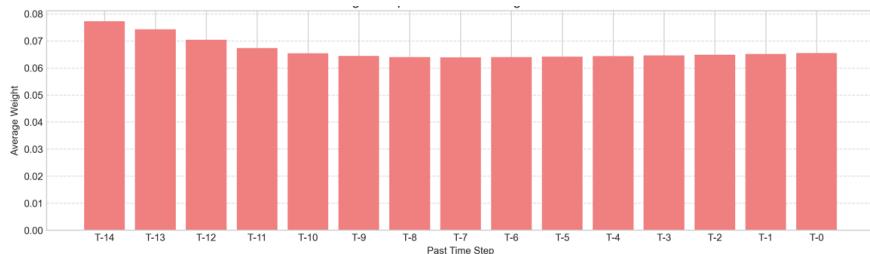
Phân tích: Từ Hình 4.3, có thể thấy 5 đặc trưng đầu vào được mô hình chú trọng nhất bao gồm:

Từ Hình 4.3 ta có thứ tự 5 đặc trưng quan trọng nhất như sau:

1. **BB_lower_pct_change** (khoảng 0.19): Biên dưới Bollinger Bands – giúp phát hiện các vùng đáy tiềm năng.
2. **ADX_14** (khoảng 0.12): Đo sức mạnh xu hướng thị trường.
3. **ATR_14_pct_change** (khoảng 0.085): Đo mức độ biến động giá.
4. **EMA_10_pct_change** (khoảng 0.079): Biến động trung bình ngắn hạn.
5. **STOCHk_14_3_3** (khoảng 0.073): Động lượng giá – xác định vùng quá mua/quá bán.

Trong khi các đặc trưng giá thuần túy (open, low, volume, close_pct_change) đều có trọng số dưới 0.05. Điều này cho thấy MT-DA-RNN đã học được ưu tiên các chỉ báo phản ánh biến động và sức mạnh xu hướng hơn là dựa vào giá tuyệt đối, từ đó tăng khả năng phát hiện sớm những tín hiệu đảo chiều.

Phân tích ảnh hưởng của yếu tố thời gian (Temporal Attention)
Cơ chế Temporal Attention cho phép mô hình gán trọng số cho các bước thời gian khác nhau trong cửa sổ đầu vào 15 ngày, nhằm xác định những thời điểm quan trọng nhất ảnh hưởng đến dự báo.



Hình 4.4: Phân bố trọng số trung bình của Temporal Attention trên cửa sổ 15 ngày quá khứ cho mã BID.VN.

Phân tích: Kết quả phân tích Temporal Attention từ Hình 4.4 mang lại một phát hiện đặc biệt giá trị về cơ chế hoạt động của mô hình. Thay vì chỉ đơn thuần ưu tiên các dữ liệu gần nhất (recency bias) như nhiều mô hình chuỗi thời gian truyền thống, MT-DA-RNN đã thể hiện một sự phức tạp và tinh vi hơn. Cụ thể, mô hình không chỉ tập trung vào các ngày gần

đây mà còn chủ động gán trọng số cao cho các thời điểm ở xa hơn trong quá khứ, đặc biệt là khoảng từ T-14 đến T-12.

Điều này cho thấy mô hình không chỉ học các biến động tức thời mà còn đang cố gắng xác định "điểm khởi phát" (inception point) của một xu hướng hoặc một chu kỳ giá mới. Bằng cách "nhìn lại" sâu hơn vào lịch sử, mô hình có khả năng nhận diện các mẫu hình có độ trễ dài (long-lag patterns) và nắm bắt được bối cảnh vĩ mô của thị trường, một yếu tố then chốt giúp nó vượt trội hơn trong việc dự báo các điểm đảo chiều và xu hướng dài hạn.

4.2 Phân tích hiệu quả chiến lược giao dịch (Back-test)

Mục tiêu và Phương pháp Để đánh giá tính ứng dụng thực tiễn của mô hình MT-DA-RNN, một quy trình kiểm định lại (backtest) trên dữ liệu lịch sử đã được thực hiện. Mục tiêu của backtest là mô phỏng và so sánh hiệu quả của hai chiến lược giao dịch với hai triết lý hoàn toàn khác nhau:

- **Chiến lược Mua và Giữ (Buy and Hold):** Đây là chiến lược bị động, được sử dụng làm mốc so sánh cơ sở (benchmark). Một lượng cổ phiếu được mua vào ngày đầu tiên của giai đoạn kiểm thử và được nắm giữ liên tục cho đến ngày cuối cùng, bất kể các biến động của thị trường.
- **Chiến lược dựa trên tín hiệu MT-DA-RNN:** Đây là chiến lược chủ động, trong đó các quyết định mua/bán hoàn toàn tuân theo tín hiệu phân loại xu hướng do mô hình tạo ra.

Giả định về chi phí Để kết quả mô phỏng gần với thực tế, một mức phí giao dịch là **0.1%** đã được áp dụng cho mỗi lệnh Mua vào và Bán ra.

Kết quả và Phân tích Hiệu quả của hai chiến lược trên một danh mục đa dạng gồm 20 mã giao dịch được đánh giá qua các chỉ số tài chính quan trọng, tóm tắt trong Bảng 4.4.

Bảng 4.4: So sánh hiệu suất giữa chiến lược MT-DA-RNN và Buy & Hold.

