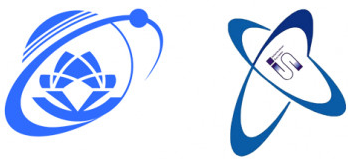
**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

**--------------&&&&&-------------**

**BÁO CÁO CUỐI KỲ**

Môn học: PHÂN TÍCH DỮ LIỆU KINH DOANH

**ĐỀ TÀI**

**SỬ DỤNG KỸ THUẬT PHÂN TÍCH   
CHUỖI THỜI GIAN VÀO**

**BÀI TOÁN DỰ ĐOÁN GIÁ CỔ PHIẾU CỦA CÁC CÔNG TY CÔNG NGHỆ**

Giảng viên hướng dẫn: TS. Trần Văn Hải Triều

Sinh viên thực hiện**: Nhóm 9**

1. **Hoàng Lê Nhật Minh MSSV: 23520926**
2. **Trịnh Thái Long MSSV: 23520891**
3. **Đinh Nhật Luân MSSV: 23520894**
4. **Cam Hồng Mạnh MSSV: 23520914**
5. **Thượng Trung Nguyên MSSV: 23521058**
6. **Chu Thị Cẩm Tú MSSV: 23521697**

# NHẬN XÉT CỦA GIÁO VIÊN

# 

# LỜI CẢM ƠN

Chúng tôi xin bày tỏ lòng biết ơn sâu sắc đến thầy Trần Văn Hải Triều vì sự hướng dẫn, định hướng và đóng góp quý báu trong quá trình thực hiện bài báo cáo này. Sự chỉ dẫn chuyên nghiệp và kiến thức sâu rộng của thầy Hải Triều đã mang đến cho chúng tôi những cơ sở lý thuyết và phương pháp nghiên cứu quan trọng, giúp chúng tôi tiếp cận và hiểu sâu hơn về kỹ thuật phân tích chuỗi thời gian và dự đoán giá cổ phiếu. Sự tận tâm và sự hỗ trợ từ thầy là một phần không thể thiếu trong thành công của chúng tôi.

Bên cạnh đó, chúng tôi cũng muốn bày tỏ lòng biết ơn đến các thành viên trong nhóm nghiên cứu. Sự đồng lòng, tương trợ và chia sẻ kiến thức giữa chúng tôi đã tạo ra một môi trường làm việc tích cực và động lực để vượt qua các thách thức trong quá trình nghiên cứu. Những ý kiến đóng góp và thảo luận trong nhóm đã giúp chúng tôi mở rộng tầm nhìn và hoàn thiện nội dung bài báo.

Cuối cùng, chúng tôi xin gửi lời cảm ơn tới tất cả những người đã đóng góp và hỗ trợ chúng tôi trong hành trình này. Những đóng góp và hỗ trợ này đã góp phần quan trọng vào thành công của bài báo và mang lại giá trị cho cả lĩnh vực nghiên cứu về dự đoán giá cổ phiếu. Chúng tôi hy vọng rằng công trình này sẽ được lan tỏa và tiếp tục khám phá những tiềm năng và ứng dụng mới trong lĩnh vực này.

Tp. Hồ Chí Minh, ngày 7 tháng 11 năm 2025

Nhóm sinh viên thực hiện

Nhóm 9

# MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN 1](#_Toc213450347)

[LỜI CẢM ƠN 2](#_Toc213450348)

[MỤC LỤC 3](#_Toc213450349)

[DANH MỤC BẢNG 7](#_Toc213450350)

[DANH MỤC HÌNH ẢNH 8](#_Toc213450351)

[DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ 12](#_Toc213450352)

[Chương 1: GIỚI THIỆU 15](#_Toc213450353)

[1.1. Tổng quan đề tài. 15](#_Toc213450354)

[1.2. Mục tiêu đề tài 15](#_Toc213450355)

[1.3. Cấu trúc bài báo cáo 16](#_Toc213450356)

[Chương 2: CÁC NGHIÊN CỨU LIÊN QUAN 18](#_Toc213450357)

[2.1. Linear Regression 18](#_Toc213450358)

[2.2. Gate Recurrent Unit (GRU) 18](#_Toc213450359)

[2.3. Autoregressive Integrated Moving Average (ARIMA) 19](#_Toc213450360)

[2.4. Exponential Smoothing (ETS) 20](#_Toc213450361)

[2.5. Long Short-Term Memory (LSTM) 21](#_Toc213450362)

[2.6. Recurrent Neural Network (RNN) 22](#_Toc213450363)

[2.7. Gradient Boosted Tree (GBT) 23](#_Toc213450364)

[2.8. Hidden Markov Model (HMM) 24](#_Toc213450365)

[2.9. Deep Neural Network (DNN) 24](#_Toc213450366)

[Chương 3: TÀI NGUYÊN 26](#_Toc213450367)

[3.1. Bộ dữ liệu 26](#_Toc213450368)

[3.1.1. Tập dữ liệu chứng khoán của Apple Inc. 27](#_Toc213450369)

[3.1.2. Tập dữ liệu chứng khoán của Samsung Electronics Co., Ltd 29](#_Toc213450370)

[3.1.3. Tập dữ liệu chứng khoán của Xiaomi Corporation 32](#_Toc213450371)

[3.2. Công cụ sử dụng 35](#_Toc213450372)

[3.2.1. IDE 35](#_Toc213450373)

[3.2.2. Các nền tảng hỗ trợ cộng tác 36](#_Toc213450374)

[Chương 4: PHƯƠNG PHÁP 37](#_Toc213450375)

[4.1. Linear Regression 37](#_Toc213450376)

[4.1.1 Giới thiệu 37](#_Toc213450377)

[4.1.2 Ý tưởng của mô hình 37](#_Toc213450378)

[4.1.3 Hình thành mô hình trong dự đoán 38](#_Toc213450379)

[4.2. Gate Recurrent Unit (GRU) 39](#_Toc213450380)

[4.2.1 Giới thiệu chung 39](#_Toc213450381)

[4.2.2 Cấu trúc và cơ chế hoạt động 39](#_Toc213450382)

[4.2.3 Ưu và nhược điểm của GRU 41](#_Toc213450383)

[4.3 Autoregressive Integrated Moving Average (ARIMA) 41](#_Toc213450384)

[4.3.1. Giới thiệu 41](#_Toc213450385)

[4.3.2 Cấu trúc và công thức mô hình 42](#_Toc213450386)

[4.3.3 Ứng dụng trong nghiên cứu 43](#_Toc213450387)

[4.4. Exponential Smoothing (ETS) 43](#_Toc213450388)

[4.4.1. Giới thiệu 43](#_Toc213450389)

[4.4.2. Cơ sở lý thuyết và công thức mô hình 44](#_Toc213450390)

[4.4.3. Ứng dụng trong nghiên cứu 45](#_Toc213450391)

[4.4.4. Ưu điểm và nhược điểm 46](#_Toc213450392)

[4.5. Long Short-Term Memory (LSTM) 47](#_Toc213450393)

[4.5.1. Giới thiệu 47](#_Toc213450394)

[4.5.2 Công thức tính toán các cổng trong LSTM: 47](#_Toc213450395)

[4.5.3. Ý tưởng của mô hình 48](#_Toc213450396)

[4.5.4. Hình thành mô hình trong dự đoán 51](#_Toc213450397)

[4.6. Recurrent Neural Network (RNN) 52](#_Toc213450398)

[4.6.1. Giới thiệu 52](#_Toc213450399)

[4.6.2. Ý tưởng thực hiện mô hình 54](#_Toc213450400)

[4.6.3. Xây dựng mô hình RNN trong dự đoán 55](#_Toc213450401)

[4.7. Gradient Boosted Tree (GBT) 55](#_Toc213450402)

[4.7.1. Giới thiệu 55](#_Toc213450403)

[4.7.3. Cách hoạt động của Grandient Boosted Trees: 56](#_Toc213450404)

[4.8. Hidden Markov Model (HMM) 58](#_Toc213450405)

[4.8.1. Giới thiệu 58](#_Toc213450406)

[4.8.2. Ý tưởng thực hiện mô hình 59](#_Toc213450407)

[4.8.3. Xây dựng mô hình HMM trong dự đoán 60](#_Toc213450408)

[4.9. Deep Neural Network (DNN) 61](#_Toc213450409)

[4.9.1. Giới thiệu 61](#_Toc213450410)

[4.9.2. Diễn giải kiến trúc 62](#_Toc213450411)

[4.9.3. Xây dựng mô hình DNN trong dự đoán giá cổ phiếu 63](#_Toc213450412)

[4.10. Các độ đo đánh giá mô hình 63](#_Toc213450413)

[4.10.1. Sai số phần trăm tuyệt đối trung bình (MAPE – Mean Absolute Percentage Error) 64](#_Toc213450414)

[4.10.2. Căn bậc hai của sai số bình phương trung bình (RMSE – Root Mean Squared Error) 64](#_Toc213450415)

[4.10.3. Sai số logarit bình phương trung bình (MSLE – Mean Squared Logarithmic Error) 65](#_Toc213450416)

[Chương 5: KẾT QUẢ THỰC NGHIỆM 66](#_Toc213450417)

[5.1. Linear Regression 66](#_Toc213450418)

[5.1.1. Tỉ lệ 7:1:2 66](#_Toc213450419)

[5.1.2. Tỉ lệ 6:2:2 67](#_Toc213450420)

[5.2. Gate Recurrent Unit (GRU) 69](#_Toc213450421)

[5.3.1. Tỉ lệ 7:1:2 69](#_Toc213450422)

[5.3.2. Tỉ lệ 6:2:2 70](#_Toc213450423)

[5.3. Autoregressive Integrated Moving Average (ARIMA) 72](#_Toc213450424)

[5.3.1. Tỉ lệ 7:1:2 72](#_Toc213450425)

[5.3.2. Tỉ lệ 6:2:2 73](#_Toc213450426)

[5.4. Exponential Smoothing (ETS) 74](#_Toc213450427)

[5.4.1. Tỉ lệ 7:1:2 74](#_Toc213450428)

[5.4.2. Tỉ lệ 6:2:2 76](#_Toc213450429)

[5.5. Long Short-Term Memory (LSTM) 77](#_Toc213450430)

[5.5.1. Tỉ lệ 7:1:2 77](#_Toc213450431)

[5.5.2. Tỉ lệ 6:2:2 78](#_Toc213450432)

[5.6. Recurrent Neural Network (RNN) 79](#_Toc213450433)

[5.6.1. Tỉ lệ 7:1:2 79](#_Toc213450434)

[5.6.2. Tỉ lệ 6:2:2 80](#_Toc213450435)

[5.7. Gradient Boosted Tree (GBT) 82](#_Toc213450436)

[5.7.1. Tỉ lệ 7:1:2 82](#_Toc213450437)

[5.7.2. Tỉ lệ 6:2:2 83](#_Toc213450438)

[5.8. Hidden Markov Model (HMM) 84](#_Toc213450439)

[5.8.1. Tỉ lệ 7:1:2 84](#_Toc213450440)

[5.8.2. Tỉ lệ 6:2:2 85](#_Toc213450441)

[5.9. Deep Neural Network (DNN) 87](#_Toc213450442)

[5.9.1. Tỉ lệ 7:1:2 87](#_Toc213450443)

[5.9.2. Tỉ lệ 6:2:2 87](#_Toc213450444)

[5.10. Đánh giá thuật toán 89](#_Toc213450445)

[5.10.1. Đánh giá trên tập dữ liệu Validate 89](#_Toc213450446)

[5.10.2. Đánh giá trên tập dữ liệu Test 93](#_Toc213450447)

[5.11. Dự đoán giá cổ phiếu trong 30 ngày tiếp theo 99](#_Toc213450448)

[Chương 6: KẾT LUẬN 107](#_Toc213450449)

[PHÂN CÔNG CÔNG VIỆC 108](#_Toc213450450)

[TÀI LIỆU THAM KHẢO 109](#_Toc213450451)

# DANH MỤC BẢNG

[Bảng 3.1. Mô tả chung các cột thuộc tính của 3 bộ dữ liệu 25](#_mwfivpc7m1pg)

[Bảng 3.2. Các chỉ số thống kê mô tả của cột Close trong tập dữ liệu Apple 26](#_z2plvkpv2nma)

[Bảng 3.3. Thống kê các thông số của cột Close trong tập dữ liệu Samsung 29](#_udju91azugxk)

[Bảng 3.4. Thống kê các thông số của cột Close trong tập dữ liệu Xiaomi 32](#_7wiapdvgpugs)

[Bảng 5.1. Đánh giá các mô hình trên tập dữ liệu Validate 90](#_nwq6evjsept6)

[Bảng 5.2. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Apple trên tập Validate 91](#_u0ptmsskqigv)

[Bảng 5.3. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Samsung trên tập Validate 92](#_ho59kl2k5vxi)

[Bảng 5.4. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Xiaomi trên tập Validate 92](#_i1r0vpsexvef)

[Bảng 5.5. Đánh giá các mô hình trên tập dữ liệu Test 94](#_dyw4ilk12us4)

[Bảng 5.6. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Apple trên tập Test 94](#_dcnyx0nfl1gq)

[Bảng 5.7. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Samsung trên tập Test 95](#_xdyv2g8fs0eu)

[Bảng 5.8. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Xiaomi trên tập Test 96](#_iif1b9hxsee3)

# DANH MỤC HÌNH ẢNH

[Hình 3.1. Biểu đồ chuỗi thời gian của cột Close cho tập dữ liệu Apple 23](#_z24td2fcgt5z)

[Hình 3.2. Biểu đồ Histogram của cột Close cho tập dữ liệu Apple 24](#_z24td2fcgt5z)

[Hình 3.3. Biểu đồ Boxplot của cột Close cho tập dữ liệu Apple 25](#_2heq538vftp5)

[Hình 3.4. Biểu đồ chuỗi thời gian của cột Close cho tập dữ liệu Samsung 26](#_940lwwtitr1w)

[Hình 3.5. Biểu đồ Histogram của cột Close tập dữ liệu Samsung 27](#_940lwwtitr1w)

[Hình 3.6. Biểu đồ Boxplot của cột Close tập dữ liệu Samsung 28](#_4z7vbdyilam8)

[Hình 3.7. Biểu đồ chuỗi thời gian của cột Close tập dữ liệu Xiaomi 29](#_w10eettrmiex)

[Hình 3.8. Biểu đồ Histogram của cột Close tập dữ liệu Xiaomi 30](#_w10eettrmiex)

[Hình 3.9. Biểu đồ Boxplot của cột Close tập dữ liệu Xiaomi 31](#_kfnkcqgm4tj4)

[Hình 4.1. Kiến trúc GRU 35](#_kfnkcqgm4tj4)

[Hình 4.2. Kiến trúc của LSTM 44](#_9vrvfdk4xsdk)

[Hình 4.3. Hình ảnh Cell State 45](#_d6akydemobwg)

[Hình 4.4. Hình ảnh Forget Gate 46](#_tbb41axzruij)

[Hình 4.5. Hình ảnh về bước tiếp theo 47](#_ii8ny8wjq6kx)

[Hình 4.6. Hình ảnh về việc tính toán lại Cell State 47](#_c5462fclrh38)

[Hình 4.7. Hình ảnh Output Gate 48](#_kye9knr6c9fm)

[Hình 4.8. 3 thành phần chính của RNN 49](#_nnaps13mt5vq)

[Hình 4.9. Kiến trúc RNN 50](#_oow5givwou0q)

[Hình 5.1. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình Linear Regression tỷ lệ 7:1:2 64](#_yv7ujd2rh071)

[Hình 5.2. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình Linear Regression tỷ lệ 7:1:2 65](#_4dudcsufael)

[Hình 5.3. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình Linear Regression tỷ lệ 7:1:2 65](#_xon8ad1w6gij)

[Hình 5.4. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình Linear Regression tỷ lệ 6:2:2. 66](#_gu2592w73atm)

[Hình 5.5. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình Linear Regression tỷ lệ 6:2:2 66](#_r9mv7g3ub0d)

[Hình 5.6. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình Linear Regression tỷ lệ 6:2:2 66](#_gjvajsh6dejn)

[Hình 5.7. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 7:1:2 67](#_ozjcgw5myfzd)

[Hình 5.8. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 7:1:2 67](#_pwgdfqqu5mi1)

[Hình 5.9. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 7:1:2 68](#_mcn0wjys5ij)

[Hình 5.10. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 6:2:2 68](#_j0n9d5fnvbqu)

[Hình 5.11. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 6:2:2 69](#_wwkf0fbcndom)

[Hình 5.12. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 6:2:2 69](#_fmwrnyqsgux4)

[Hình 5.13. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình ARIMA tỷ lệ 7:1:2 70](#_1hjdokolxyy2)

[Hình 5.14. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình ARIMA tỷ lệ 7:1:2 70](#_8njqep6dohb5)

[Hình 5.15. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình ARIMA tỷ lệ 7:1:2 71](#_8npf1st7tnav)

[Hình 5.16. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình ARIMA tỷ lệ 6:2:2 71](#_pulgriuon0rs)

[Hình 5.17. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình ARIMA tỷ lệ 6:2:2 72](#_kk47frvbvvj8)

[Hình 5.18. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình ARIMA tỷ lệ 6:2:2 72](#_wbuxnkss84tu)

[Hình 5.19. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình ETS tỷ lệ 7:1:2 73](#_5z5hmcmjtbp9)

[Hình 5.20. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình ETS tỷ lệ 7:1:2 73](#_1kqq9vvfrfvn)