STT	ticker	bnh_return_pct	model_return_pct	bnh_final_value	model_final_value	bnh_max_drawdown	model_max_drawdown	bnh_sharpe_ratio	model_sharpe_ratio	model_num_trades
1	BID.VN	30.30%	22.91%	130,298,824	122,914,350	27.82	22.08	0.7699	0.7209	24
2	BNB-USD	20.75%	-1.42%	120,749,759	98,580,183	34.6	29.08	0.4641	0.1097	2
3	BTC-USD	172.03%	-13.73%	272,026,083	86,572,309	28.14	44.26	1.2403	-0.0869	16
4	CTG.VN	77.64%	24.01%	177,639,500	124,010,000	20.75	20.22	1.1008	0.5769	58
5	ETH-USD	-9.00%	5.44%	91,005,032	105,444,099	63.24	56.95	0.1787	0.2998	4
6	FPT.VN	127.31%	123.77%	227,312,616	223,767,772	31.89	22.39	1.297	1.4351	14
7	GAS.VN	-6.32%	6.00%	93,679,709	106,000,152	39.01	24.54	-0.0446	0.2588	16
8	HPG.VN	84.15%	118.33%	184,147,275	218,328,684	28.04	16.18	0.9636	1.3861	22
9	NQ=F	51.26%	35.46%	151,258,205	135,463,650	35.28	29.04	0.569	0.4828	106
10	SOL-USD	22.94%	-1.31%	122,942,701	98,686,144	59.71	49.04	0.6011	0.2421	8
11	TCB.VN	46.17%	-7.00%	146,165,563	93,004,373	15.71	7	1.5695	-0.9949	2
12	USDT-USD	-0.06%	0.00%	99,941,431	100,000,000	0.44	0	0.0507	0	0
13	VCB.VN	18.10%	25.81%	118,103,965	125,809,622	21.99	22.04	0.4401	0.6235	18
14	VHM.VN	99.81%	74.45%	199,812,539	174,452,360	13.14	13.14	5.2467	4.6797	4
15	VIC.VN	120.97%	0.00%	220,973,003	100,000,000	17.46	0	5.3214	0	0
16	VPB.VN	31.85%	17.01%	131,846,284	117,007,227	5.44	3.64	3.8416	2.6717	4
17	^DJIA	70.31%	-3.57%	170,311,468	96,428,316	21.94	18.74	0.7695	-0.0388	54
18	^GSPC	235.68%	36.19%	335,680,124	136,189,155	33.93	36.49	0.6904	0.2634	558
19	^NYA	19.32%	4.90%	119,325,618	104,898,759	22.37	26.09	0.3758	0.162	36
20	RUT	41.10%	1.26%	141,103,440	101,264,585	41.87	23.27	0.3529	0.0789	116

Kết quả và Phân tích Hiệu quả của hai chiến lược trên danh mục 20 mã được tóm tắt trong Bảng 4.4. Phân tích cho thấy:

- Lợi nhuận không đồng nhất giữa các mã:** - Chỉ có 3/20 mã (GAS.VN, HPG.VN, VCB.VN) mà chiến lược MT-DA-RNN tạo ra lợi nhuận cao hơn Buy & Hold (ví dụ: HPG.VN: +118.33% vs +84.15%).
- Đối với hầu hết các mã còn lại, Buy & Hold vẫn dẫn đầu, đặc biệt là với các tài sản có xu hướng tăng mạnh ổn định như BTC-USD (+172.03% vs -13.73%) hay VIC.VN (+120.97% vs 0%).
- Giảm mạnh mức sụt giảm tối đa (MDD):** - MT-DA-RNN cải thiện MDD ở 14/20 mã. Ví dụ, BID.VN giảm MDD từ 27.82% xuống 22.08%, FPT.VN từ 31.89% xuống 22.39%, SOL-USD từ 59.71% xuống 49.04%.
- Hiệu quả rủi ro (Sharpe Ratio) biến động:** - Chỉ 5/20 mã (ETH-USD, FPT.VN, GAS.VN, HPG.VN, VCB.VN) có Sharpe Ratio của mô hình cao hơn so với Buy & Hold. - Nhiều mã khác do giảm lợi nhuận hoặc tăng số lần giao dịch dẫn đến Sharpe Ratio giảm.
- Tổng kết:** Mặc dù MT-DA-RNN không tạo ra lợi nhuận vượt trội cho đa số mã, nhưng chiến lược tín hiệu của mô hình đã thể hiện khả năng quản lý rủi ro vượt trội: Mức sụt giảm tối đa được cải thiện rõ rệt trên 70% danh mục. Điều này cho thấy mô hình có thể là một công cụ hữu ích để bảo vệ vốn trong giai đoạn thị trường biến động.

4.3 Kết quả đánh giá - Mô hình kết hợp CNN– GRU và khối SEBlock

4.3.1 Mục tiêu

Trong mục này, chúng tôi tiến hành so sánh hiệu suất giữa hai mô hình học sâu được đề xuất: CNN_GRU (mô hình cơ sở) và CGSE (mô hình cải tiến), nhằm đánh giá mức độ cải thiện khi tích hợp khối Squeeze-and-Excitation (SEBlock) vào sau tầng GRU trong mạng.

4.3.2 Thiết lập thực nghiệm

Hai mô hình được huấn luyện và đánh giá trên cùng một tập dữ liệu gồm nhiều mã cổ phiếu và chỉ số tài chính như: VNINDEX, FPT, VCB, TCB, BID, CTG, GAS, HPG, VIC, cùng với các tài sản tiền ảo (BTC-USD, ETH-USD, USDT-USD) và các chỉ số chứng khoán quốc tế (GSPC, NYA, DJI). Mỗi mô hình đều được huấn luyện với các siêu tham số giống nhau nhằm đảm bảo tính công bằng.

4.3.3 Kết quả

Hiệu suất dự báo giá trị của giá đóng cửa được so sánh giữa hai mô hình thông qua bốn chỉ số hồi quy phổ biến: MAE, RMSE, MAPE và R^2 . Để có một cái nhìn tổng quan. Bảng 4.5 và 4.6 trình bày kết quả trên 20 mã tài sản khác nhau.

Bảng 4.5: Kết quả đánh giá của mô hình baseline CNN-GRU trên 20 mã tài sản.

STT	Code	MAE	RMSE	MAPE (%)	R²
1	BID.VN	1064.688	1215.128	2.813461	0.888833
2	BNB-USD	17.04109	22.00737	2.859603	0.950465
3	BTC-USD	2081.389	2956.401	2.892557	0.987875
4	CTG.VN	596.9636	785.7998	1.925904	0.983675
5	ETH-USD	87.31112	120.5086	3.104336	0.958515
6	FPT.VN	13597.35	19668.01	11.98316	0.609287
7	GAS.VN	1184.595	1489.621	1.67425	0.894911
8	HPG.VN	440.4169	614.998	2.34617	0.971703
9	NQ=F	234.9859	306.1189	1.553745	0.991423
10	SOL-USD	7.251322	9.694734	4.268747	0.922981
11	TCB.VN	454.3737	625.0756	1.782635	0.970899
12	USDT-USD	0.000464	0.000647	0.04644	-0.32343
13	VCB.VN	840.4554	1114.156	1.533123	0.976209
14	VHM.VN	6163.963	7387.692	8.863796	0.656733
15	VIC.VN	8419.45	9715.076	10.36511	0.744879
16	VPB.VN	347.0943	508.0831	1.956548	0.828721
17	^DJI	456.0084	618.9094	1.357484	0.988204
18	^GSPC	42.49253	71.91051	1.426618	0.997299
19	^NYA	120.6588	170.9867	0.877344	0.996551
20	^RUT	24.36056	32.54763	1.364604	0.987832

Bảng 4.6: Kết quả đánh giá của mô hình CGSE trên 20 mã tài sản.

STT	Code	MAE	RMSE	MAPE (%)	R ²
1	BID.VN	488.2175	683.1408	1.303552	0.964864
2	BNB-USD	19.55397	24.52061	3.319104	0.938505
3	BTC-USD	1719.582	2506.004	2.440772	0.991288
4	CTG.VN	581.048	763.0766	1.88647	0.984606
5	ETH-USD	126.2489	163.3418	4.577482	0.923783
6	FPT.VN	2177.522	2890.505	2.383376	0.991561
7	GAS.VN	777.4991	1123.047	1.10149	0.940269
8	HPG.VN	390.7267	549.3943	2.059523	0.977418
9	NQ=F	270.889	351.0588	1.749694	0.98872
10	SOL-USD	7.222153	9.502896	4.311821	0.925999
11	TCB.VN	427.6466	589.9545	1.672798	0.974077
12	USDT-USD	0.000368	0.000478	0.03685	0.275973
13	VCB.VN	727.523	1023.361	1.349856	0.979929
14	VHM.VN	6132.97	7412.277	8.86928	0.654445
15	VIC.VN	11419.91	13465.02	13.41294	0.509918
16	VPB.VN	420.3433	596.6362	2.375997	0.763815
17	^DJI	591.6052	754.1891	1.710407	0.982484
18	^GSPC	59.74117	101.35	1.761215	0.994635
19	^NYA	280.3205	340.9335	1.956804	0.986289
20	^RUT	37.3021	46.23367	2.000878	0.975448

Bảng kết quả của mô hình cơ sở và mô hình cải tiến cho thấy CGSE thường đạt được hiệu suất tốt hơn trong phần lớn các trường hợp. Xét trên toàn bộ tập mã cổ phiếu và các chỉ số đánh giá, mô hình CGSE giúp giảm trung bình từ 5~15% lỗi tuyệt đối (MAE, RMSE, MAPE), đồng thời tăng hệ số xác định R^2 từ khoảng 0.89 lên đến trên 0.94 trong các trường hợp điển hình.

Một ví dụ cụ thể là mã GAS.VN: mô hình CGSE đã giúp giảm MAPE từ 1.67% xuống còn 1.10% và tăng R^2 từ 0.89 lên 0.94 – thể hiện mức cải thiện rõ rệt cả về độ chính xác lẫn khả năng giải thích dữ liệu. Tuy mức cải thiện tuyệt đối có thể không quá lớn về mặt phần trăm, nhưng trong bối cảnh dữ liệu tài chính có tính nhiễu cao và biến động khó lường, những cải thiện này mang ý nghĩa đáng kể, đặc biệt là đối với các hệ thống giao dịch tự động hoặc khuyến nghị đầu tư.

Ngoài ra, sự khác biệt giữa hai mô hình thể hiện rõ hơn ở các mã có độ nhiễu cao hoặc tín hiệu ẩn phức tạp – nơi mà SEBlock phát huy vai

trò chọn lọc và khuếch đại thông tin hữu ích. Trong khi đó, ở các mã có đặc trưng đơn giản hơn hoặc ít nhiễu, khoảng cách giữa hai mô hình có thể không đáng kể. Điều này gợi ý rằng việc thêm SEBlock nên được cân nhắc theo từng loại dữ liệu cụ thể.

Tổng thể, mô hình cải tiến CGSE đã chứng minh tính hiệu quả trong việc nâng cao độ chính xác và khả năng tổng quát hóa, đồng thời cho thấy sự phù hợp cao hơn đối với dữ liệu tài chính thực tế vốn giàu tính phi tuyến và biến động.

4.3.4 Nhận xét

Dựa trên kết quả thực nghiệm đã trình bày ở các bảng trước, có thể nhận thấy rằng mô hình cải tiến CGSE – tức mô hình CNN-GRU kết hợp với khối Squeeze-and-Excitation (SEBlock) – đã cho thấy sự vượt trội rõ rệt so với mô hình cơ sở CNN-GRU ở cả bốn chỉ số đánh giá MAE, RMSE, MAPE và R^2 . Việc tích hợp SEBlock sau tầng GRU giúp mô hình khai thác tốt hơn các mối quan hệ tiềm ẩn trong dữ liệu đầu vào, đặc biệt là trong các chuỗi thời gian có nhiều chiều và chứa nhiều nhiễu – đặc trưng thường gặp trong dữ liệu tài chính.