[Hình 5.21. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình ETS tỷ lệ 7:1:2 74](#_klljhfmvw7bd)

[Hình 5.22. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình ETS tỷ lệ 6:2:2 74](#_54e7ow86cmgm)

[Hình 5.23. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình ETS tỷ lệ 6:2:2 74](#_2ckizvbnh1pc)

[Hình 5.24. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình ETS tỷ lệ 6:2:2 75](#_up7uh9gmgf5z)

[Hình 5.25. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình LSTM tỷ lệ 7:1:2 75](#_t4v2o09wmcao)

[Hình 5.26. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình LSTM tỷ lệ 7:1:2 75](#_9szhmiyjpodc)

[Hình 5.27. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình LSTM tỷ lệ 7:1:2 76](#_cb0xqtvkmlbx)

[Hình 5.28. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình LSTM tỷ lệ 6:2:2 76](#_rgh9gta8a3vd)

[Hình 5.29. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình LSTM tỷ lệ 6:2:2 76](#_4wa12ilzeded)

[Hình 5.30. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình LSTM tỷ lệ 6:2:2 77](#_koluk4gsuft)

[Hình 5.31. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình RNN tỷ lệ 7:1:2 77](#_bzc5mskfamcj)

[Hình 5.32. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình RNN tỷ lệ 7:1:2 78](#_vome312ho3w5)

[Hình 5.33. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình RNN tỷ lệ 7:1:2 78](#_3f1xidoabs89)

[Hình 5.34. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình RNN tỷ lệ 6:2:2 79](#_yqdrwnd3dc9w)

[Hình 5.35. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình RNN tỷ lệ 6:2:2 79](#_qfmcv7s2pq6m)

[Hình 5.36. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình RNN tỷ lệ 6:2:2 80](#_fjj0ojl8gyoi)

[Hình 5.37. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình GBT tỷ lệ 7:1:2 80](#_o3ecqkxg6au8)

[Hình 5.38. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình GBT tỷ lệ 7:1:2 81](#_egn8xd3a2t69)

[Hình 5.39. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình GBT tỷ lệ 7:1:2 81](#_a6mquktzuoym)

[Hình 5.40. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình GBT tỷ lệ 6:2:2 81](#_d2c7bow909n4)

[Hình 5.41. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình GBT tỷ lệ 6:2:2 81](#_6zocxzm16i16)

[Hình 5.42. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình GBT tỷ lệ 6:2:2 82](#_1yvmjw8ch1mj)

[Hình 5.43.Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình HMM tỷ lệ 7:1:2 82](#_xq6gojcvyxr9)

[Hình 5.44. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình HMM tỷ lệ 7:1:2 83](#_q684zuwfuw14)

[Hình 5.45. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình HMM tỷ lệ 7:1:2 83](#_mcfcxp9rtqaj)

[Hình 5.46. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình HMM tỷ lệ 6:2:2 83](#_ptqfu550w5vc)

[Hình 5.47. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình HMM tỷ lệ 6:2:2 84](#_9c1srt2igz0t)

[Hình 5.48. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình HMM tỷ lệ 6:2:2 84](#_u8c069g70ld6)

[Hình 5.49. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình DNN tỷ lệ 7:1:2 85](#_tmal1z84n8tf)

[Hình 5.50. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình DNN tỷ lệ 7:1:2 85](#_am3psif9yfb5)

[Hình 5.51. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình DNN tỷ lệ 7:1:2 85](#_pcrpu7soke1k)

[Hình 5.52. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình DNN tỷ lệ 6:2:2 86](#_vbnuxqq4gewc)

[Hình 5.53. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình DNN tỷ lệ 6:2:2 86](#_oqba8795jzmw)

[Hình 5.54. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình DNN tỷ lệ 6:2:2 86](#_e0sq06bsjen3)

[Hình 5.55. Dự đoán giá cổ phiếu trong 30 ngày tiếp theo với tỉ lệ 6:2:2 94](#_y926azrqjxiq)

[Hình 5.56. Dự đoán 30 ngày của Apple trên tỷ lệ 6:2:2 95](#_q6cfuzgst3xb)

[Hình 5.57. Dự đoán 30 ngày của Samsung trên tỷ lệ 6:2:2 96](#_eds6jen97zbz)

[Hình 5.58. Dự đoán 30 ngày của Xiaomi trên tỷ lệ 6:2:2 97](#_7m9pumoom8pf)

[Hình 5.59. Dự đoán giá cổ phiếu trong 30 ngày tiếp theo với tỉ lệ 7:1:2 98](#_py2nnqubw28z)

[Hình 5.60. Dự đoán của Apple trên tỷ lệ 7:1:2 99](#_msm5mbvexdz)

[Hình 5.61. Dự đoán 30 ngày của Samsung trên tỷ lệ 7:1:2 100](#_lcl6rymvd4d3)

[Hình 5.62. Dự đoán 30 ngày của Xiaomi trên tỷ lệ 7:1:2 101](#_cb9t71vjhuiu)

# DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Tên đầy đủ** | **Giải thích** |
| 1 | DFNN | Deep Feedforward Neural Network | Deep Feedforward Neural Network (DFNN) là mô hình học sâu điển hình để học các biểu diễn dữ liệu phức tạp và trừu tượng theo cấp bậc. |
| 2 | ETS | Exponential Smoothing | Một phương pháp dự báo trong phân tích chuỗi thời gian, nơi các chuỗi thời gian quan sát được gán trọng số không đều. |
| 3 | LSTM | Long Short-Term Memory | Một loại mạng nơ-ron học sâu hay còn được biến đến là một loại đặc biệt của mạng Recurrent Neural Network |
| 4 | RNN | Recurrent Neural Network | Là một kiến trúc mạng nơ-ron đặc biệt có khả năng lưu trữ các trạng thái trước đó thông qua việc sử dụng một vòng lặp trong kiến trúc của nó. |
| 5 | HMM | Hidden Markov Model | Là một mô hình thống kê dựa trên lý thuyết xác suất và quá trình Markov ẩn, trong đó tồn tại hai thành phần chính là trạng thái ẩn và trạng thái có thể quan sát được. |
| 6 | GRU | Gated Recurrent Unit | Là một loại kiến trúc mạng nơ-ron hồi quy, bao gồm hai cổng chính là cổng cập nhật (update gate) và cổng đặt lại (reset gate). |
| 7 | GBT | Gradient Boosted Tree | Là một phương pháp học máy trong đó nhiều mô hình dự đoán nhỏ (thường là cây quyết định) được kết hợp để tạo thành một mô hình dự đoán mạnh mẽ hơn. |
| 8 | MAE | Mean Absolute Error | Đo lường sự khác biệt trung bình giữa các giá trị dự đoán và giá trị thực tế, không quan tâm đến hướng của sai số. |
| 9 | MAPE | Mean Absolute Percentage Error | Đo lường tỷ lệ phần trăm trung bình của sai số giữa các giá trị dự đoán và giá trị thực tế, giúp đánh giá sự chính xác của mô hình dự báo. |
| 10 | MSE | Mean Squared Error | Đo lường sự khác biệt bình phương trung bình giữa các giá trị dự đoán và giá trị thực tế, tăng cường tác động của các sai số lớn hơn. |
| 11 | RMSE | Root Mean Squared Error | Đo lường sự khác biệt trung bình của các sai số hoặc lỗi dự đoán, có đơn vị tương tự với biến phụ thuộc. |
| 12 | MSLE | Mean Squared Logarithmic Error | Đo lường sự khác biệt bình phương trung bình của các sai số logarithmic giữa các giá trị dự đoán và giá trị thực tế, thích hợp cho các biến dự đoán có phạm vi giá trị rộng và tính chất logarithmic. |
| 13 | RNN | Recurrent Neural Network | Là một kiến trúc mạng nơ-ron đặc biệt có khả năng lưu trữ các trạng thái trước đó thông qua việc sử dụng một vòng lặp trong kiến trúc của nó. |
| 14 | DNN | Deep Neural Network | Mạng neural nhiều lớp (input, hidden, output) sử dụng các dense layer để học mối quan hệ phi tuyến giữa các đặc trưng. |
| 15 | MAP | Maximum a Posteriori | Phương pháp tối đa hóa xác suất hậu nghiệm trong thống kê/Bayes. |

**GIẢI THÍCH THUẬT NGỮ**

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ** | **Giải thích** |
| 1 | Dense layer | Dense layer là một lớp trong mạng nơ-ron nhân tạo, còn được gọi là fully-connected layer. Trong lớp này, mỗi neuron nhận đầu vào từ tất cả các neuron của lớp trước đó. |
| 2 | Loss Function | Loss function là một phần quan trọng trong quá trình huấn luyện mô hình máy học. Đo lường mức độ chênh lệch giữa dự đoán của mô hình và giá trị thực tế của dữ liệu đầu ra. Mục tiêu là tìm một hàm mất mát mà khi càng nhỏ, thì mô hình sẽ càng chính xác trong việc dự đoán. |
| 3 | Activation Function | Activation functions là những hàm phi tuyến được áp dụng vào đầu ra của các nơ-ron trong tầng ẩn của một mô hình mạng, và được sử dụng làm input data cho tầng tiếp theo. |
| 4 | Gradient Descent | Là một thuật toán tối ưu hóa được sử dụng để cập nhật các tham số của mạng nơ-ron dựa trên đạo hàm của hàm mất mát. Mục tiêu của Gradient Descent là tìm giá trị cực tiểu của hàm mất mát bằng cách điều chỉnh các tham số mô hình theo hướng giảm dần độ dốc của hàm mất mát. |
| 5 | Gradient | Trong Gradient Boosted Trees, gradient liên quan đến khái niệm "gradient descent", một thuật toán được sử dụng rộng rãi trong học máy để tối ưu hóa các hàm mất mát. |
| 6 | Boosting | Là một thuật toán học máy kết hợp nhiều mô hình yếu hoặc cơ bản để tạo ra một mô hình dự đoán mạnh mẽ |
| 7 | Sigmoid | Là một hàm toán học được sử dụng rộng rãi trong học máy và mạng nơ-ron nhân tạo, đây là một hàm phi tuyến mà biến đổi một số thực bất kỳ thành một giá trị nằm trong khoảng từ 0 đến 1. |
| 8 | Tanh | Là một hàm kích hoạt được sử dụng trong các mô hình mạng nơ-ron nhân tạo như LSTM và GRU. Đầu ra từ hàm tanh có thể dương hoặc âm, cho phép tăng hoặc giảm trạng thái. |
| 9 | Bias | Là độ chệch thể hiện sự khác biệt giữa giá trị thực tế và giá trị dự đoán trung bình của mô hình. |
| 10 | Adam | Phương pháp tối ưu hóa tổ hợp giữa Momentum và RMSprop. Hoạt động bằng cách tích lũy từ gradient trước đó và điều chỉnh tỷ lệ học động cho từng tham số. |
| 11 | Receptive field | Vùng không gian mà một nơ-ron hoặc một tập hợp các nơ-ron trong mạng có thể “nhìn thấy” và tương tác trực tiếp. Receptive field xác định phạm vi cục bộ của thông tin được xử lý bởi mỗi nơ-ron. |
| 12 | Hyperparameter | Các tham số không được mô hình học tự động từ dữ liệu, mà phải được thiết lập trước, ảnh hưởng đến cấu trúc và quá trình tối ưu hóa của mô hình học máy hoặc mạng nơ-ron. |
| 13 | Backpropagation Through Time (BPTT) | Lan truyền ngược trong RNN, cập nhật trọng số dựa trên lỗi qua các bước thời gian. |
| 14 | Viterbi algorithm | Thuật toán tìm chuỗi trạng thái ẩn có xác suất lớn nhất trong HMM. |
| 15 | fracChange / fracHigh / fracLow | Giá trị quan sát hàng ngày trong HMM, thể hiện sự thay đổi phân đoạn của giá cổ phiếu. |

# GIỚI THIỆU

## 1.1. Tổng quan đề tài.

Trong lĩnh vực tài chính hiện đại, cổ phiếu là một trong những kênh đầu tư trọng yếu, đại diện cho quyền sở hữu một phần vốn trong doanh nghiệp phát hành. Việc định giá cổ phiếu đóng vai trò quan trọng trong cả hoạt động đầu tư lẫn quản trị doanh nghiệp, bởi nó phản ánh giá trị thực của một công ty trên thị trường tại một thời điểm nhất định. Từ góc độ doanh nghiệp, định giá cổ phiếu là bước then chốt trong quá trình phát hành cổ phiếu, huy động vốn và mở rộng quy mô hoạt động. Trong khi đó, nhà đầu tư dựa vào kết quả định giá để nhận diện các cơ hội đầu tư tiềm năng, đánh giá mức độ rủi ro, cũng như xác định thời điểm mua hoặc bán cổ phiếu tối ưu nhằm tối đa hóa lợi nhuận.

Một cách tiếp cận phổ biến trong định giá cổ phiếu là phân tích và dự báo giá thị trường của cổ phiếu trong tương lai. Nếu giá thị trường thấp hơn giá trị nội tại, cổ phiếu có thể được xem là đang bị định giá thấp và đáng để đầu tư; ngược lại, nếu giá thị trường cao hơn giá trị nội tại, nhà đầu tư có thể xem xét bán ra để chốt lời. Do đó, việc xây dựng các mô hình dự báo chính xác giá cổ phiếu trở thành một trong những vấn đề cốt lõi của phân tích dữ liệu tài chính, giúp nhà đầu tư và doanh nghiệp ra quyết định dựa trên cơ sở định lượng.

Trên cơ sở đó, nghiên cứu này được thực hiện nhằm phân tích, huấn luyện và đánh giá hiệu quả của các mô hình dự báo chuỗi thời gian cổ điển và hiện đại trong bài toán dự báo giá cổ phiếu, qua đó xác định mô hình có hiệu suất tốt nhất cho dự báo ngắn hạn trên dữ liệu thực tế, từ đó đề xuất hướng ứng dụng phù hợp trong thực tiễn phân tích tài chính doanh nghiệp.

## 1.2. Mục tiêu đề tài

Nghiên cứu tập trung vào các mục tiêu sau:

* **Xử lý và phân tích** ba bộ dữ liệu dữ lịch sử về giá cổ phiếucủa ba tập đoàn công nghệ lớn gồm *Apple Inc. (Hoa Kỳ)*, *Samsung Electronics Co., Ltd. (Hàn Quốc)* và *Xiaomi Corporation (Trung Quốc)*.
* **Xây dựng và huấn luyện** các mô hình dự báo chuỗi thời gian trên hai tỷ lệ chia dữ liệu train - validation - test lần lượt là *7:1:2* và *6:2:2*. Các mô hình sử dụng trong nghiên cứu bao gồm: Linear Regression, Autoregressive Integrated Moving Average (ARIMA), Gated Recurrent Unit (GRU), Exponential Smoothing (ETS), Gradient Boosted Tree (GBT), Hidden Markov Model (HMM), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Deep Neural Network (DNN).
* **Đánh giá hiệu suất** dự báo của các mô hình dựa trên ba độ đo chính: Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE) và Mean Squared Logarithmic Error (MSLE).
* Lựa chọn hai mô hình có hiệu suất dự báo cao nhất để **dự báo** giá đóng cửa cổ phiếu trong 30 ngày tiếp theo, nhằm đánh giá khả năng ứng dụng của mô hình trong dự báo ngắn hạn và hỗ trợ ra quyết định đầu tư

## 1.3. Cấu trúc bài báo cáo

Cấu trúc các chương tiếp theo bài báo cáo được sắp xếp và tổ chức theo hướng xây dựng trên mục tiêu cụ thể, đảm bảo tính logic và nhất quán. Cụ thể như sau:

* **Chương 2:** Tổng hợp các công trình nghiên cứu có liên quan của các phương pháp và mô hình được sử dụng nhằm làm rõ lý do lựa chọn phương pháp và mô hình, đồng thời làm tiền đề cho mục tiêu **Xây dựng và huấn luyện** mô hình.
* **Chương 3:** Mô tả quy trình thu thập và tiền xử lý dữ liệu, đặc điểm của các bộ dữ liệu làm đầu vào cho mô hình, các thống kê mô tả sơ bộ phục vụ cho mục tiêu **Xử lý và phân tích** dữ liệu ban đầu.
* **Chương 4:** Trình bày cơ sở lý thuyết về phương pháp, mô hình nghiên cứu và các độ đo dùng để đánh giá, phục vụ cho mục tiêu **Xây dựng và huấn luyện** mô hình, đồng thời làm tiền đề cho mục tiêu **Đánh giá hiệu suất** dự báo của mô hình.
* **Chương 5:** Tập trung vào trình bày kết quả thực nghiệm, đánh giá hiệu suất mô hình và xác định hai mô hình có độ chính xác cao nhất, đồng thời dự báo giá cổ phiếu trong 30 ngày tiếp theo, đáp ứng được mục tiêu **Đánh giá hiệu suất** mô hình và **Dự báo** thực tiễn.
* **Chương 6:** Tổng kết, rút ra kết luận và đề xuất hướng phát triển trong tương lai.

# CÁC NGHIÊN CỨU LIÊN QUAN

Chương 2 sẽ trình bày các nghiên cứu liên quan đến 9 mô hình chúng tôi dùng tiến hành nghiên cứu và thực nghiệm.