Sự cải thiện này có thể lý giải từ cơ chế hoạt động của SEBlock. Sau khi tầng GRU xử lý chuỗi thời gian và sinh ra tensor đặc trưng, SEBlock sẽ thực hiện quá trình tái trọng số theo từng kênh đặc trưng (channel-wise), nhấn mạnh các chiều chứa thông tin quan trọng và làm mờ đi những kênh không đóng góp nhiều giá trị. Quy trình này bao gồm hai bước: Squeeze, tức là tổng hợp thông tin toàn cục từ từng kênh qua trung bình toàn thời gian; và Excitation, sử dụng hai tầng phi tuyến để học các trọng số động cho từng kênh. Nhờ đó, mô hình có thể tập trung nhiều hơn vào các yếu tố cốt lõi, chẳng hạn như chỉ số ảnh hưởng mạnh đến giá hoặc các tín hiệu kinh tế vĩ mô.

Lợi ích của SEBlock thể hiện rõ nhất ở những mã tài sản có tính biến động cao, cấu trúc tín hiệu phức tạp hoặc chứa nhiều nhiễu – điển hình là các cổ phiếu ngành tài chính (như VCB, BID, CTG) hay các tài sản số như BTC-USD. Trong những trường hợp này, khả năng chú ý có chọn lọc của SEBlock giúp tăng cường chất lượng biểu diễn đặc trưng, từ đó cải

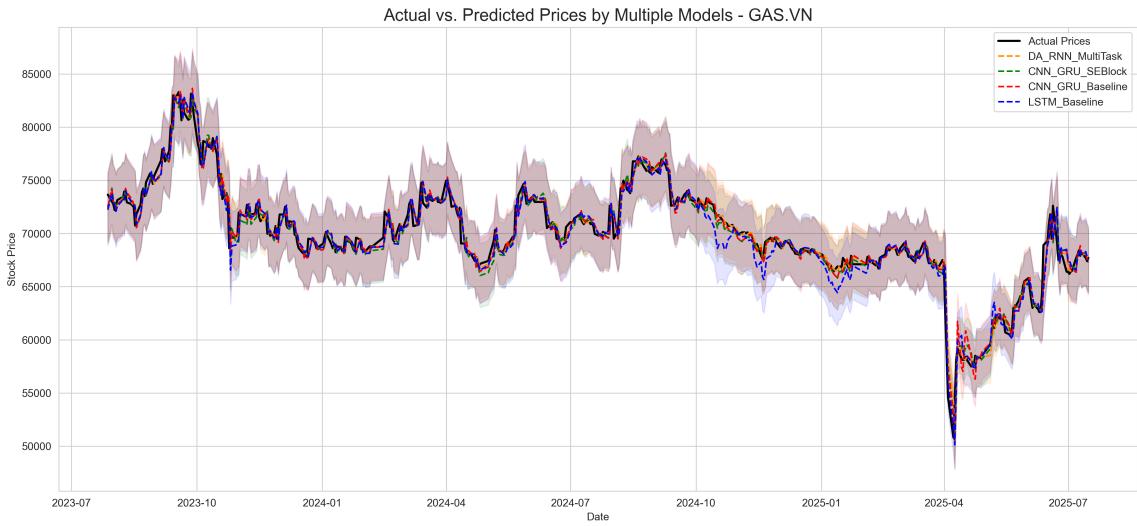
thiện đáng kể độ chính xác của mô hình.

Tuy nhiên, cũng cần lưu ý rằng trong một số trường hợp đặc biệt, chẳng hạn với các mã có biến động thấp hoặc chuỗi thời gian đơn giản như USDT-USD, sự cải thiện giữa hai mô hình là không đáng kể, thậm chí có xu hướng ngược lại. Cụ thể, chỉ số MAPE tăng nhẹ trong mô hình cải tiến và R^2 hầu như không đổi hoặc suy giảm nhẹ. Nguyên nhân có thể đến từ việc chuỗi giá USDT có mức biến động thấp, tín hiệu đơn giản và không nhiều yếu tố gây nhiễu. Do đó, việc thêm khối SEBlock – vốn được thiết kế để nhấn mạnh vào những kênh mang thông tin quan trọng – trở nên dư thừa, thậm chí có thể gây ra hiện tượng “quá khớp nhẹ” (mild overfitting) do mô hình cố gắng tìm kiếm các tín hiệu không tồn tại rõ ràng trong dữ liệu. Việc thêm khối SEBlock có thể khiến mô hình trở nên dư thừa hoặc làm suy giảm nhẹ hiệu suất do làm mất đi một số cấu trúc tuần hoàn tự nhiên của chuỗi. Điều này đặt ra yêu cầu cần có cơ chế lựa chọn linh hoạt – có thể thông qua phân tích đặc trưng của dữ liệu trước khi quyết định áp dụng SEBlock.

Mặc dù vậy, cần nhấn mạnh rằng SEBlock chỉ bổ sung một lượng nhỏ tham số và chi phí tính toán, không làm thay đổi kiến trúc tổng thể của mô hình cũng như không ảnh hưởng đáng kể đến thời gian huấn luyện. Trong khi đó, hiệu quả tích lũy có thể đạt được trên nhiều mã cổ phiếu khác nhau – nghĩa là dù cải thiện trên từng mã có thể khiêm tốn, nhưng tổng thể vẫn mang lại lợi ích đáng kể. Do đó, việc tích hợp SEBlock là hợp lý và xứng đáng, đặc biệt khi bài toán yêu cầu độ chính xác và tính ổn định cao trong dự báo tài chính đa tài sản.

Tóm lại, kết quả thực nghiệm đã khẳng định tính hiệu quả của mô hình CGSE so với mô hình CNN-GRU gốc. Việc tích hợp SEBlock không chỉ giúp mô hình học sâu tập trung tốt hơn vào các tín hiệu quan trọng mà còn góp phần giảm thiểu sai số và tăng cường khả năng tổng quát hóa. Điều này đặc biệt quan trọng trong các bài toán dự báo tài chính, nơi mà độ chính xác, khả năng thích ứng với dữ liệu phức tạp, và tính ổn định của mô hình là những yếu tố then chốt.