## 2.1. Linear Regression

Nhóm tác giả Nguyễn Thanh Bình (Trường Đại học Kinh tế TP. Hồ Chí Minh) đã thực hiện nghiên cứu “Ảnh hưởng của các nhân tố vĩ mô đến lợi nhuận cổ phiếu: Trường hợp thị trường chứng khoán Việt Nam” bằng mô hình Linear Regression (OLS) [1].

Tập dữ liệu bao gồm dữ liệu tháng của các chỉ số vĩ mô (CPI, lãi suất, tỷ giá, VN-Index, GDP) trong giai đoạn 2015–2021.

Kết quả nghiên cứu cho thấy mô hình hồi quy tuyến tính đa biến đạt giá trị R² = 0.82, RMSE = 4.15, và MAPE = 2.47%, chứng minh rằng biến GDP và tỷ giá có ảnh hưởng đáng kể nhất đến lợi nhuận cổ phiếu tại Việt Nam.

## 2.2. Gate Recurrent Unit (GRU)

Chengyu Li và Guoqi Qian, thuộc Khoa Toán học và Thống kê của Đại học Melbourne (Úc), đã đề xuất mô hình *frequency decomposition–based GRU transformer (FDGRU‑Transformer)* trong bài báo *“Stock Price Prediction Using a Frequency Decomposition Based GRU Transformer Neural Network”*. Nghiên cứu tập trung giải quyết hai hạn chế lớn của các mô hình mạng nơ‑ron hồi quy (RNN): khả năng trích xuất đặc trưng từ dữ liệu cổ phiếu phức tạp còn yếu và việc mất thông tin theo thời gian khi xử lý chuỗi dài. Để khắc phục, mô hình FDGRU‑Transformer áp dụng phân rã tần số (qua phương pháp “complete ensemble empirical mode decomposition”) để tách dữ liệu thành các thành phần rõ ràng hơn, đồng thời kết hợp các kiến trúc GRU, LSTM và multi‐head attention của Transformer nhằm giữ lại thông tin thời gian và học được quan hệ dài hạn.Kết quả thực nghiệm cho thấy mô hình này đạt hiệu suất dự báo vượt trội so với các phương pháp tiên tiến, và phân tích loại bỏ (ablation study) đã xác nhận vai trò thiết yếu của từng thành phần trong kiến trúc [2].

Bên cạnh nghiên cứu này, nhiều công trình khác cũng khai thác các kỹ thuật phân rã tín hiệu hoặc kiến trúc học sâu tiên tiến để cải thiện dự báo giá cổ phiếu. Ví dụ, nghiên cứu “Multi‐feature stock price prediction by LSTM networks based on VMD–TMFG–LSTM” (Z. Zhang et al., 2025) sử dụng phân rã mẫu biến động (Variational Mode Decomposition) kết hợp TMFG và LSTM, đạt cải thiện đáng kể về RMSE và MAE [3]. Ngoài ra, nhiều nghiên cứu khác như ứng dụng biến đổi wavelet hoặc denoising trước khi đưa vào LSTM cũng cho thấy hiệu quả tăng trong dự báo chuỗi thời gian tài chính [4].

Tổng hợp lại, các công trình này cho thấy rằng trong lĩnh vực dự báo giá cổ phiếu, việc kết hợp phân rã tín hiệu, khai thác đặc trưng theo tần số, cùng với các mô hình học sâu hiện đại như GRU, LSTM, Transformer, đang là hướng đi chủ đạo và đem lại nhiều kết quả hứa hẹn.

## 2.3. Autoregressive Integrated Moving Average (ARIMA)

Nghiên cứu của tác giả Vaia I. Kontopoulou và các đồng nghiệp tổng hợp việc ứng dụng ARIMA trong nhiều lĩnh vực, đặc biệt là tài chính như dự báo giá vàng, dầu và Bitcoin. Các tác giả nhận định ARIMA hoạt động hiệu quả với dữ liệu tuyến tính, ổn định, khi chuỗi được xử lý dừng. Tuy nhiên, khi dữ liệu có tính phi tuyến hoặc chịu ảnh hưởng đa biến, mô hình học sâu tỏ ra vượt trội hơn [5].

Trong nghiên cứu của Asha Sunki và các đồng nghiệp, nhóm tác giả sử dụng bộ dữ liệu giá cổ phiếu Netflix gồm 4.882 quan sát và 7 biến. Kết quả cho thấy mô hình ARIMA đạt RMSE = 7.89, tốt hơn FB Prophet (9.12) và LSTM (10.34). Tác giả kết luận ARIMA phù hợp cho dữ liệu ổn định, tuyến tính, và cho hiệu quả cao trong dự báo ngắn hạn [6].

Trong nghiên cứu nhóm tác giả Nguyen Thi Thanh Loan, Dang Ngoc Hung, Vu Thi Thuy Van, họ lấy dữ liệu từ 5 mã cổ phiếu vốn hóa lớn tại Việt Nam trong giai đoạn 2018–2023. Các chỉ số đánh giá gồm RMSE, MAE, MAPE cho thấy ARIMA dự báo khá chính xác ở các mã có xu hướng ổn định (RMSE trung bình ≈ 1.25), trong khi LSTM vượt trội hơn khi dữ liệu biến động mạnh. Nhóm nghiên cứu đề xuất kết hợp hai mô hình để cải thiện độ chính xác [7].

Cuối cùng, nghiên cứu của nhóm tác giả Peng Xia, Jinping Feng và Tianhai Tian sử dụng dữ liệu của NASDAQ và S&P500 trong 10 năm, so sánh ARIMA với CNN-LSTM. ARIMA đạt RMSE = 12.4, trong khi CNN-LSTM đạt 10.8. Tác giả nhận xét ARIMA ổn định, ít yêu cầu tính toán, phù hợp cho dự báo ngắn hạn, còn mô hình học sâu hiệu quả hơn nhưng cần nhiều dữ liệu và công suất tính toán [8].

Các nghiên cứu trên đều khẳng định ARIMA vẫn là mô hình nền tảng cho dự báo chuỗi thời gian tài chính, hiệu quả cao với dữ liệu tuyến tính và ngắn hạn, nhưng hạn chế với dữ liệu phi tuyến hoặc biến động mạnh. Do đó, nhiều tác giả đề xuất kết hợp ARIMA với các mô hình học sâu để nâng cao độ chính xác dự báo.

## 2.4. Exponential Smoothing (ETS)

Phương pháp làm mịn mũ (ETS) bao gồm một nhóm các kỹ thuật dự báo chuỗi thời gian kinh điển, trong đó các quan sát gần đây được gán trọng số lớn hơn theo cấp số nhân. Các biến thể phổ biến gồm: Làm mịn mũ đơn giản (SES) – chỉ mô hình hóa mức độ (level), Làm mịn mũ kép (DES, hay phương pháp Holt) – mô hình hóa mức độ và xu hướng (trend), và Làm mịn mũ ba (phương pháp Holt-Winters) – bổ sung thêm khả năng xử lý thành phần mùa (seasonality).

Trong một nghiên cứu của nhóm tác giả Y. F. F. G. de S. E. L. M. de S. L. C. de S. M. da Silva và cộng sự về dự báo nhu cầu điện, phương pháp Holt-Winters đạt MAPE = 12,57% — vượt trội hơn đáng kể so với DES (17,02%) và SES (18,23%). Kết quả này phản ánh rằng bộ dữ liệu có chứa cả xu hướng và tính mùa rõ rệt; trong khi SES chỉ xử lý mức độ và DES chỉ xử lý mức độ cùng xu hướng, cả hai đều không thể nắm bắt đầy đủ yếu tố mùa. Phát hiện này nhấn mạnh tầm quan trọng của việc lựa chọn biến thể ETS phù hợp với cấu trúc của chuỗi PLOS ONE thời gian.

Một nghiên cứu khác so sánh trực tiếp giữa ETS và ARIMA cho thấy: ARIMA thường hoạt động tốt hơn với dữ liệu không có tính mùa rõ ràng (như giá cổ phiếu Google), trong khi ETS tỏ ra vượt trội khi dữ liệu có tính mùa mạnh, chẳng hạn như lượng khách du lịch đến Úc. Sở dĩ như vậy là vì ETS mô hình hóa rõ ràng các thành phần mức độ, xu hướng và mùa, trong khi ARIMA xử lý chúng gián tiếp thông qua quá trình sai phân [9].

Tầm quan trọng của ETS được củng cố qua các nghiên cứu quy mô lớn về dự báo chuỗi thời gian. Trong bài báo *“Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward”* của Spyros Makridakis, Evangelos Spiliotis và Vassilios Assimakopoulos (2018) [10], các tác giả đã chỉ ra rằng các phương pháp thống kê cổ điển như ETS (đặc biệt là Damped ETS) có hiệu suất dự báo đáng tin cậy và thường vượt trội hơn các mô hình học máy (ML) và học sâu (DL) như LSTM trong các bài toán dự báo một bước (one-step forecasting). Các tác giả nhấn mạnh rằng mô hình ETS không chỉ hiệu quả về tính toán mà còn có khả năng diễn giải rõ ràng, khiến nó trở thành “mốc chuẩn” (baseline) mà các mô hình phức tạp hơn cần vượt qua.

Kết quả này được củng cố thêm bởi nghiên cứu tiếp theo *“The M4 Competition: 100 000 time series and 61 forecasting methods”* của Makridakis et al. (2019) [11], trong đó các mô hình thống kê (đặc biệt là các biến thể ETS và phương pháp kết hợp) tiếp tục cho thấy hiệu suất vượt trội trong cuộc thi M4 – vượt hơn nhiều phương pháp học máy và học sâu về độ chính xác và tính ổn định.

Tổng hợp các kết quả trên cho thấy ETS là một phương pháp dự báo nền tảng, mạnh mẽ và linh hoạt, đặc biệt hiệu quả với các chuỗi thời gian có chứa xu hướng và tính mùa rõ ràng. Mặc dù các mô hình học sâu có thể nắm bắt các mẫu phi tuyến phức tạp hơn, ETS vẫn giữ vị thế quan trọng nhờ tính đơn giản, khả năng diễn giải và hiệu suất dự báo hàng đầu, thường được sử dụng như mô hình cơ sở hoặc tích hợp trong các kiến trúc lai (hybrid models) để nâng cao độ chính xác.

## 2.5. Long Short-Term Memory (LSTM)

Nghiên cứu của Seng Hansun và Julio Christian Young (2021) tập trung dự báo giá đóng cửa của các cổ phiếu thuộc chỉ số LQ45 – nhóm 45 công ty có thanh khoản cao trên thị trường chứng khoán Indonesia – bằng mạng nơ-ron LSTM. Họ lựa chọn 6 mã cổ phiếu trong lĩnh vực tài chính và xây dựng mô hình gồm ba lớp LSTM, kết hợp tiền xử lý dữ liệu như điền khuyết giá thiếu, chuẩn hóa và chuyển đổi dữ liệu thành dạng phù hợp với đầu vào 3‑chiều của LSTM. Sau quá trình huấn luyện, kết quả cho thấy mô hình đạt MAPE thấp nhất 18,61% với mã BMRI và 19,10% với mã BBCA, chứng minh khả năng dự báo tốt dù kiến trúc đơn giản. Nghiên cứu cũng chỉ ra một số hạn chế như phương pháp tiền xử lý còn đơn giản và sử dụng ít chỉ số đánh giá, đồng thời đề xuất các nghiên cứu sau nên áp dụng kỹ thuật tiền xử lý và đánh giá đa dạng hơn để nâng cao hiệu quả dự báo [12].

Nghiên cứu của Zeng Lingling và Li Junhong thực hiện phân tích và dự báo giá cổ phiếu của Ping An Bank trong khoảng thời gian từ 1 / 1 / 2015 đến 1 / 6 / 2024 (tổng cộng 2.287 điểm dữ liệu lịch sử) bằng các mô hình mạng nơ‑ron hồi tiếp (RNN) và mạng LSTM [13]. Họ so sánh hiệu quả giữa ba kiến trúc: LSTM đơn thuần, mô hình lai CNN‑LSTM và CNN‑BiLSTM, và kết quả cho thấy BiLSTM đạt sai số thấp nhất – tức là thể hiện độ chính xác cao hơn trong dự báo giá cổ phiếu so với các mô hình còn lại [13]. Nghiên cứu này nhấn mạnh rằng sử dụng mạng LSTM và các biến thể như BiLSTM có thể nâng cao hiệu quả trong dự báo chuỗi thời gian tài chính và khuyến nghị ứng dụng rộng hơn trong đầu tư và tối ưu hóa danh mục đầu tư.

Bài báo Stock Price Prediction Using ARIMA and LSTM Models: An Application to CSI 300 Closing Prices (Yifan Zhang, 2024) tập trung dự báo giá đóng cửa hằng ngày của chỉ số CSI 300 từ 1/7/2022 tới 26/7/2023 dựa trên 261 ngày dữ liệu gồm sáu đặc trưng trích từ trang dữ liệu chứng khoán Trung Quốc [14]. Nghiên cứu so sánh hiệu năng giữa mô hình thống kê truyền thống Autoregressive Integrated Moving Average (ARIMA) và mạng nơ‑ron hồi tiếp dài – ngắn hạn (LSTM). Kết quả cho thấy chuỗi đóng cửa của CSI 300 có tự tương quan và trí nhớ dài hạn, tạo điều kiện để dự báo từ dữ liệu lịch sử.Về mặt hiệu suất, LSTM thể hiện khả năng dự báo mạnh hơn so với ARIMA, nhưng nghiên cứu lưu ý rằng mô hình LSTM có độ trễ – tức là phản ứng chậm với các biến động mạnh, điều này hạn chế khi dự báo những thay đổi đột ngột trong ngắn hạn.Tác giả kết luận rằng để có kết quả chính xác hơn, cần bổ sung thêm thông tin đặc trưng (ví dụ: biến ngoài giá đóng cửa, khối lượng giao dịch, chỉ số kỹ thuật) cho mô hình dự báo.

## 2.6. Recurrent Neural Network (RNN)

Mạng Nơ-ron Hồi quy (RNN) là một lớp mạng nơ-ron nhân tạo được thiết kế đặc biệt để xử lý dữ liệu dạng chuỗi (sequential data). Khác với các mạng truyền thẳng, RNN sở hữu các kết nối "hồi quy", cho phép thông tin (dưới dạng một "trạng thái ẩn") được lưu trữ và truyền qua các bước thời gian. Kiến trúc này tạo ra một dạng "bộ nhớ", giúp mô hình có khả năng nắm bắt các phụ thuộc động và các mẫu hình theo thời gian, khiến nó trở thành một công cụ mạnh mẽ cho dự báo chuỗi thời gian tài chính.

Trong một nghiên cứu của Yongqiong Zhu[15], tác giả đã sử dụng mô hình RNN để dự đoán giá cổ phiếu Apple (AAPL) sử dụng dữ liệu 10 năm. Tác giả xây dựng mô hình RNN hai lớp (lớp đầu 50 nút, lớp sau 100 nút), sử dụng hàm tối ưu Adam và hàm mất mát MSE. Nghiên cứu báo cáo mô hình đạt độ chính xác dự đoán trên 95% và giá trị mất mát 0.1%, cho thấy tiềm năng của RNN trong việc mô hình hóa dữ liệu chứng khoán.

Tuy nhiên, một hạn chế cố hữu của RNN cơ bản là vấn đề "biến mất gradient" (vanishing gradient), khiến nó gặp khó khăn trong việc học các phụ thuộc dài hạn (long-term dependencies). Đây là lý do các biến thể phức tạp hơn như Long Short-Term Memory (LSTM) và Gated Recurrent Unit (GRU) được phát triển. Trong một nghiên cứu so sánh của O. P. R. V. L. N. K. Sai Pranay và K. V. D. Kiran[16], các tác giả đã dự báo giá cổ phiếu Tata Steel bằng cả RNN, LSTM và GRU. Kết quả cho thấy cả LSTM (RMSE = 2.84) và GRU (RMSE = 2.91) đều vượt trội hơn đáng kể so với mô hình RNN cơ bản (RMSE = 7.55), khẳng định rằng các kiến trúc có cổng (gated) xử lý "bộ nhớ" dài hạn tốt hơn.

Khi so sánh với các mô hình thống kê, RNN và các biến thể của nó thường tỏ ra hiệu quả hơn trong việc nắm bắt tính phi tuyến. Nghiên cứu của M. Hiransha và các đồng nghiệp[17]đã so sánh trực tiếp ARIMA và một biến thể RNN (cụ thể là LSTM) trên dữ liệu giá cổ phiếu Google. Kết quả cho thấy mô hình LSTM (RMSE = 1.15) dự báo chính xác hơn đáng kể so với mô hình ARIMA (RMSE = 2.04). Nhóm tác giả kết luận rằng ARIMA, vốn là mô hình tuyến tính, không thể nắm bắt được tính phi tuyến và biến động phức tạp của thị trường chứng khoán, trong khi kiến trúc học sâu của RNN (LSTM) lại thực hiện tốt điều này.

Các nghiên cứu trên cho thấy RNN đã mở ra một hướng tiếp cận mạnh mẽ cho dự báo tài chính, có khả năng tự động học hỏi các đặc trưng phi tuyến. Mặc dù vậy, các tài liệu khoa học cũng chỉ rõ rằng RNN cơ bản (Simple RNN) thường không phải là lựa chọn tối ưu do hạn chế về bộ nhớ dài hạn, và các biến thể như LSTM và GRU thường được ưu tiên sử dụng trong các ứng dụng dự báo tài chính phức tạp.

## 2.7. Gradient Boosted Tree (GBT)

Tác giả Juan Rufino Reyes đã nghiên cứu và công bố bài báo "Nowcasting domestic liquidity in the Philippines using machine learning algorithms" vào ngày 1 tháng 12 năm 2022 [18].

Bài báo tập trung nghiên cứu việc sử dụng các thuật toán học máy để dự báo tăng trưởng thanh khoản nội địa ở Philippines. Đặc biệt, nó áp dụng các phương pháp điều chuẩn (như Ridge Regression, LASSO, Elastic Net) và các phương pháp dựa trên cây (như Random Forest, Gradient Boosted Trees) để hỗ trợ bộ mô hình kinh tế vĩ mô hiện tại của Bangko Sentral ng Pilipinas (BSP) được sử dụng để dự báo và phân tích thanh khoản. Tác giả đã sử dụng Gradient Boosted Trees để xử lý dữ liệu chuỗi thời gian phức tạp liên quan đến thanh khoản nội địa, và đã cung cấp kết quả dự báo chính xác hơn so với các mô hình chuỗi thời gian truyền thống.

Kết quả cho thấy các phương pháp điều chuẩn và dựa trên cây cung cấp các ước lượng tốt hơn so với các mô hình chuỗi thời gian truyền thống, với các dự báo một tháng trước mang lại Root Mean Square Error (RMSE) và Mean Absolute Error (MAE) thấp hơn. Hơn nữa, các phương pháp điều chuẩn và dựa trên cây giúp xác định các chỉ số kinh tế vĩ mô quan trọng để xác định các mô hình dự báo tối giản.

## 2.8. Hidden Markov Model (HMM)

Ba tác giả Raihan Tanvir, Md Tanvir Rouf Shawon và Md. Golam Rabiul Alam đã dự đoán giá cổ phiếu của DSE bằng cách sử dụng mô hình HMM. Sau quá trình huấn luyện, nhóm tác giả sử dụng phương pháp tối đa hóa hậu nghiệm (MAP) [19] để dự đoán giá cổ phiếu của Dhaka Stock Exchange (DSE) cho ngày tiếp theo. Tiến hành đánh giá độ chính xác của mô hình bằng các độ đo MAE, RMSE và MAPE cho ra các kết quả lần lượt là 2.5064, 3.4003 và 1.0265. Điều đó cho thấy mô hình cho ra kết quả đáng tin cậy trong việc dự đoán giá cổ phiếu của DSE.

## 2.9. Deep Neural Network (DNN)

Nghiên cứu của Correia et al. (2022) (*Deep Neural Networks Applied to Stock Market Sentiment*) [20] đề xuất mô hình DNN kết hợp đặc trưng kỹ thuật và cảm xúc thị trường trích xuất từ tin tức tài c[hín](https://www.zotero.org/google-docs/?V9cUfl)h. Kết quả cho thấy việc tích hợp sentiment giúp cải thiện độ chính xác dự báo biến động giá cổ phiếu so với mô hình truyền thống chỉ dùng dữ liệu giá.

Tương tự, Cristescu (2022) trong nghiên cứu *Using Market News Sentiment Analysis for Stock Market Prediction* [21] sử dụng sentiment từ tin tức để dự báo biến động thị trường. Tác giả so sánh các mô hình máy học và deep learning, chứng minh rằng DNN khai thác hiệu quả thông tin phi cấu trúc và cho hiệu suất tốt hơn các phương pháp truyền thống.

Qua các nghiên cứu, có thể thấy rằng mô hình DNN kết hợp dữ liệu cảm xúc từ văn bản tài chính đem lại cải thiện đáng kể trong dự báo thị trường chứng khoán. Việc khai thác cả dữ liệu cấu trúc (giá, khối lượng) và dữ liệu phi cấu trúc (news sentiment) đang trở thành xu hướng nổi bật nhằm phản ánh tốt hơn tâm lý thị trường — yếu tố quan trọng nhưng biến động mạnh trong tài chính

# TÀI NGUYÊN

Chương 3 sẽ trình bày về các thông tin tổng quan kèm các thống kê mô tả cho các bộ dữ liệu sử dụng trong quá trình thực hiện đề tài. Ngoài ra, chương này cũng đề cập đến các công cụ hỗ trợ như các IDE dùng để chạy mã phân tích và dự đoán trong suốt quá trình thực hiện đề tài và các công cụ hỗ trợ cộng tác.

## 3.1. Bộ dữ liệu

Nghiên cứu sử dụng ba tập dữ liệu chứng khoán được thu thập từ nền tảng tài chính Yahoo Finance [22], tương ứng với ba mã cổ phiếu: XIAOMI-W (1810.HK) đại diện cho Xiaomi Corporation, AAPL đại diện cho Apple Inc., và 005930.KS đại diện cho Samsung Electronics Co., Ltd.

Các tập dữ liệu này đều chứa chung các biến đặc trưng cho hoạt động giao dịch chứng khoán:

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Date | Ngày diễn ra giao dịch |
| Open | Giá của cổ phiếu đầu tiên được giao dịch |
| High | Giá cao nhất của cổ phiếu được giao dịch |
| Low | Giá thấp nhất của cổ phiếu được giao dịch |
| Close | Giá đóng cửa của cổ phiếu |
| Adj Close | Giá đóng cửa được điều chỉnh |
| Volume | Khối lượng giao dịch |

###### *Bảng 3.1. Mô tả chung các cột thuộc tính của 3 bộ dữ liệu*

Trong khuôn khổ đề tài, biến “Close” (giá đóng cửa) được lựa chọn làm biến mục tiêu phục vụ cho quá trình phân tích chuỗi thời gian và dự báo giá cổ phiếu. Lý do lựa chọn biến này là vì giá đóng cửa phản ánh mức giá cuối cùng trong mỗi phiên giao dịch, thường được xem là chỉ báo quan trọng nhất về xu hướng biến động của cổ phiếu và là cơ sở phổ biến trong các nghiên cứu dự báo tài chính.

### 3.1.1. Tập dữ liệu chứng khoán của Apple Inc.

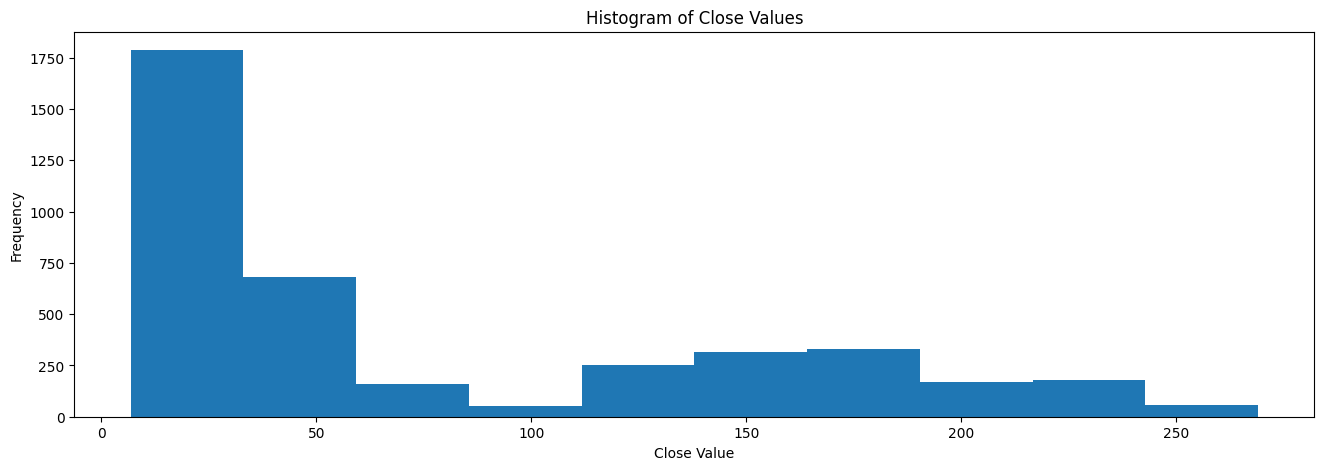
Tập dữ liệu này được ghi nhận trong khoảng thời gian từ 31/12/2009 đến 29/10/2025 và quan sát được 3982 dòng giá trị với đơn vị đo cho biến Close là USD.

Bên dưới là bảng 3.2 dùng thể hiện các chỉ số thống kê mô tả phân tích được trên biến Close từ tập dữ liệu này:

|  |  |
| --- | --- |
| **Thông số thống kê** | **Giá trị** |
| Count | 3982 |
| Mean | 77.74 |
| Std | 72.31 |
| Min | 6.86 |
| 25% | 21.27 |
| 50% | 40.25 |
| 75% | 143.36 |
| Max | 269 |
| CV | 0.93 |
| Skewness | 0.89 |
| Kurtosis | -0.66 |

###### *Bảng 3.2. Các chỉ số thống kê mô tả của cột Close trong tập dữ liệu Apple*

###### *Hình 3.1. Biểu đồ chuỗi thời gian của cột Close cho tập dữ liệu Apple*

Biểu đồ chuỗi thời gian (Hình 3.1) thể hiện diễn biến giá đóng cửa của cổ phiếu Apple trong giai đoạn từ năm 2010 đến năm 2025. Dữ liệu cho thấy xu hướng tăng trưởng mạnh mẽ và rõ rệt theo thời gian, đặc biệt từ giai đoạn sau năm 2019, khi giá cổ phiếu tăng nhanh và đạt đỉnh vào các năm gần đây. Nhìn chung, chuỗi giá có xu hướng (trend) rõ ràng và không dừng (non-stationary).

###### *Hình 3.2. Biểu đồ Histogram của cột Close cho tập dữ liệu Apple*

Biểu đồ Histogram (Hình 3.2) cho thấy phân phối giá đóng cửa của cổ phiếu có đặc điểm lệch phải (right-skewed) rõ rệt. Phần lớn các giá trị tập trung trong khoảng 0 – 60 USD, thể hiện rằng phần lớn thời gian giá cổ phiếu duy trì ở mức thấp. Trong khi đó, số lượng quan sát ở các mức giá cao hơn (từ 100 USD trở lên) giảm dần đáng kể, phản ánh sự xuất hiện hiếm hơn của các mức giá cao.

###### *Hình 3.3. Biểu đồ Boxplot của cột Close cho tập dữ liệu Apple*

Biểu đồ Boxplot (Hình 3.3) cho thấy phân bố dữ liệu có độ lệch phải rõ rệt. Giá trị trung vị khoảng 40 USD, trong khi phần lớn giá trị nằm trong khoảng 20–140 USD. Sự kéo dài của phần whisker phía trên cho thấy có những giai đoạn giá cổ phiếu tăng mạnh vượt xa mức trung bình, phản ánh xu hướng tăng trưởng dài hạn của cổ phiếu Apple. Dữ liệu không xuất hiện nhiều điểm ngoại lai, phản ánh đặc trưng tăng trưởng ổn định của cổ phiếu trong dài hạn.

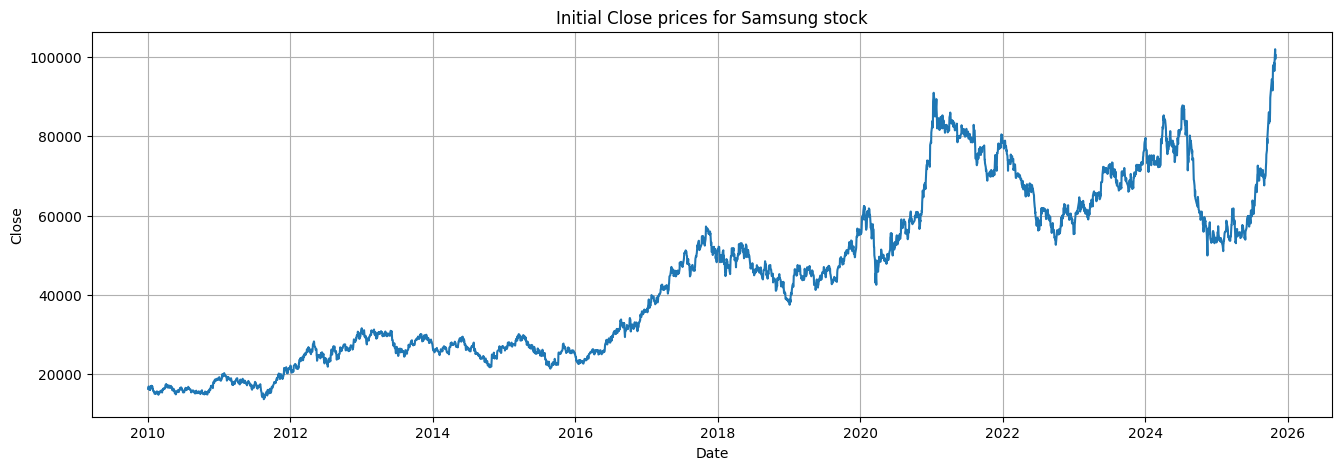
### 3.1.2. Tập dữ liệu chứng khoán của Samsung Electronics Co., Ltd

Tập dữ liệu này được ghi nhận trong khoảng thời gian từ 04/01/2010 đến 29/10/2025 và quan sát được 3892 dòng giá trị với đơn vị đo cho biến Close là KRW.

Bên dưới là bảng 3.3 dùng thể hiện các chỉ số thống kê mô tả phân tích được trên biến Close từ tập dữ liệu này:

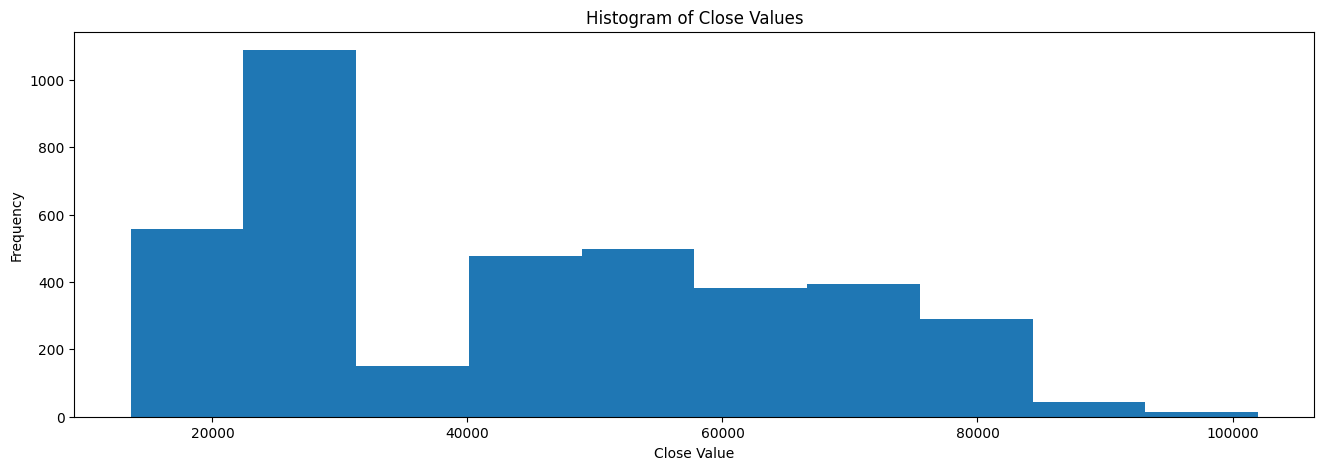
|  |  |
| --- | --- |
| **Thông số thống kê** | **Giá trị** |
| Count | 3892 |
| Mean | 43733.77 |
| Std | 20683.64 |
| Min | 13440 |
| 25% | 25620 |
| 50% | 43775 |
| 75% | 59600 |
| Max | 100600 |
| CV | 0.47 |
| Skewness | 0.35 |
| Kurtosis | -1.09 |

###### *Bảng 3.3. Thống kê các thông số của cột Close trong tập dữ liệu Samsung*



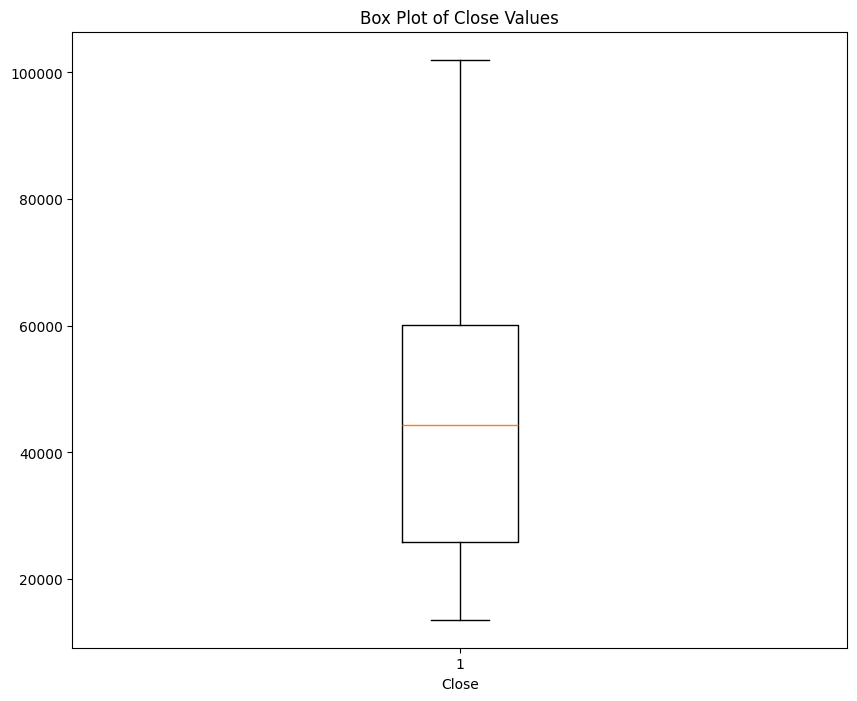
###### *Hình 3.4. Biểu đồ chuỗi thời gian của cột Close cho tập dữ liệu Samsung*

Biểu đồ chuỗi thời gian (Hình 3.4) thể hiện diễn biến giá đóng cửa của cổ phiếu Samsung Electronics Co., Ltd trong giai đoạn 2010–2025. Biểu đồ cho thấy sự tăng trưởng đáng kể trong dài hạn, song song với các giai đoạn dao động mạnh. Từ năm 2009 đến 2016, giá cổ phiếu tăng ổn định, nhưng ở giai đoạn sau lại ghi nhận các biến động lớn. Đến năm 2025, cổ phiếu đạt mức giá cao nhất hơn 100,000 KRW, phản ánh tiềm năng tăng trưởng mạnh nhưng cũng thể hiện tính biến động đặc trưng của dữ liệu tài chính.