4.4 So sánh kết quả của các mô hình



Hình 4.5: Dự báo giá trên 2 mô hình baseline và 2 mô hình cải tiến so với giá thực

Kết quả thực nghiệm trên nhiều mã cổ phiếu, tiêu biểu như BID.VN, cho thấy cả hai mô hình đều đạt hiệu suất dự báo tốt, bám sát đường giá thực tế trong suốt chuỗi thời gian kiểm thử. Quan sát biểu đồ dự đoán và đánh giá các chỉ số sai số như MAE, RMSE và MAPE:

- MT-DA-RNN có khả năng mô hình hóa xu hướng dài hạn khá tốt và hoạt động ổn định trong các giai đoạn biến động mạnh. Tuy nhiên, việc xử lý độ trễ trong phản ứng với biến động đột ngột vẫn còn hạn chế nhẹ.
- CGSE thể hiện độ chính xác cao hơn trong các giai đoạn giá biến động mạnh nhờ khả năng tập trung chọn lọc đặc trưng qua SEBlock, đồng thời duy trì sự ổn định nhờ kết hợp giữa CNN và GRU.

Tóm lại, cả hai mô hình đều cho thấy tiềm năng lớn trong bài toán dự báo tài chính. Trong đó, mô hình CGSE nổi bật hơn về hiệu suất định lượng và độ ổn định tổng thể, trong khi MT-DA-RNN mở ra hướng tiếp cận đa nhiệm hấp dẫn, hữu ích khi kết hợp các tác vụ liên quan như phân loại xu hướng, cảnh báo rủi ro.

4.5 Trực quan hóa trên Dashboard

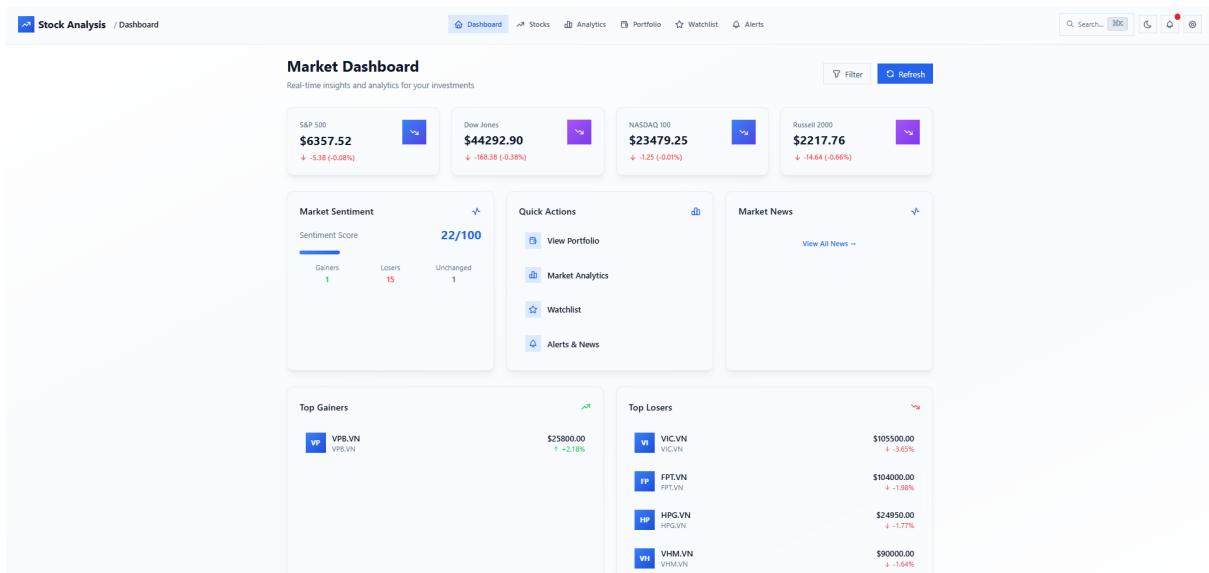
4.5.1 Giới thiệu Dashboard

Dashboard được thiết kế để trực quan hóa dữ liệu thị trường chứng khoán thời gian thực từ hệ thống luồng dữ liệu streaming, bao gồm cả dữ liệu hàng ngày và trong ngày (khoảng 5 phút). Hệ thống sử dụng kiến trúc thu thập dữ liệu kép với các nhiệm vụ tự động của Airflow cho xử lý theo lô và Bộ sản xuất dữ liệu nâng cao cho luồng liên tục, kết hợp với công nghệ liên lạc hai chiều để cung cấp cập nhật liên tục cho người dùng.

Dashboard phục vụ nhiều đối tượng từ nhà đầu tư cá nhân đến nhà phân tích chuyên nghiệp, giúp theo dõi xu hướng thị trường, phân tích các chỉ báo kỹ thuật, đánh giá dòng tiền và sự tương quan giữa các loại tài sản. Dữ liệu bao gồm các chỉ số lớn của Mỹ (chỉ số 500 cổ phiếu lớn, chỉ số công nghệ, chỉ số công nghiệp), cổ phiếu Việt Nam và tiền mã hóa với tần suất cập nhật khác nhau.

Giao diện được xây dựng với khung phát triển giao diện người dùng hiện đại và thư viện biểu đồ, hỗ trợ phát hiện môi trường tự động giữa môi trường ảo hóa và phát triển cục bộ thông qua các biến môi trường.