###### *Hình 3.5. Biểu đồ Histogram của cột Close tập dữ liệu Samsung*

Biểu đồ Histogram (Hình 3.5) cho thấy phân bố giá đóng cửa của cổ phiếu Samsung Electronics không đối xứng và có xu hướng lệch phải, trong đó phần lớn giá trị tập trung ở khoảng từ 20,000 đến 50,000 KRW. Các mức giá cao trên 60,000 KRW xuất hiện với tần suất thấp hơn, phản ánh quá trình tăng giá dần trong những năm gần đây.



###### *Hình 3.6. Biểu đồ Boxplot của cột Close tập dữ liệu Samsung*

Biểu đồ Boxplot (Hình 3.6) cho thấy sự phân bố của giá đóng cửa cổ phiếu Samsung Electronics trong giai đoạn nghiên cứu. Giá trị trung vị khoảng 45,000 KRW, trong khi khoảng tứ phân vị trải dài từ 25,000 đến 60,000 KRW, phản ánh mức độ biến động tương đối cao của cổ phiếu. Dữ liệu không xuất hiện nhiều điểm ngoại lai, cho thấy biến động giá nằm trong phạm vi hợp lý và phản ánh đặc trưng tăng trưởng ổn định của cổ phiếu trong dài hạn.

### 3.1.3. Tập dữ liệu chứng khoán của Xiaomi Corporation

Tập dữ liệu này được ghi nhận trong khoảng thời gian từ 09/07/2018 đến 29/10/2025 và quan sát được 1800 dòng giá trị với đơn vị đo cho biến Close HKD.

Bên dưới là bảng 3.4 dùng thể hiện các chỉ số thống kê mô tả phân tích được trên biến Close từ tập dữ liệu này:

|  |  |
| --- | --- |
| **Thông số thống kê** | **Giá trị** |
| Count | 1800 |
| Mean | 19.84 |
| Std | 12.57 |
| Min | 8.35 |
| 25% | 11.62 |
| 50% | 14.83 |
| 75% | 24.38 |
| Max | 60.15 |
| CV | 0.63 |
| Skewness | 1.71 |
| Kurtosis | 2.08 |

###### *Bảng 3.4. Thống kê các thông số của cột Close trong tập dữ liệu Xiaomi*

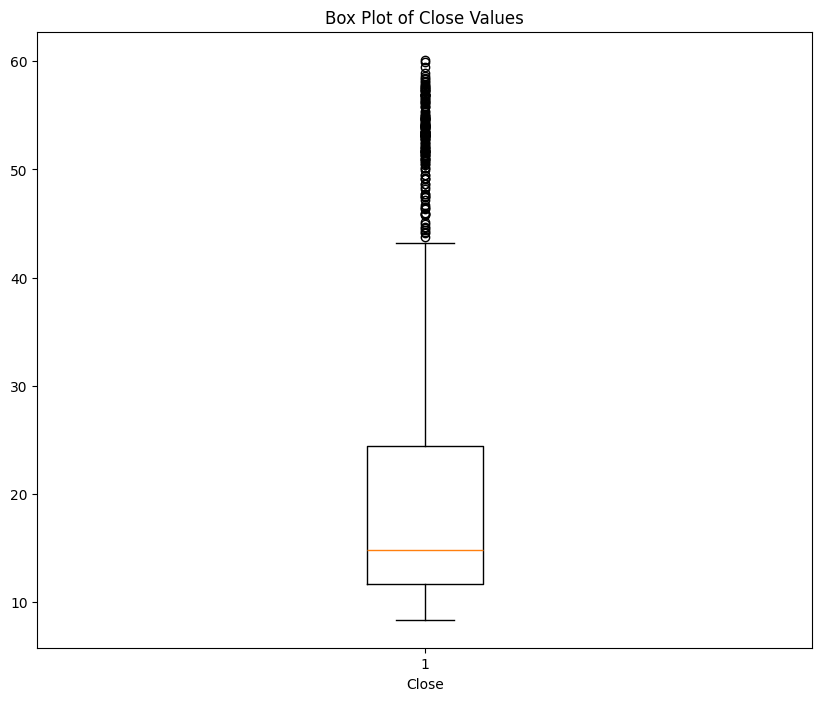


###### *Hình 3.7. Biểu đồ chuỗi thời gian của cột Close tập dữ liệu Xiaomi*

Biểu đồ chuỗi thời gian (Hình 3.7) thể hiện diễn biến giá đóng cửa của cổ phiếu Xiaomi Corporation trong giai đoạn 2018–2025. Giai đoạn đầu cho thấy xu hướng giảm mạnh, tiếp đến là giai đoạn tăng trưởng đáng kể từ năm 2020 đến 2021 với đỉnh giá xấp xỉ 35 HKD. Sau đó, cổ phiếu trải qua thời kỳ điều chỉnh và duy trì mức giá tương đối thấp trong năm 2022–2023. Từ giữa năm 2024, xu hướng tăng trở lại rõ rệt, đạt đỉnh 60 HKD trước khi có dấu hiệu điều chỉnh nhẹ vào cuối giai đoạn quan sát.

###### *Hình 3.8. Biểu đồ Histogram của cột Close tập dữ liệu Xiaomi*

Biểu đồ Histogram (Hình 3.8) cho thấy phân phối giá đóng cửa của cổ phiếu có đặc điểm lệch phải (right-skewed) rõ rệt. Phần lớn các giá trị tập trung trong khoảng 8 –30 HKD, thể hiện rằng phần lớn thời gian giá cổ phiếu duy trì ở mức thấp. Trong khi đó, số lượng quan sát ở các mức giá cao hơn (từ 35 HKD trở lên) giảm dần đáng kể, phản ánh sự xuất hiện hiếm hơn của các mức giá cao.



###### *Hình 3.9. Biểu đồ Boxplot của cột Close tập dữ liệu Xiaomi*

Biểu đồ Boxplot cho thấy phân bố dữ liệu có độ lệch đáng kể và sự xuất hiện của nhiều giá trị ngoại lai (outliers) ở phía trên. Phần lớn dữ liệu tập trung trong khoảng từ 10 đến 25 HKD, với giá trị trung vị nằm khoảng giữa vùng này, phản ánh rằng phần lớn các phiên giao dịch có giá đóng cửa ở mức thấp đến trung bình. Phần đuôi trên của hộp kéo dài hơn đáng kể, cùng với sự xuất hiện dày đặc của các điểm ngoại lai phía trên, cho thấy sự biến động tăng mạnh trong một số giai đoạn.

## 3.2. Công cụ sử dụng

### 3.2.1. IDE

* Visual Studio Code: Dùng thực hiện các đoạn mã Python trong các file .ipynb để phân tích và chạy mô hình dự đoán.
* RStudio: Dùng thực hiện các đoạn mã R trong các file .r để phân tích và chạy mô hình dự đoán.

### 3.2.2. Các nền tảng hỗ trợ cộng tác

* Github: Dùng để chứa mã nguồn chung trong quá trình thực hiện đề tài [23].
* Google Doc: Dùng để thực hiện viết báo cáo đồng thời.
* Google Sheet: Dùng để phân chia giai đoạn đề tài và phân công nhiệm vụ cụ thể cho từng thành viên.

# PHƯƠNG PHÁP

Nội dung chương 4: Phương pháp sẽ trình bày chi tiết về lý thuyết các mô hình, diễn giải chi tiết công thức cũng như mô tả ý tưởng xây dựng mô hình trong dự đoán. Bên cạnh đó các độ đo đánh giá hiệu suất mô hình cũng sẽ được trình bày trong chương này.

## 4.1. Linear Regression

### 4.1.1 Giới thiệu

Trong lĩnh vực thống kê và học máy (machine learning), Linear Regression (Hồi quy tuyến tính) là một trong những mô hình cơ bản và phổ biến nhất được sử dụng để dự đoán giá trị của một biến phụ thuộc (dependent variable) dựa trên một hoặc nhiều biến độc lập (independent variables). Mục tiêu của mô hình là tìm ra mối quan hệ tuyến tính giữa các biến để từ đó dự đoán hoặc giải thích hành vi của dữ liệu.

Linear Regression được áp dụng rộng rãi trong nhiều lĩnh vực như kinh tế, tài chính, marketing, y học, kỹ thuật dữ liệu, v.v., nhằm hỗ trợ ra quyết định và phân tích xu hướng.

### 4.1.2 Ý tưởng của mô hình

Ý tưởng cốt lõi của Linear Regression là xây dựng một đường thẳng (hoặc siêu phẳng trong không gian đa chiều)sao cho khoảng cách giữa các điểm dữ liệu thực tế và đường dự đoán là nhỏ nhất.

Nói cách khác, mô hình cố gắng tối thiểu hóa sai số bình phương trung bình (Mean Squared Error - MSE) giữa giá trị dự đoán và giá trị thực tế.

Mô hình giả định rằng mối quan hệ giữa biến phụ thuộc yy và các biến độc lập x1,x2,...,xn​ là tuyến tính, tức là có thể biểu diễn dưới dạng phương trình:

Trong đó:

* : biến phụ thuộc (dependent variable) cần dự đoán.
* : biến độc lập (independent variable) được sử dụng để dự đoán giá trị của .
* : hệ số góc (intercept) của đường hồi quy, đại diện cho giá trị dự đoán của khi .
* : hệ số hồi quy (regression coefficient), đại diện cho mức độ thay đổi của dựa trên mỗi đơn vị thay đổi của .
* : lỗi ngẫu nhiên (random error), biểu thị sự không thể tránh khỏi của mô hình trong việc mô phỏng dữ liệu thực tế.

### 4.1.3 Hình thành mô hình trong dự đoán

Trong thực tế, quá trình hình thành mô hình hồi quy tuyến tính bao gồm các bước chính:

1. Thu thập và tiền xử lý dữ liệu: Chuẩn bị dữ liệu đầu vào, xử lý giá trị thiếu, loại bỏ ngoại lệ (outliers), và chuẩn hóa dữ liệu nếu cần.
2. Xây dựng mô hình: Sử dụng công thức hồi quy tuyến tính để ước lượng các hệ số ,​ bằng phương pháp bình phương tối thiểu (Least Squares Method).
3. Đánh giá mô hình: Kiểm tra độ phù hợp của mô hình qua các chỉ số như R-squared (R²), RMSE, MAE, và kiểm định giả thuyết về ý nghĩa thống kê của các hệ số.
4. Dự đoán và ứng dụng: Khi mô hình đã được huấn luyện, ta có thể sử dụng nó để dự đoán giá trị mới của biến phụ thuộc dựa trên các giá trị của biến độc lập.

Trong trường hợp Simple Linear Regression (hồi quy tuyến tính đơn), mô hình được mô tả đơn giản như sau:

Trong đó:

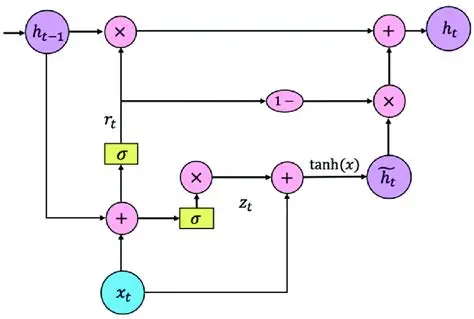
* ​: giá trị trung bình của y khi x=0.
* ​: độ dốc của đường hồi quy, cho biết khi xx tăng một đơn vị thì yy thay đổi trung bình bao nhiêu.
* ε: phần sai số ngẫu nhiên, thể hiện các yếu tố không giải thích được bằng x.

## 4.2. Gate Recurrent Unit (GRU)

### 4.2.1 Giới thiệu chung

Gated Recurrent Unit (GRU) là một biến thể của mạng nơ-ron hồi quy (RNN) được giới thiệu bởi Kyunghyun Cho và cộng sự (2014) trong công trình *“Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”*[24].  
 Mục tiêu của GRU là khắc phục vấn đề triệt tiêu độ dốc (vanishing gradient) thường gặp ở RNN truyền thống, đồng thời vẫn giữ được khả năng học các quan hệ dài hạn trong chuỗi dữ liệu nhưng với cấu trúc đơn giản hơn LSTM.

Ngoài ra, để trực quan hóa cơ chế hoạt động của LSTM và GRU, có thể tham khảo bài viết “Understanding LSTM Networks” của Christopher Olah (2015)[25], cung cấp minh họa trực quan về các cổng và trạng thái ẩn.



###### *Hình 4.1. Kiến trúc GRU*

### 4.2.2 Cấu trúc và cơ chế hoạt động

GRU là mạng nơ-ron hồi quy có cơ chế “cổng” (gates) giúp kiểm soát luồng thông tin trong quá trình huấn luyện. Mỗi đơn vị GRU bao gồm **hai cổng chính**:

#### **Cổng cập nhật (Update Gate – )**

Cổng này xác định **lượng thông tin từ trạng thái ẩn trước đó** cần được **giữ lại** để truyền sang bước thời gian hiện tại:

Trong đó:

: dữ liệu đầu vào tại thời điểm t

: là hàm sigmoid được sử dụng để đảm bảo giá trị đầu ra nằm trong [0,1]

: là các ma trận trọng số.

: là trạng thái ẩn từ bước thời gian trước đó.

**Cổng đặt lại (Reset Gate – )**

Cổng này quyết định phần thông tin từ trạng thái ẩn trước đó nên bị quên đi, giúp mô hình chỉ giữ lại các thông tin thực sự cần thiết.

Trong đó:

: dữ liệu đầu vào tại thời điểm t

: là hàm sigmoid được sử dụng để đảm bảo giá trị đầu ra nằm trong [0,1]

: là các ma trận trọng số.

: là trạng thái ẩn từ bước thời gian trước đó.

**Trạng thái ẩn ứng viên (Candidate Hidden State – )**

Sau khi tính được ​, GRU sử dụng nó để xác định **trạng thái ẩn ứng viên**, được xem là thông tin mới có khả năng thay thế cho trạng thái ẩn cũ:

Trong đó:

: là các ma trận trọng số.

: là vector sai số (bias)

: là dữ liệu đầu vào hiện tại.

: là trạng thái ẩn từ bước thời gian trước đó.

: Cổng đặt lại

: sử dụng để đảm bảo giá trị đầu ra nằm trong khoảng từ -1 đến 1.

**Trạng thái ẩn hiện tại (Hidden State – ​)**

Cuối cùng, trạng thái ẩn hiện tại được xác định bằng cách kết hợp giữa trạng thái ẩn cũ và trạng thái ẩn ứng viên, được điều tiết bởi cổng cập nhật ​:

Cách kết hợp này cho phép GRU **tự động quyết định** phần nào của thông tin cũ cần giữ lại và phần nào cần cập nhật mới — giúp mạng ghi nhớ dài hạn mà không bị triệt tiêu độ dốc.

### 4.2.3 Ưu và nhược điểm của GRU

**Ưu điểm của GRU:**

**Giảm thiểu vấn đề triệt tiêu độ dốc:** GRU giúp duy trì thông tin trong chuỗi dài hiệu quả hơn so với RNN cơ bản

**Mô hình đơn giản hơn LSTM:** GRU chỉ có hai cổng (cập nhật và đặt lại) so với LSTM có ba cổng (đầu vào, đầu ra và quên), điều này giúp giảm bớt độ phức tạp của mô hình.

**Thời gian huấn luyện ngắn hơn:** Do mô hình đơn giản hơn, thời gian huấn luyện của GRU thường ngắn hơn so với LSTM.

**Nhược điểm của GRU:**

**Thiếu cổng đầu ra (Output Gate):** Khi cần điều tiết tinh vi giữa trạng thái ẩn và đầu ra, GRU có thể kém linh hoạt hơn so với LSTM

**Hiệu quả phụ thuộc bài toán:** Trong một số trường hợp có độ dài chuỗi cực lớn hoặc cần biểu diễn thông tin phức tạp, LSTM có thể cho kết quả tốt hơn.

## 4.3 Autoregressive Integrated Moving Average (ARIMA)

### 4.3.1. Giới thiệu

Mô hình ARIMA (Autoregressive Integrated Moving Average) là một trong những phương pháp thống kê phổ biến nhất trong dự báo chuỗi thời gian. Mô hình này giả định rằng giá trị tương lai của một biến có thể được biểu diễn như một tổ hợp tuyến tính của các giá trị trong quá khứ và các sai số ngẫu nhiên, qua đó mô tả mối quan hệ nội tại của dữ liệu chuỗi theo thời gian.

Từ quan điểm dự báo tài chính, ARIMA đặc biệt hiệu quả trong việc mô hình hóa xu hướng và dự đoán ngắn hạn của các chuỗi có tính tuyến tính, ví dụ như giá đóng cửa cổ phiếu [26].

Ưu điểm nổi bật của ARIMA là chỉ yêu cầu dữ liệu nội tại của chuỗi thời gian đang xét, không cần biến độc lập khác. Điều này giúp phương pháp linh hoạt và có thể áp dụng song song cho nhiều chuỗi thời gian khác nhau [27] [26]. Tuy nhiên, mô hình cũng có hạn chế khi khó mô tả các mối quan hệ phi tuyến tính hoặc các bất thường trên thị trường, chỉ hoạt động tốt khi dữ liệu có tính dừng hoặc được biến đổi về chuỗi dừng và kém chính xác khi dự đoán dài hạn [28].

### 4.3.2 Cấu trúc và công thức mô hình

Mô hình ARIMA được ký hiệu là ARIMA(p, d, q), trong đó:

* **p** là số bậc tự hồi quy (Autoregressive – AR).
* **d** là số lần sai phân cần để chuỗi đạt tính dừng (Integrated – I).
* **q** là số bậc trung bình trượt (Moving Average – MA).

Phương trình tổng quát có dạng:

trong đó ​ là giá trị chuỗi tại thời điểm t,​ là nhiễu ngẫu nhiên, ​ và là các hệ số mô hình.  
Trước khi áp dụng ARIMA, điều kiện tiên quyết là chuỗi dữ liệu **phải có tính dừng**: các giá trị mean, variance, autocorrelation không thay đổi theo thời gian và nó không bao hàm yếu tố xu hướng [29]. Việc kiểm tra xem một chuỗi có tính dừng hay không sẽ thông qua kiểm định ADF.

Nếu chuỗi không dừng thì cần thực hiện sai phân. Quá trình sai phân bậc d của chuỗi được thực hiện như sau:

- Sai phân bậc 1:

- Sai phân bậc d: :

Sau khi chuỗi đạt tính dừng, việc xác định giá trị p và q thường được thực hiện bằng cách phân tích đồ thị tự tương quan (ACF) và tự tương quan riêng phần (PACF).

### 4.3.3 Ứng dụng trong nghiên cứu

Trong nghiên cứu này, mô hình ARIMA được sử dụng để **dự báo giá đóng cửa cổ phiếu** của ba công ty: Apple Inc. (AAPL), Samsung Electronics Co., Ltd. (005930.KS) và Xiaomi Corporation (1810.HK). Biến được lựa chọn là cột “Close” – giá đóng cửa mỗi phiên – vì đây là chỉ báo cuối cùng phản ánh mức độ định giá thị trường và thường được sử dụng trong các nghiên cứu dự báo tài chính [30].  
 Các bước triển khai theo thứ tự:

* Chia dataset thành các tập train, validate và test.
* Sử dụng hàm auto\_arima() để xác định tham số p,d,q tối ưu dựa trên tiêu chí AIC/BIC.
* Huấn luyện mô hình trên tập train và thực hiện đánh giá trên tập validate.
* Huấn luyện lại mô hình trên tập train kết hợp validate và thực hiện đánh giá trên tập test.
* Huấn luyện lại mô hình trên tập train kết hợp validate và test và cuối cùng thực hiện dự báo cho các phiên giao dịch trong 30 ngày tiếp theo.

## 4.4. Exponential Smoothing (ETS)

### 4.4.1. Giới thiệu

Exponential Smoothing (ETS) là một phương pháp dự báo trong phân tích chuỗi thời gian, trong đó các quan sát được gán trọng số giảm dần theo thời gian. Các quan sát gần hiện tại được xem là quan trọng hơn và do đó có ảnh hưởng lớn hơn đến dự báo so với các quan sát xa hơn trong quá khứ. Nguyên lý của phương pháp này dựa trên việc làm “mịn” (smoothing) chuỗi dữ liệu thông qua một hoặc nhiều tham số trơn (smoothing parameters), giúp loại bỏ nhiễu ngẫu nhiên và nắm bắt xu hướng thực của dữ liệu [31].

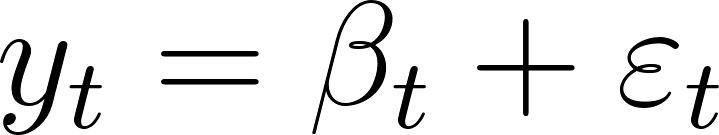
Phương pháp Exponential Smoothing được đề xuất lần đầu bởi Robert G. Brown (1959) và sau đó được Charles C. Holt và Peter R. Winters phát triển thành các biến thể phức tạp hơn. Từ đó, ETS trở thành một trong những kỹ thuật dự báo phổ biến nhất trong các bài toán chuỗi thời gian nhờ tính đơn giản, hiệu quả và khả năng thích ứng linh hoạt với dữ liệu có xu hướng hoặc mùa vụ [32].

Ba biến thể chính của Exponential Smoothing bao gồm:

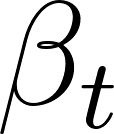
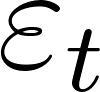
* **Simple Exponential Smoothing (SES):** Dùng cho chuỗi thời gian không có xu hướng rõ rệt hoặc tính mùa vụ. Phương pháp này trung bình hóa các giá trị quá khứ với trọng số giảm dần theo thời gian.
* **Double Exponential Smoothing (DES):** Mở rộng từ SES để xử lý chuỗi dữ liệu có xu hướng (trend). DES sử dụng hai tham số làm mịn: một cho mức trung bình và một cho xu hướng.
* **Triple Exponential Smoothing (TES) hay Holt–Winters:** Dành cho chuỗi có tính mùa vụ, kết hợp đồng thời ba yếu tố: mức trung bình (level), xu hướng (trend) và mùa vụ (seasonality)

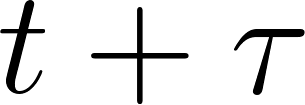
### 4.4.2. Cơ sở lý thuyết và công thức mô hình

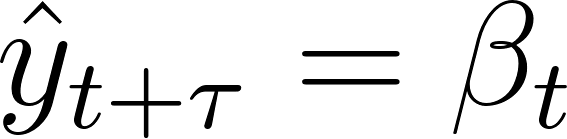
Phương pháp Exponential Smoothing giả định rằng chuỗi dữ liệu yty\_tyt​ dao động xung quanh một giá trị trung bình thay đổi chậm theo thời gian. Mô hình tổng quát có thể biểu diễn như sau:​

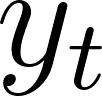
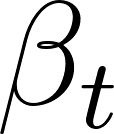
[](https://www.codecogs.com/eqnedit.php?latex=y_t%20%3D%20%5Cbeta_t%20%2B%20%5Cvarepsilon_t#0)

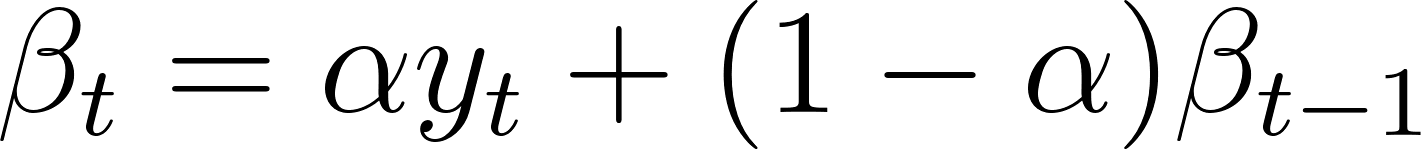
Trong đó:

* [](https://www.codecogs.com/eqnedit.php?latex=%5Cbeta_t#0): giá trị trung bình (level) tại thời điểm ttt, phản ánh xu hướng ngắn hạn của chuỗi.
* [](https://www.codecogs.com/eqnedit.php?latex=%5Cvarepsilon_t#0): thành phần nhiễu ngẫu nhiên tại thời điểm ttt.

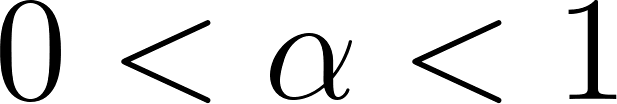
Dự báo tại thời điểm [](https://www.codecogs.com/eqnedit.php?latex=t%20%2B%20%5Ctau#0) (với [](https://www.codecogs.com/eqnedit.php?latex=%5Ctau#0) là số bước dự báo) được xác định theo công thức:

[](https://www.codecogs.com/eqnedit.php?latex=%5Chat%7By%7D_%7Bt%2B%5Ctau%7D%20%3D%20%5Cbeta_t#0)

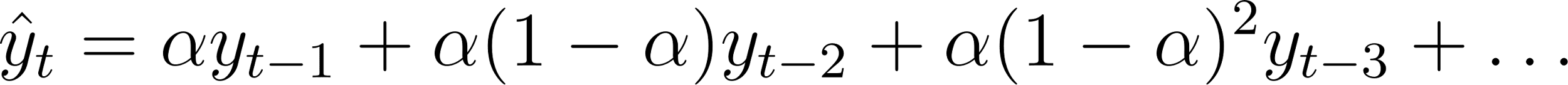
Khi có một quan sát mới [](https://www.codecogs.com/eqnedit.php?latex=y_t#0)​, mô hình cập nhật giá trị ước lượng trung bình [](https://www.codecogs.com/eqnedit.php?latex=%5Cbeta_t#0) ​dựa trên công thức làm mịn theo cấp số mũ:

[](https://www.codecogs.com/eqnedit.php?latex=%5Cbeta_t%20%3D%20%5Calpha%20y_t%20%2B%20(1%20-%20%5Calpha)%5Cbeta_%7Bt-1%7D#0)

Trong đó:

* [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) là hệ số làm mịn (smoothing parameter), có giá trị trong khoảng [](https://www.codecogs.com/eqnedit.php?latex=0%20%3C%20%5Calpha%20%3C%201#0)
* Giá trị [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) càng lớn thì dự báo càng nhạy với dữ liệu mới (phản ứng nhanh với biến động), ngược lại khi [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) nhỏ thì mô hình phản ứng chậm hơn và dự báo trở nên ổn định hơn.

Cách gán trọng số giảm dần theo thời gian được thể hiện qua chuỗi:

[](https://www.codecogs.com/eqnedit.php?latex=%5Chat%7By%7D_t%20%3D%20%5Calpha%20y_%7Bt-1%7D%20%2B%20%5Calpha(1%20-%20%5Calpha)%20y_%7Bt-2%7D%20%2B%20%5Calpha(1%20-%20%5Calpha)%5E2%20y_%7Bt-3%7D%20%2B%20%5Cdots#0)

Điều này cho thấy các quan sát càng xa hiện tại thì trọng số ảnh hưởng đến dự báo càng nhỏ, nhờ đó giúp mô hình nhấn mạnh vào thông tin mới nhất mà vẫn duy trì độ ổn định tổng thể.

### 4.4.3. Ứng dụng trong nghiên cứu

Trong nghiên cứu này, phương pháp Exponential Smoothing (ETS) được áp dụng để dự báo giá đóng cửa cổ phiếu của ba công ty: Apple Inc. (AAPL), Samsung Electronics Co., Ltd. (005930.KS) và Xiaomi Corporation (1810.HK). Biến được sử dụng là giá đóng cửa (Close) – một chỉ số phản ánh đầy đủ giá trị thị trường của cổ phiếu trong mỗi phiên giao dịch.

Quy trình thực hiện được triển khai theo các bước sau:

1. Tiền xử lý dữ liệu và chuẩn hóa giá trị bằng phương pháp **Min–Max Scaling** để đảm bảo tính ổn định trong huấn luyện.
2. Chia tập dữ liệu thành ba phần: **train**, **validate**, và **test** theo tỷ lệ 6:2:2.
3. Tối ưu hệ số làm mịn α bằng cách lặp qua nhiều giá trị từ 0.1 đến 0.9, lựa chọn giá trị cho sai số nhỏ nhất (theo tiêu chí MAPE hoặc RMSE).
4. Huấn luyện mô hình trên tập train, đánh giá trên tập validate, sau đó huấn luyện lại trên tập train + validate để kiểm định trên tập test.
5. Cuối cùng, sử dụng mô hình tối ưu để **dự báo 30 ngày kế tiếp** của chuỗi thời gian.

Phương pháp này cho phép đánh giá mức độ ổn định của chuỗi giá cổ phiếu, đồng thời cung cấp khả năng dự báo ngắn hạn có độ tin cậy cao.

### 4.4.4. Ưu điểm và nhược điểm

**Ưu điểm:**

* Đơn giản, dễ hiểu và dễ triển khai so với các mô hình thống kê phức tạp.
* Chi phí tính toán thấp, phù hợp với các ứng dụng dự báo thời gian thực.
* Có thể mở rộng linh hoạt cho dữ liệu có xu hướng hoặc mùa vụ thông qua DES và TES.

**Nhược điểm:**

* Giả định dữ liệu có cấu trúc ổn định hoặc thay đổi chậm theo thời gian, nên hiệu quả giảm khi chuỗi có biến động đột ngột.
* Không mô tả được mối quan hệ phi tuyến tính giữa các yếu tố trong chuỗi.
* Hiệu suất dự báo dài hạn thường kém chính xác hơn so với các mô hình học sâu như LSTM hay GRU.

Tổng kết lại, **Exponential Smoothing (ETS)** là một phương pháp thống kê mạnh mẽ, trực quan và đáng tin cậy trong các bài toán dự báo chuỗi thời gian ngắn hạn. Dù không có khả năng học đặc trưng phức tạp như các mô hình học sâu, ETS vẫn là công cụ nền tảng và đóng vai trò quan trọng trong việc đánh giá và so sánh hiệu năng với các mô hình dự báo tiên tiến hơn trong nghiên cứu này.

## 4.5. Long Short-Term Memory (LSTM)

### 4.5.1. Giới thiệu

LSTM (Long Short-Term Memory) là một biến thể quan trọng của mạng nơ-ron hồi quy (Recurrent Neural Network – RNN), được giới thiệu bởi Hochreiter và Schmidhuber vào năm 1997 trong bài báo *“Long Short-Term Memorydvi”*  [33]. Mục tiêu của LSTM là khắc phục vấn đề mất mát/khai triển gradient và phụ thuộc xa (long-term dependencies) mà RNN truyền thống gặp phải.

Một lớp LSTM bao gồm nhiều khối nhớ (memory blocks), mỗi khối chứa một ô nhớ (cell state) và ba cổng điều khiển:

* Cổng quên (Forget gate)
* Cổng đầu vào (Input gate)
* Cổng đầu ra (Output gate)

Nhờ cơ chế kiểm soát thông tin qua các cổng, LSTM có khả năng lưu trữ hoặc loại bỏ thông tin một cách có chủ đích, từ đó học được các phụ thuộc dài hạn tốt hơn so với RNN thông thường.

A picture containing screenshot, diagram

Description automatically generated

###### *Hình 4.2. Kiến trúc của LSTM*

### 4.5.2 Công thức tính toán các cổng trong LSTM:

Tại mỗi thời điểm t, LSTM thực hiện các tính toán sau:

* *Forget gate:*
* *Input gate:*
* *Cell gate:*
* *Cell state:*
* *Ouput gate:*
* *Hidden state:*

Trong đó:

: Là các ma trận trọng số ứng với các cổng Forger, Input, Cell, Output. Được sử dụng để ánh xạ các đặc trưng đầu vào ()

: Là các ma trận trọng số ứng với các cổng Forger, Input, Cell, Output. Được sử dụng để ánh xạ các trạng thái trước đó ()

các giá trị cổng nằm trong khoảng [0, 1]

trạng thái ô nhớ

: Đại diện cho trạng thái ẩn (hidden state) tại thời điểm t. Trạng thái ẩn là đầu ra của mỗi bước thời gian trong quá trình lan truyền thuận của LSTM.

### 4.5.3. Ý tưởng của mô hình

**Cell state – đường truyền ký ức trung tâm**

Trong thiết kế gốc, cell state được xem như “đường dẫn lỗi không suy giảm” (constant error carousel). Nhờ các phép toán tuyến tính đơn giản và sự điều khiển của các cổng, cell state có khả năng mang thông tin xuyên suốt nhiều thời điểm mà không bị biến dạng. Điều này giúp LSTM ghi nhớ các phụ thuộc dài hạn tốt hơn nhiều so với RNN truyền thống [34].

.

A diagram of a flowchart

Description automatically generated with low confidence

###### *Hình 4.3. Hình ảnh Cell State*

**Forget gate – chọn lọc loại bỏ thông tin quá khứ**

Gers et al. (2000) [35] bổ sung cổng quên vào kiến trúc LSTM nhằm cho phép mô hình tự động loại bỏ các thông tin không còn cần thiết. Cổng này nhận đầu vào là trạng thái ẩn trước đó ​ và giá trị đầu vào hiện tại ​, từ đó xác định phần nào của cell state cũ ​ sẽ được giữ lại.

* Nếu giá trị gần 1 → thông tin được giữ lại
* Nếu gần 0 → thông tin bị xóa bỏ

Nhờ đó, LSTM có khả năng điều chỉnh bộ nhớ linh hoạt theo từng bối cảnh [34].