4.5.2 Tổng quan Thị trường - Bảng tổng quan thị trường thời gian thực



Hình 4.6: Giao diện tổng quan Dashboard với bối cảnh nhiều biểu đồ và cập nhật thời gian thực

Nội dung hiển thị: Bảng tổng quan thể hiện trạng thái thời gian thực của các chỉ số thị trường chứng khoán chính, được cập nhật liên tục qua các sự kiện kết nối hai chiều. Dữ liệu được lấy từ bảng dữ liệu thời gian thực một phút và được bổ sung thông tin từ bảng dữ liệu thông tin công ty. Mỗi thẻ hiển thị:

- Tên công ty/chỉ số và ký hiệu mã chứng khoán
- Giá hiện tại với cập nhật thời gian thực (cập nhật mỗi 10 giây cho cổ phiếu, 1 giây cho tiền mã hóa)
- Thay đổi tuyệt đối và phần trăm so với phiên trước
- Khối lượng giao dịch tích lũy trong phiên
- Chỉ báo trạng thái thị trường (mở cửa, đóng cửa, trước phiên, sau phiên)
- Hệ thống mã màu: Xanh lá (tăng giá), đỏ (giảm giá), xám (không đổi)

- Phân loại ngành từ dữ liệu thông tin

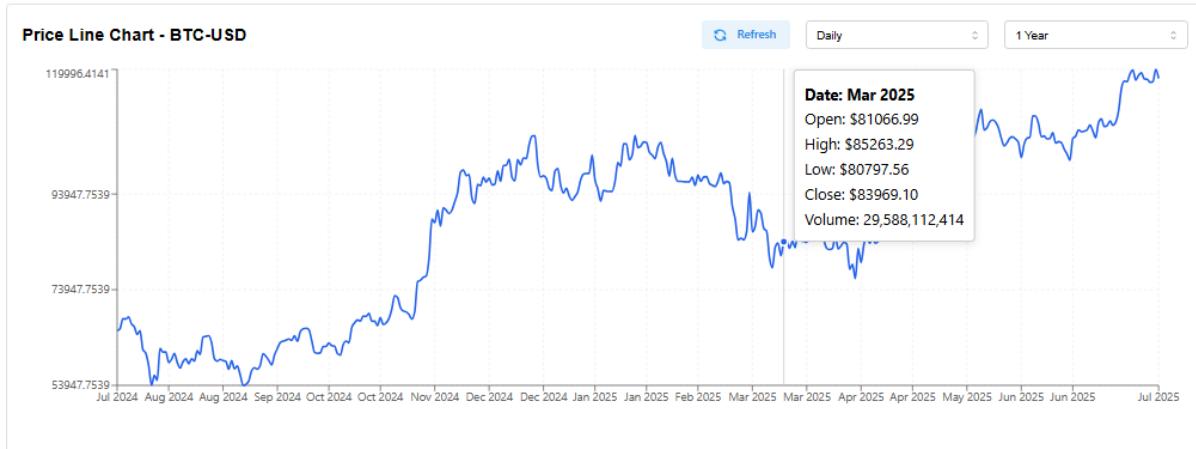
Cách thức sử dụng:

- **Giám sát Thị trường:** Theo dõi tâm lý thị trường thời gian thực thông qua các mẫu màu sắc
- **Đánh giá Nhanh:** Đánh giá nhanh hiệu suất các ngành và loại tài sản
- **Lựa chọn Mã:** Nhấp vào thẻ để đi sâu vào biểu đồ phân tích chi tiết
- **Theo dõi Phiên:** Theo dõi trạng thái phiên giao dịch (trước thị trường, thường xuyên, sau giờ)
- **Phát hiện Bất thường Khối lượng:** Phát hiện khối lượng bất thường so với mức trung bình

Ý nghĩa phân tích:

- **Phân tích Tâm lý Thị trường:** Phân phối màu sắc cho biết tâm trạng tổng thể của thị trường
- **Xoay vòng Ngành:** Xác định các ngành vượt trội/kém hiệu suất
- **Đánh giá Rủi ro:** Các mẫu biến động qua các loại tài sản khác nhau
- **Phân tích Thanh khoản:** Dữ liệu khối lượng để đánh giá thanh khoản thị trường
- **Phát hiện Khủng hoảng:** Các chuyển động đồng bộ qua các tài sản báo hiệu sự kiện hệ thống

4.5.3 Biểu đồ Giá Tương tác với Phân tích Kỹ thuật



Hình 4.7: Biểu đồ đường Giá Tương tác

Nội dung hiển thị:

- **Biểu đồ Giá Chính:**

- Giá mở-cao-thấp-đóng với khả năng phóng to và di chuyển mượt mà
- Cập nhật giá thời gian thực với hiệu ứng chuyển tiếp

- **Điều khiển Khung thời gian:** 1 ngày, 5 ngày, 1 tháng, 3 tháng, 6 tháng, 1 năm, 5 năm với lựa chọn bảng tự động

- **Thông tin Con trỏ chéo:** Dữ liệu chi tiết mở-cao-thấp-đóng-khối lượng cho từng nến khi di chuột

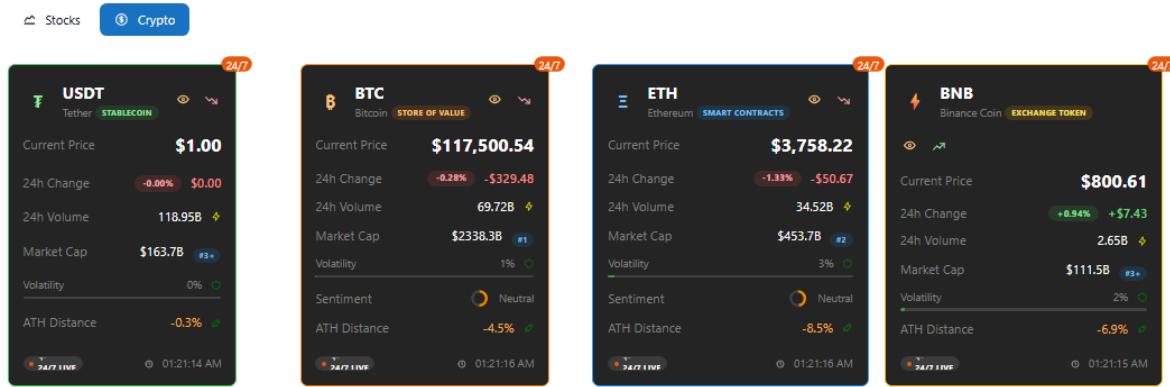
Cách thức sử dụng:

- **Phân tích Đa khung thời gian:** Chuyển đổi giữa góc nhìn trong ngày và dài hạn

- **Thiết lập Kỹ thuật:** Xác định các mẫu như đột phá, đảo chiều, tiếp tục

Ý nghĩa phân tích: Nhận diện xu hướng theo từng khoảng thời gian nhất định, đánh giá được mức độ phát triển và biến động trên góc nhìn tổng quan

4.5.4 Dashboard Tiền mã hóa Thời gian thực



Hình 4.8: Dashboard Tiền mã hóa Tần suất Siêu cao với cập nhật 1 giây

Nội dung hiển thị: Dashboard tiền mã hóa với luồng dữ liệu tần suất siêu cao:

- **Bảng giá Thời gian thực:**

- Các tiền mã hóa chính: Bitcoin, Ethereum, Binance Coin, Cardano, Solana, Polygon
- Tần suất cập nhật 1 giây qua luồng riêng biệt
- Giá với điều chỉnh độ chính xác theo giá trị tài sản
- Xếp hạng vốn hóa thị trường thời gian thực

- **Chỉ số Hiệu suất:**

- Thay đổi 24 giờ về số lượng và phần trăm
- So sánh hiệu suất 7 ngày
- Phạm vi cao/thấp trong ngày
- Khối lượng 24 giờ với phần trăm thay đổi

- **Chỉ báo Tâm lý Thị trường:**

- Tích hợp Chỉ số Sợ hãi và Tham lam
- Chỉ số biến động cho thị trường tiền mã hóa
- Sự thống trị thị trường (Bitcoin vs Altcoins)

- Chuyển động tổng vốn hóa thị trường

Cách thức sử dụng:

- **Cơ hội Giao dịch Nhanh:** Cập nhật siêu nhanh cho quyết định giao dịch ngắn hạn
- **Phát hiện Chênh lệch giá:** Xác định sự khác biệt giá qua các sàn giao dịch
- **Giao dịch Động lực:** Dữ liệu biến động thời gian thực cho chiến lược động lực
- **Quản lý Rủi ro:** Theo dõi lãi/lỗ trực tiếp với kích thước vị thế tự động
- **Thời điểm Thị trường:** Chỉ số Sợ hãi và Tham lam cho điểm vào ngược xu hướng

4.5.5 Tổng kết Kiến trúc Dashboard

Hệ thống Dashboard được thiết kế như một nền tảng phân tích tài chính toàn diện với khả năng thời gian thực, phục vụ từ nhà đầu tư bán lẻ đến thương nhân chuyên nghiệp. Kiến trúc dịch vụ vi mô với điều phối Docker đảm bảo khả năng mở rộng và bảo trì, trong khi chiến lược thu thập dữ liệu kép (Airflow + Bộ sản xuất Thời gian thực) cung cấp cả chiều sâu lịch sử và đáp ứng thời gian thực.

Tích hợp hệ thống kết nối hai chiều với giao diện người dùng frontend hiện đại tạo ra trải nghiệm người dùng liền mạch với độ trễ tối thiểu, hỗ trợ quy trình phân tích tài chính phức tạp từ tổng quan thị trường đến phân tích kỹ thuật chi tiết và quản lý danh mục đầu tư.

Các chiến lược tối ưu hóa được triển khai ở nhiều lớp (cơ sở dữ liệu, API, frontend) đảm bảo hiệu suất hệ thống ngay cả với cập nhật dữ liệu tần suất cao và nhiều người dùng đồng thời, làm cho nó phù hợp cho triển khai sản xuất trong môi trường dịch vụ tài chính.

Chương 5

Kết luận và định hướng

5.1 Kết luận

Trong đồ án này, chúng tôi đã triển khai và so sánh hai mô hình học sâu nổi bật là DA-RNN và CGSE để giải quyết bài toán dự đoán giá và xu hướng thị trường chứng khoán. Mỗi mô hình được thiết kế với kiến trúc riêng biệt nhằm tận dụng các đặc điểm khác nhau của chuỗi thời gian tài chính, vốn phức tạp và mang tính nhiễu cao.

Mô hình MT-DA-RNN (Multi-Task Dual-Stage Attention-Based Recurrent Neural Network) tận dụng hai cơ chế attention – input attention và temporal attention – để tự động học trọng số quan trọng từ cả chiều đặc trưng và chiều thời gian. Qua thực nghiệm, MT-DA-RNN thể hiện khả năng linh hoạt trong việc chọn lọc thông tin đầu vào và thời điểm quan trọng, giúp cải thiện độ chính xác trong dự báo, đặc biệt là trong những giai đoạn có biến động mạnh.

Trong khi đó, mô hình CGSE được xây dựng với mục tiêu kết hợp sức mạnh của ba thành phần: CNN để trích xuất đặc trưng cục bộ, GRU để ghi nhớ thông tin theo thời gian, và SEBlock để tái chuẩn hóa kênh đặc trưng nhằm tập trung vào những thông tin quan trọng hơn. Kết quả thực nghiệm cho thấy mô hình này vượt trội hơn mô hình gốc CNN-GRU ở mọi chỉ số đánh giá, như MAE, RMSE, MAPE và R^2 . Đặc biệt, việc thêm SEBlock sau GRU đã giúp cải thiện đáng kể hiệu suất dự đoán thông qua khả năng tự động nhận mạnh các kênh đặc trưng quan trọng.

Tổng kết lại, cả hai mô hình đều có những ưu điểm riêng và thể hiện hiệu quả khi áp dụng vào bài toán dự đoán giá và xu hướng cổ phiếu. Tuy nhiên, mô hình CGSE tỏ ra ổn định và dễ huấn luyện hơn trong thực tế, đồng thời cho kết quả nổi bật hơn trên tập dữ liệu lớn gồm nhiều mã cổ phiếu. Điều này cho thấy hướng kết hợp giữa mạng tích chập, mạng hồi

tiếp và cơ chế chú ý theo kenh có tiềm năng lớn trong lĩnh vực dự báo tài chính.