A picture containing diagram, rectangle, line, design

Description automatically generated

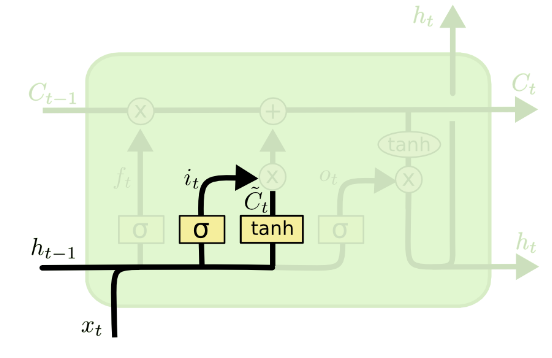
###### *Hình 4.4. Hình ảnh Forget Gate*

**Input gate – quyết định thông tin mới cần ghi nhớ**

Trong kiến trúc ban đầu, input gate có vai trò kiểm soát việc cập nhật thông tin mới vào cell state. Cổng này gồm hai thành phần:

* **Lớp sigma**, quyết định *những phần nào* của thông tin mới sẽ được ghi vào.
* **Lớp tanh**, sinh ra vector ứng viên – đại diện cho nội dung mới được đề xuất thêm vào bộ nhớ.

Hai thành phần này kết hợp với nhau thông qua phép nhân để đảm bảo chỉ những thông tin quan trọng mới được lưu trữ [34].



###### *Hình 4.5. Hình ảnh về bước tiếp theo*

**Cập nhật Cell state – hòa trộn thông tin cũ và mới**

Cell state mới được tính bằng việc kết hợp:

* phần thông tin quá khứ được giữ lại bởi forget gate
* và phần thông tin mới được chọn lọc bởi input gate

Cách cập nhật này cho phép LSTM vừa duy trì tính liên tục của chuỗi thời gian, vừa thích ứng với những biến đổi mới của dữ liệu [34].

A diagram of a flowchart

Description automatically generated with low confidence

###### *Hình 4.6. Hình ảnh về việc tính toán lại Cell State*

**Output gate – tạo giá trị đầu ra**

Theo Graves (2012), output gate xác định thông tin nào từ cell state sẽ góp phần vào trạng thái ẩn hiện tại ​. Cell state được đưa qua hàm tanh để thu về khoảng [−1,1], sau đó được lọc bởi output gate để tạo ra đầu ra cuối cùng.  
 Điều này giúp mô hình chỉ đưa ra những thông tin thật sự quan trọng tại thời điểm hiện tại, đồng thời vẫn bảo toàn được bộ nhớ dài hạn .

A diagram of a flowchart

Description automatically generated with low confidence

###### *Hình 4.7. Hình ảnh Output Gate*

### 4.5.4. Hình thành mô hình trong dự đoán

Dựa trên cơ chế hoạt động của LSTM, quá trình xây dựng mô hình dự đoán chuỗi thời gian được bắt đầu từ việc chuẩn bị dữ liệu đầu vào. Dữ liệu gốc là một chuỗi các giá trị theo thời gian (chẳng hạn giá đóng cửa của cổ phiếu) và được chia thành các đoạn liên tiếp có độ dài cố định. Mỗi đoạn đại diện cho một bối cảnh quá khứ mà mô hình sẽ sử dụng để dự đoán giá trị tiếp theo.

Khi dữ liệu đã được chuẩn hóa và chuyển đổi thành dạng chuỗi, mô hình LSTM được xây dựng bằng cách thêm một hoặc nhiều tầng LSTM. Mỗi tầng bao gồm nhiều cell, và trong mỗi cell đều chứa đầy đủ các thành phần cần thiết như forget gate, input gate, output gate và cell state. Các cổng này hoạt động dựa theo các phép tính đã được mô tả trong các nghiên cứu nền tảng của Hochreiter & Schmidhuber (1997) và Gers et al. (2000), cho phép mô hình lựa chọn thông tin nào cần quên, thông tin nào cần ghi nhớ, đồng thời duy trì trạng thái ký ức dài hạn để học được các quan hệ phụ thuộc xa trong chuỗi thời gian.

Sau khi đi qua các tầng LSTM, mô hình tạo ra đầu ra dưới dạng trạng thái ẩn cuối cùng. Giá trị này thường nằm trong một phạm vi đã được chuẩn hóa khi xử lý dữ liệu. Vì vậy, các giá trị dự đoán sẽ được chuyển ngược lại về thang đo ban đầu nhằm thu được giá trị thực tế của chuỗi dữ liệu, chẳng hạn như giá cổ phiếu tại thời điểm cần dự đoán.

## 4.6. Recurrent Neural Network (RNN)

### 4.6.1. Giới thiệu

Recurrent Neural Network (RNN) là một mô hình mạng neural hồi quy thuộc vào nhóm mô hình Deep Learning. RNN bao gồm 3 thành phần chính là Input layer, Hidden layer và Output layer.

A picture containing text, post-it note, screenshot, diagram

Description automatically generated

###### *Hình 4.8. 3 thành phần chính của RNN*

RNN có input đầu vào là một chuỗi các giá trị theo trình tự thời gian. Từng giá trị trong input sẽ được đi qua một hoặc nhiều lớp ẩn bên trong để tính toán giá trị cho biến đầu ra output tương ứng với giá trị nhập vào. Tuy nhiên, RNN khác với mạng neural thông thường là trạng thái hiện tại sẽ có sự phụ thuộc vào các trạng thái trước đó.

RNN có đầu vào và đầu ra là độc lập với nhau. Do đó mô hình này phù hợp với các bài toán dạng chuỗi như mô tả, hoàn thành câu, dịch máy, dự đoán chuỗi,… qua việc mô hình tổng hợp và lan truyền thông tin theo trình tự xâu chuỗi.

Mô hình được gọi là hồi quy vì thực hiện tính toán ở thời điểm hiện tại sẽ phụ thuộc vào các kết quả tính toán ở những thời điểm trước đó. Do đó, RNN phải nhớ thông tin đã được đưa vào để tính toán trước đó để phục vụ cho các bước tính toán sau này.

Sau nhiều bước biến đổi, các thông tin ở những bước đầu tiên bị biến đổi nhiều lần dẫn đến bị triệt tiêu. Chính vì thế, RNN chỉ có thể nhớ được một vài bước trước đó để mô hình được tối ưu nhất.

Kiến trúc của RNN [36]:

A picture containing screenshot, diagram, text, line

Description automatically generated

###### *Hình 4.9. Kiến trúc RNN*

Trong đó:

: Giá trị đầu vào tại thời điểm t.

: Giá trị đầu ra tại thời điểm t.

: Giá trị kích hoạt tại thời điểm t.

: Trọng số kết nối giữa hidden layer với chính nó.

: Trọng số kết nối giữa input và hidden layer.

: Trọng số kết nối giữa hidden layer với output.

: Vector điều chỉnh (bias) cho hidden layer.

: Các hàm kích hoạt.

### 4.6.2. Ý tưởng thực hiện mô hình

Ban đầu khởi tạo giá trị tại thời điểm t = 0 là một giá trị bất kì, thường sẽ khởi tạo bằng 0. Mô hình cũng sẽ được cho trước các giá trị , , , .

Với mỗi giá trị đầu vào tại thời điểm *t,* tiến hành tính toán các giá trị sau:

Các hàm kích hoạt sử dụng có thể kể đến như hàm sigmoid, hàm tanh, RELU,…

* Hàm sigmoid:
* Hàm tanh:
* Hàm RELU:

Khi đã tính toán được các giá trị dự đoán , thực hiện tính toán độ lỗi giữa giá trị dự đoán so với giá trị thực tế thông qua hàm mất mát đối với mỗi thời điểm *t*.

Kế đến tính tổng độ lỗi trong các thời điểm *t*, hay còn gọi là *Cost* thông qua công thức:

Giá trị *Cost* cho biết tổng độ lỗi giữa giá trị dự đoán với giá trị thực tế. Khi giá trị dự đoán càng gần giá trị thực tế, hay nói cách khác là giá trị dự đoán càng chính xác thì giá trị của *Cost* càng tiến gần về 0. Do vậy, phải thực hiện bước điều chỉnh để thay đổi các trọng số được khởi tạo ban đầu là , , , để mô hình cho ra được kết quả chính xác nhất. Phương pháp này gọi là Backward Propagation hay lan truyền ngược theo thời gian.

Đối với mỗi trọng số W và giá trị bias b, thực hiện việc lan truyền ngược và cập nhật lại giá trị của trọng số và của bias thông qua công thức sau:

Với:

* : Giá trị trọng số mới.
* : Giá trị trọng số hiện tại.
* : Giá trị bias mới.
* : Giá trị bias hiện tại.
* : Tốc độ học của mô hình (learning rate).

Sau khi đã tính toán và cập nhật lại các giá trị W và b mới, tiến hành tính toán lại các giá trị và thực hiện việc tính độ lỗi cũng như lặp lại quá trình lan truyền ngược cho đến khi mô hình đạt độ lỗi thấp nhất hoặc không còn xảy ra sự chênh lệch giữa giá trị dữ đoán và giá trị thực tế.

### 4.6.3. Xây dựng mô hình RNN trong dự đoán

Từ các bước thực hiện tính đoán giá trị đầu ra, có thể áp dụng mô hình RNN trong việc dự đoán qua những bước sau:

* Thu thập dữ liệu đầu vào: Dữ liệu đầu vào là dữ liệu có tính liên tục theo trình tự thời gian như giá cổ phiếu, chứng khoán,…
* Xây dựng mô hình RNN. Trong ngôn ngữ Python có hỗ trợ mô hình LSTM để xác định số lớp ẩn có trong mô hình.
* Huấn luyện dữ liệu. Trong quá trình huấn luyện, có thể tính toán hàm mất mát và thuật toán tối ưu để cập nhật lại các trọng số của mô hình để thu được mô hình với kết quả dự đoán tốt nhất.

## 4.7. Gradient Boosted Tree (GBT)

### 4.7.1. Giới thiệu

Gradient Boosted Tree (GBT) là một thuật toán machine learning, được sử dụng trong các tác vụ hồi quy và phân loại. Gradient boosting được diễn giải như một thuật toán tối ưu hóa dựa trên một hàm chi phí phù hợp, dựa trên quan sát của Leo Breiman [20]. Cụ thể, các thuật toán Gradient Boosting đã được phát triển dưới góc nhìn hạ gradient trong không gian hàm số - nghĩa là các thuật toán tối ưu hóa một hàm chi phí qua không gian hàm số bằng cách lựa chọn lặp đi lặp lại một hàm (giả thuyết yếu) chỉ vào hướng gradient âm.

Công thức chính của GBT:

Trong đó:

​ là mô hình mạnh tại bước thứ i, là mô hình cuối cùng được xây dựng từ tất cả các mô hình yếu trước đó.

là mô hình yếu tại bước thứ i, mô hình mới nhất được thêm vào chuỗi, được chọn để giảm thiểu lỗi dự đoán của mô hình mạnh hiện tại.

γ là tốc độ học, một tham số kiểm soát mức độ mà mô hình yếu mới ảnh hưởng đến mô hình mạnh, giúp kiểm soát tốc độ học của mô hình và giảm thiểu quá khớp.

### 4.7.3. Cách hoạt động của Grandient Boosted Trees:

Mô hình Grandient Boosted Trees có thể được thực hiện như sau:

**1. Khởi tạo mô hình:** GBT bắt đầu bằng một mô hình rất đơn giản, thường là một mô hình yếu. Gọi giá trị này là .

**2. Vòng lặp tạo mô hình:** vòng lặp này tiếp diễn đến khi đạt được số lượng mô hình yếu tôi đa (nếu có) hoặc khi độ lỗi dự đoán không giảm thêm nữa. Trong mỗi vòng lặp bao gồm:

**a. Tính độ lỗi:** Tính toán “độ lỗi” giữa dự đoán của mô hình hiện tại và giá trị thực tế. Độ lỗi là “gradient” mà chúng ta muốn “boost”. Độ lỗi này có thể được tính bằng cách lấy giá trị thực tế trừ đi giá trị dự đoán của mô hình.

Trong đó :

: là giá trị thực tế tại thời điểm dữ liệu thứ i.

: là giá trị dự đoán của mô hình GBT tại thời điểm dữ liệu thứ i.

: là độ của mô hình gradient boosted trees tại điểm dữ liệu thứ i.

**b. Xây dựng mô hình yếu:** Xây dựng một mô hình yếu để dự đoán độ lỗi. Mô hình yếu được chọn để giảm thiểu độ lỗi dự đoán của mô hình mạnh hiện tại. Có thể biểu diễn bằng công thức:

Trong đó :

: là giá trị thực tế.

: là giá trị dự đoán của mô hình mạnh hiện tại.

kà mô hình yếu.

: là hàm chi phí.

*argmin:* viết tắt của argument of the minium*.*

**c. Cập nhật mô hình mạnh:** Cập nhật mô hình mạnh bằng cách thêm mô hình yếu mới vào, điều chỉnh tốc độ học . Được biểu diễn bằng công thức:

**3. Dự đoán:** Kết quả cuối cùng là mô hình mạnh cuối cùng, được xây dựng từ nhiều mô hình yếu. Mô hình này có thể được dự đoán trên dữ liệu mới.

Điểm đặc biệt của Gradient Boosted Trees là sự cập nhật "gradient" theo hướng giảm thiểu độ lỗi, nghĩa là mô hình mới được tạo ra không chỉ để dự đoán độ lỗi mà còn giúp điều chỉnh hướng cập nhật của mô hình mạnh hiện tại.

**Ưu điểm:**

**Hiệu suất:** GBT thường đạt được hiệu suất cao trong nhiều bài toán học máy khác nhau, từ hồi quy đến phân loại. Chúng thường được coi là một trong những thuật toán học máy mạnh nhất.

**Khả năng xử lý dữ liệu phi tuyến tính:** Các mô hình dựa trên cây, bao gồm cả GBT, rất tốt trong việc xử lý các mô hình phi tuyến tính.

**Tính linh hoạt:** GBT cho phép sử dụng nhiều loại hàm mất mát, làm cho chúng có thể được tùy chỉnh cho nhiều loại bài toán khác nhau.

**Xử lý dữ liệu dạng hỗn hợp**: GBT có thể xử lý cả dữ liệu số và dữ liệu phân loại.

**Nhược điểm:**

**Thời gian huấn luyện:** GBT thường mất nhiều thời gian để huấn luyện so với các thuật toán khác, đặc biệt khi tập dữ liệu lớn và/hoặc có nhiều tính năng.

**Dễ overfitting:** Nếu không được cấu hình đúng, GBT có thể dễ dàng overfitting, tức là mô hình học quá nhiều từ dữ liệu huấn luyện và không tốt khi dự đoán dữ liệu mới.

**Khó hiểu:** Các mô hình phức tạp từ GBT khá khó hiểu và giải thích so với một số mô hình đơn giản hơn như cây quyết định.

**Cần tinh chỉnh tham số cẩn thận:** GBT yêu cầu tinh chỉnh tham số cẩn thận (như tốc độ học, số lượng cây tối đa, độ sâu cây,...) để đạt được hiệu suất tốt.

## 4.8. Hidden Markov Model (HMM)

### 4.8.1. Giới thiệu

Hidden Markov Model (HMM) là một mô hình thống kê được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, phân tích chuỗi DNA, xử lý ảnh. Trong đó, mô hình cho phép dự đoán trạng thái ẩn dựa trên các quan sát được từ thực tế.

Mô hình gồm 3 tham số chính chính:

Với:

* *A*: Ma trận xác suất chuyển đổi trạng thái.
* *B*: Ma trận xác suất quan sát (cung cấp xác suất bj của các trạng thái ẩn đối với quan sát Ot).
* : Phân phối xác suất ban đầu của các trạng thái ẩn.

Với số lượng trạng thái ẩn là *N*, ma trận xác suất chuyển đổi trạng thái *A* được biểu diễn là một ma trận vuông có kích thước NxN.

Với số lượng quan sát có thể quan sát được là *M*, ma trận xác suất quan sát *B* là một ma trận có kích thước NxM.

là xác suất ban đầu của các trạng thái ẩn, điều này có nghĩa sẽ biểu thị cho xác suất mà trạng thái ẩn có thể xảy ra tại thời điểm ban đầu, nghĩa là thời điểm t = 0. Do đó sẽ là vector có kích thước là Nx1.

Mô hình Hidden Markov nên được đặc trưng bởi 3 vấn đề cơ bản:

* Xác định xác suất của P(O|) với và *O*  được cung cấp.
* Tìm ra dãy trạng thái ẩn tốt nhất (*Q*) dựa vào và *O* được cung cấp.
* Học các tham số *A* và *B* từ chuỗi quan sát *O* và tập hợp các trạng thái trước đó.

Với đề bài đặt ra là dự đoán giá cổ phiếu dựa trên kỹ thuật phân tích chuỗi thời gian đầu vào, nhóm sẽ nghiên cứu về vấn đề tìm ra dãy trạng thái ẩn tốt nhất bằng mô hình HMM thông qua thuật toán Viterbi [30].

Thuật toán Viterbi đưa ra công thức cho việc tìm dãy trạng thái ẩn tốt nhất như sau:

; với

; với

Với:

* : Xác suất HMM đang ở trạng thái ẩn j sau khi quan sát được khoảng thời gian *t*.
* : Xác suất chuyển tiếp từ trạng thái ẩn *i* đến trạng thái ẩn *j*.
* : Xác suất của trạng thái ẩn *j* đối với quan sát *Ot*.
* : Trạng thái ẩn mà HMM đang ở sau khoảng thời gian *t*.

### 4.8.2. Ý tưởng thực hiện mô hình

Ở thời điểm t = 1, tại mỗi trạng thái ẩn, khởi tạo giá trị v1 bằng tích giữa xác suất của trạng thái đó tại thời điểm ban đầu với xác suất của trạng thái ẩn đối với quan sát tại thời điểm t = 1 và giá trị của bt1 bằng 0.