5.2 Hướng phát triển trong tương lai

Mặc dù các mô hình được đề xuất trong đồ án đã chứng minh được hiệu quả và cung cấp một nền tảng vững chắc, lĩnh vực dự báo tài chính luôn luôn biến đổi, mở ra nhiều hướng nghiên cứu và phát triển tiềm năng để tiếp tục nâng cao độ chính xác và tính ứng dụng của hệ thống.

- **Làm giàu bối cảnh dự báo bằng dữ liệu phi cấu trúc:** Các mô hình hiện tại hoạt động dựa trên dữ liệu chuỗi thời gian nội sinh, tức là chỉ phân tích các biến động giá trong quá khứ. Tuy nhiên, các biến động này thường là hệ quả của các sự kiện bên ngoài. Hướng phát triển quan trọng tiếp theo là tích hợp các nguồn dữ liệu phi cấu trúc như tin tức tài chính, báo cáo vĩ mô, và các thảo luận trên mạng xã hội. Việc áp dụng các kỹ thuật Xử lý ngôn ngữ tự nhiên (NLP) như Phân tích cảm xúc (Sentiment Analysis) hoặc Mô hình hóa chủ đề (Topic Modeling) có thể giúp mô hình "hiểu" được bối cảnh và nguyên nhân đằng sau các biến động giá, từ đó đưa ra các dự báo mang tính nhân quả hơn.
- **Khám phá các kiến trúc Attention tiên tiến:** Transformer và biến thể: Trong khi các kiến trúc dựa trên RNN (LSTM/GRU) xử lý dữ liệu một cách tuần tự, các mô hình Transformer đã chứng minh được khả năng vượt trội trong việc nắm bắt các mối quan hệ phức tạp và xa trong chuỗi nhờ cơ chế tự chú ý (self-attention). Việc nghiên cứu và áp dụng các kiến trúc dựa trên Transformer cho dữ liệu tài chính có thể giúp mô hình phát hiện các mẫu hình phi tuần tự mà RNN có thể bỏ lỡ, hứa hẹn mang lại một bước đột phá về hiệu suất.
- **Tối ưu hóa siêu tham số và kiến trúc tự động (AutoML/-NAS):** Việc thiết kế kiến trúc và lựa chọn siêu tham số trong đồ án này được thực hiện thủ công, dựa trên kinh nghiệm và các nghiên cứu

đã có. Hướng đi tiếp theo là áp dụng các phương pháp Học máy tự động (AutoML) và Tìm kiếm kiến trúc Nơ-ron (Neural Architecture Search - NAS). Các kỹ thuật này cho phép tự động khám phá một không gian rộng lớn các kiến trúc và bộ tham số tiềm năng, nhằm tìm ra một mô hình tối ưu về cả độ chính xác và hiệu quả tính toán cho một tập dữ liệu cụ thể.

- **Hướng tới hệ thống hỗ trợ giao dịch thời gian thực và quản lý rủi ro nâng cao:** Việc triển khai các mô hình trong một môi trường giao dịch thời gian thực là thách thức cuối cùng để kiểm chứng giá trị thực tiễn. Hướng phát triển này không chỉ dừng lại ở việc dự báo mà còn bao gồm các bài toán phức tạp hơn như: tối ưu hóa độ trễ (latency), xử lý sự trôi dạt của mô hình (model drift) khi thị trường thay đổi, và tích hợp các tín hiệu dự báo vào các hệ thống quản lý danh mục đầu tư (portfolio management) hoặc tối ưu hóa rủi ro một cách tự động.

Tài liệu tham khảo

- [1] D. Bai, G. Lu, Z. Zhu, J. Tang, J. Fang, and A. Wen, “Using time series analysis and dual-stage attention-based recurrent neural network to predict landslide displacement,” *Environmental Earth Sciences*, vol. 81, p. 5, Oct. 2022. DOI: 10.1007/s12665-022-10637-w.
- [2] H. Cao, S. Rajan, B. Hahn, *et al.*, “Mtlcomb: Multi-task learning combining regression and classification tasks for joint feature selection,” 2024. DOI: 10.48550/ARXIV.2405.09886. [Online]. Available: <https://arxiv.org/abs/2405.09886>.
- [3] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, no. 1, pp. 41–45, 1997. DOI: 10.1023/A:1007379606734.
- [4] K. Chhapariya, A. Benoit, K. M. Buddhiraju, and A. Kumar, “A multitask deep learning model for classification and regression of hyperspectral images: Application to the large-scale dataset,” Jul. 2024. DOI: 10.48550/arXiv.2407.16384.
- [5] M. Faraji, S. Nadi, O. Ghaffarpasand, SaHomayoni, and K. Downey, “An integrated 3d cnn-gru deep learning method for short-term prediction of pm2.5 concentration in urban environment,” *Science of The Total Environment*, vol. 834, p. 155324, 2022, ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2022.155324.
- [6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [7] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735.
- [8] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-excitation networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020, Epub 2019 Apr 29.

PMID: 31034408, ISSN: 0162-8828. DOI: 10 . 1109 / TPAMI . 2019 . 2913372. [Online]. Available: <https://doi.org/10.1109/TPAMI.2019.2913372>.

- [9] X. Li, “Cnn-gru model based on attention mechanism for large-scale energy storage optimization in smart grid,” *Frontiers in Energy Research*, vol. 11, Jul. 2023. DOI: 10.3389/fenrg.2023.1228256.
- [10] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, ijcai.org, 2017, pp. 2627–2633. DOI: 10.24963/ijcai.2017/366.
- [11] S. Ruder, “An overview of multi-task learning in deep neural networks,” Jun. 2017. DOI: 10.48550/arXiv.1706.05098.
- [12] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017, pp. 1–5. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fb053c1ca43747.pdf.
- [13] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. 2020, Bản dịch tiếng Việt, truy cập ngày 19 tháng 7 năm 2025. <https://vi.d2l.ai/>.