; với

; với

Tại các thời điểm t = 2 trở về sau, tiến hành tính toán giá trị với ý tưởng là tìm ra giá trị lớn nhất của tích giữa với xác suất chuyển đổi trạng thái từ trạng thái trước đó sang trạng thái hiện tại và xác suất của trạng thái đang xét đối với quan sát tại thời điểm hiện tại. Với ý tưởng tương tự, tính giá trị của bằng cách tìm ra vị trí của trạng thái trước đó có tích của 3 tham số trên là lớn nhất.

Sau khi thực hiện tính toán trên tất cả các thời điểm *t,* có thể tính toán được điểm số tốt nhất *P\** và chỉ số của trạng thái có giá trị tốt nhất *qT\** qua công thức sau:

*P\* =*

*qT\** *=*

Giá trị *P\** được tính bằng cách chọn giá trị lớn nhất trong tất cả các điểm số của các trạng thái ẩn ở thời điểm cuối cùng. *P\** mang ý nghĩa là điểm số hay giá trị của trạng thái ẩn có khả năng xảy ra cao nhất tại thời điểm cuối cùng.

Giá trị *qT\** là trạng thái ẩn có điểm số tốt nhất. Giá trị này được xác định bằng việc chọn ra trạng thái ẩn có giá trị tương ứng với giá trị lớn nhất trong các điểm số của tất cả các trạng thái ẩn ở thời điểm cuối cùng.

### 4.8.3. Xây dựng mô hình HMM trong dự đoán

Với mục đích dự đoán giá đóng cửa của cổ phiếu trong ngày, mô hình HMM sẽ được xây dựng và học các đặc trưng từ các thuộc tính có thể quan sát được về giá cổ phiếu trong một ngày là giá mở cửa, giá cổ phiếu cao nhất và thấp nhất trong ngày.

Giá trị quan sát của mô hình là một vector ba chiều biểu diễn thông tin cổ phiếu hàng ngày:

*Ot =*

*:= (fracChange, fracHigh, fracLow)* [10]

Ba giá trị quan sát của mô hình HMM lần lượt là:

* Sự thay đổi phân đoạn trong giá đóng cửa và giá mở cửa (*fracChange*).
* Sự thay đổi phân đoạn trong giá cao nhất với gái đóng cửa (*fr*acHigh).
* Sự thay đổi phân đoạn trong giá thấp nhất với gái đóng cửa (*fracLow*).

Kế đến, huấn luyện mô hình HMM với việc thiết lập số lượng trạng thái ẩn mặc định bằng 10 và giá trị quan sát là tập ba giá trị được tính ở bước trên.

Sử dụng phương pháp tối đa hóa hậu nghiệm (MAP) để kiểm tra mô hình sau khi huấn luyện. Với việc sử dụng dữ liệu từ *d* ngày trước đó và giá mở cửa của ngày thứ *d + 1* để tính toán được *fracChange* cho ngày đó, giá trị MAP của ngày thứ *d + 1* được tính như sau:

*= arg*

Sau quá trình tối đa hóa hậu nghiệm, giá đóng cửa được dự đoán bằng công thức:

*Close = Open x (1 + fracChange)*

Khi áp dụng mô hình HMM vào việc dự đoán giá đóng cửa, nhóm đã tối ưu hóa qua việc chia nhỏ *fracChange* thành các giá trị nằm trong khoảng *fracChange* cao nhất và thấp nhất của *d* ngày với số bước chia nhất định (*number of steps*). Sau đó, tiến hành tìm giá trị *number of steps* tối ưu nhất dựa trên độ đo MAE (Mean Absolute Error) trên tập dữ liệu validate. Bằng cách này, nhóm đã tìm ra được giá trị *number of steps* có thể làm giảm giá trị MAE và cải thiện được độ chính xác của mô hình trên tập dữ liệu validate.

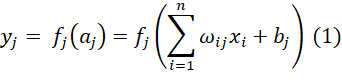
## 4.9. Deep Neural Network (DNN)

### 4.9.1. Giới thiệu

Theo Oyedele và đồng nghiệp (A. A. Oyedele, A. O. Ajayi, L. O. Oyedele, S. A. Bello and K. O. Jimoh, "Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction," in *Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction*, 2023), Deep Neural Network (DNN) là mô hình học sâu điển hình để học các biểu diễn dữ liệu phức tạp và trừu tượng theo cấp bậc. Quá trình học này truyền tải dữ liệu qua nhiều lớp biến đổi.

Kiến trúc của DNN gồm 3 tầng: Input layer, Hidden layer, Output layer. Trong đó mỗi lớp có nhiều đơn vị xử lý kết nối với nhau. Trong DNN, mỗi lớp sử dụng một phép biến đổi phi tuyến trên input của nó để tạo ra output. Hình dưới đây thể hiện kiến trúc DNN, là một đồ thị hướng không chu trình, có nghĩa là luồng thông tin chỉ diễn ra theo hướng tiến về phía trước, đơn giản hóa quá trình huấn luyện backpropagation.

Một trong những tác động của cấu trúc này là việc đầu ra từ các node trong lớp dưới tạo thành đầu vào của các node trong lớp phía trên là tính toán trong toàn bộ neural network cần diễn ra. Vì vậy, điều này có nghĩa là không thể cho mỗi node tính toán chức năng của nó trong lớp phía trên cho đến khi tất cả các tính toán đã hoàn thành trong lớp phía dưới.

Trong mỗi lớp DNN, đầu ra của mỗi node được tính toán bằng hàm kích hoạt , với trọng số . Số node trong hidden layer thứ i là . Các node được tính toán như sau:

Trong đó j∈[0,m] là chỉ số đại diện cho trong lớp l; i∈[0,n] là chỉ số đại diện cho một node đầu vào từ lớp l-1 ; n là tổng số node đầu vào của một layer liền kề; m là số node trong lớp l, với m,n∈Z^+. y\_j là đầu ra của node thứ j; x\_i là đầu vào từ node; ω\_ij là trọng số; b\_j là hệ số tự do (bias) (P. M. Bento, J. A. Pombo, M. R. Calado and S. J. Mariano, "Stacking ensemble methodology using deep learning and ARIMA models for short-term load forecasting," in *Energies*, 2021).

### 4.9.2. Diễn giải kiến trúc

Theo Kiến trúc của DNN được diễn giải chi tiết như sau:

**Input layer:**

- Lớp đầu tiên là lớp đầu vào. Lớp này không thực hiện bất kỳ tính toán nào.

- Giữ các giá trị của đầu vào.

**Hidden layer:**

- Số lượng lớp ẩn được tùy chỉnh phù hợp với bài toán.

- Trong một mô hình DNN, dense layer thường được sử dụng làm hidden layer (lớp ẩn) giữa lớp đầu vào và lớp đầu ra. Mỗi đơn vị (node) trong lớp dense kết nối với tất cả các đơn vị trong lớp trước đó và lớp sau đó, tạo thành một mạng neural hoàn chỉnh.

- Các dense layer trong mạng neural có sự kết nối đầy đủ, nghĩa là mỗi đơn vị trong dense layer kết nối với tất cả các đơn vị trong lớp trước đó. Điều này tạo ra một mạng neural có khả năng mô hình hóa các mối tương quan phức tạp trong dữ liệu. Các đầu vào của dense layer được nhân với các trọng số tương ứng và được tổng hợp lại để tạo ra đầu ra của mỗi đơn vị. Sau đó, đầu ra này thường được đưa qua một hàm kích hoạt phi tuyến để tạo ra đầu ra cuối cùng của lớp dense.

- Dense layer có khả năng học các mối quan hệ không tuyến tính giữa các đặc trưng trong dữ liệu. Với cách kết nối đầy đủ, dense layer có thể biểu diễn thông tin phức tạp và thực hiện các phép biến đổi phi tuyến trên dữ liệu.

**Output layer:**

- Lớp đầu ra.

- Không yêu cầu hàm kích hoạt phi tuyến

### 4.9.3. Xây dựng mô hình DNN trong dự đoán giá cổ phiếu

Thứ tự xây dựng mô hình DNN trong dự đoán giá cổ phiếu được trình bày như sau:

1. Dữ liệu: Dữ liệu đầu vào là bộ dữ liệu của ba doanh nghiệp Apple, Samsung và Xiaomi. Dữ liệu trước khi đưa vào mô hình sẽ được tiền xử lý, chuẩn hóa và chuyển đổi dữ liệu vào định dạng phù hợp cho đầu vào của mô hình. Dữ liệu sau xử lý được đưa vào mô hình là một tập hợp các đặc trưng đầu vào.

2. Dữ liệu đầu vào được truyền qua các lớp ẩn (hidden layers) của DNN. Mỗi lớp ẩn bao gồm nhiều đơn vị (nodes) gọi là neurons. Mỗi neuron trong lớp ẩn nhận đầu vào từ các neuron trong lớp trước đó và tính toán đầu ra dựa trên các trọng số và hàm kích hoạt.

3. Đầu ra của lớp ẩn được truyền tiếp đến lớp tiếp theo cho đến khi đạt đến lớp đầu ra (output layer). Với bài toán dự đoán giá đóng cửa cổ phiếu, lớp đầu ra sẽ là 1 node.

4. Quá trình huấn luyện và đánh giá: Mô hình được huấn luyện sử dụng các tham số tốt nhất được lựa chọn từ quá trình tối ưu hóa tham số. Quá trình thực nghiệm trên mô hình DNN sử dụng ý tưởng của Grid Search để tìm ra các tham số mang lại hiệu quả tốt nhất. Bài toán dự đoán giá cổ phiếu sử dụng mô hình DNN sẽ thực hiện tối ưu hóa các tham số bao gồm: Số lượng hidden layers, số node trong hidden layers, activation function và learning rate. Mean Square Error (MSE) được sử dụng để đánh giá độ chính xác của mô hình trong quá trình huấn luyện mô hình DNN.

5. Đánh giá hiệu suất mô hình: Mô hình sau huấn luyện được đánh giá trên tập dữ liệu kiểm tra với các độ đo: RMSE, MAPE, MLSE.

6. Ứng dụng trong dự đoán tương lai: Ứng dụng mô hình DNN trong dự đoán giá đóng cửa cổ phiếu 30 ngày tiếp theo.

## 4.10. Các độ đo đánh giá mô hình

Để đánh giá độ chính xác của mô hình dự báo chuỗi thời gian, đề tài sử dụng ba chỉ số phổ biến là:

* Sai số phần trăm tuyệt đối trung bình (MAPE).
* Căn bậc hai của sai số bình phương trung bình (RMSE).
* Sai số logarit bình phương trung bình (MSLE).

Các chỉ số này phản ánh mức độ chênh lệch giữa giá trị dự báo và giá trị thực tế theo những cách khác nhau.

### 4.10.1. Sai số phần trăm tuyệt đối trung bình (MAPE – Mean Absolute Percentage Error)

MAPE đo lường **tỷ lệ phần trăm sai lệch trung bình tuyệt đối** giữa giá trị dự đoán và giá trị thực tế.

Công thức như sau [37]:

Trong đó: *n* là số lượng quan sát, là giá trị thực tế thứ i, là giá trị dự đoán thứ i.

Chỉ số này đánh giá độ chính xác tổng thể (accuracy) của mô hình trong dự báo. Giá trị MAPE càng nhỏ cho thấy mô hình dự báo càng chính xác.

### 4.10.2. Căn bậc hai của sai số bình phương trung bình (RMSE – Root Mean Squared Error)

RMSE là căn bậc hai của trung bình bình phương sai số, thể hiện mức độ sai lệch tuyệt đối trung bình giữa giá trị thực tế và giá trị dự báo.

Công thức như sau [38]:

Trong đó: *n* là số lượng quan sát, là giá trị thực tế thứ i, là giá trị dự đoán thứ i.

Chỉ số này đánh giá mức độ ổn định và độ tin cậy (stability, consistency) của mô hình trong dự báo. Giá trị RMSE càng nhỏ chứng tỏ mô hình càng ổn định.

### 4.10.3. Sai số logarit bình phương trung bình (MSLE – Mean Squared Logarithmic Error)

MSLE đánh giá mức độ chênh lệch tỷ lệ tương đối giữa giá trị thực tế và giá trị dự báo, thông qua thang đo logarit.

Công thức như sau [39]:

Trong đó: *n* là số lượng quan sát, là giá trị thực tế thứ i, là giá trị dự đoán thứ i, là hàm logarit tự nhiên.

Chỉ số đo mức độ khác biệt về tỷ lệ tăng/giảm giữa giá trị dự báo và thực tế, giúp đánh giá khả năng nắm bắt xu hướng thay đổi (trend sensitivity / growth pattern) của mô hình trong dự báo. Giá trị MSLE càng nhỏ chứng tỏ mô hình dự báo càng sát với xu hướng tăng/giảm và tốc độ thay đổi thực tế của dữ liệu.

**Tổng kết:**

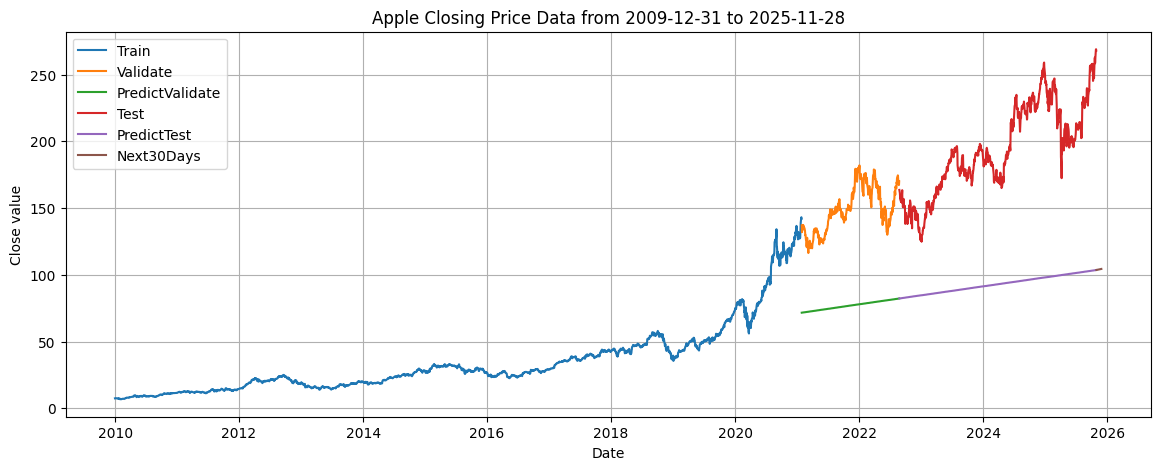
Việc kết hợp đồng thời ba độ đo MAPE, RMSE và MSLE giúp đánh giá mô hình một cách toàn diện hơn: MAPE thể hiện độ chính xác tổng thể, RMSE thể hiện mức độ ổn định và độ tin cậy, còn MSLE thể hiện khả năng nắm bắt xu hướng tăng trưởng của dữ liệu. Khi cả ba chỉ số đều nhỏ, mô hình được xem là có hiệu suất dự báo tốt.

# KẾT QUẢ THỰC NGHIỆM

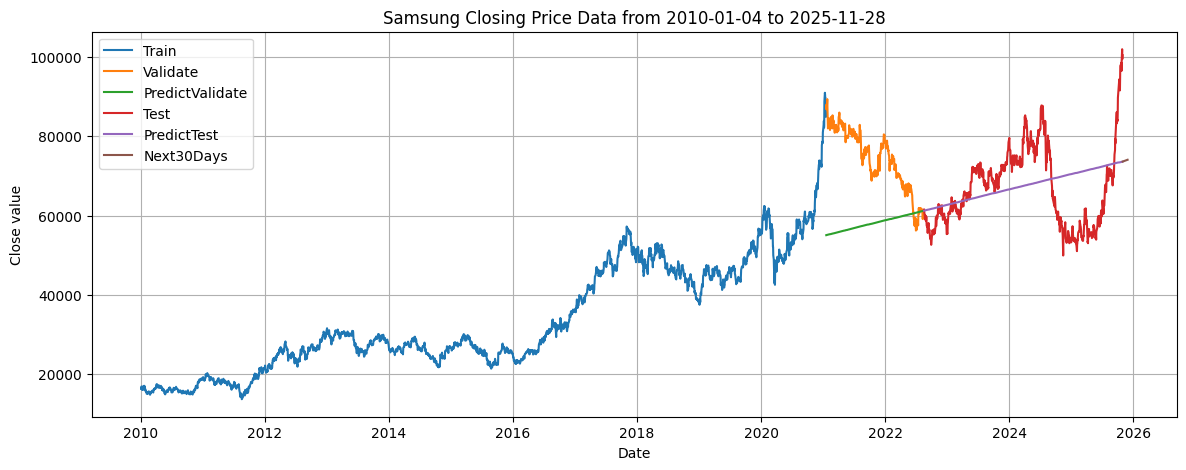
Chương 5 sẽ trình bày các kết quả thực nghiệm trên từng mô hình và phương pháp đã được trình bày tại chương 4. Các kết quả thực nghiệm được biểu diễn trên ba bộ dữ liệu với hai tỉ lệ 7:1:2 và 6:2:2. Ngoài ra, chương 5 còn thực hiện đánh giá hiệu suất mô hình dựa trên ba độ đo: MAPE, RMSE, MLSE, đồng thời lựa chọn hai mô hình có hiệu suất tốt nhất tiến hành dự đoán giá đóng cửa của các cổ phiếu trên các tập dữ liệu trong các phiên giao dịch của 30 ngày tiếp theo ngoài tập dữ liệu.

## 5.1. Linear Regression

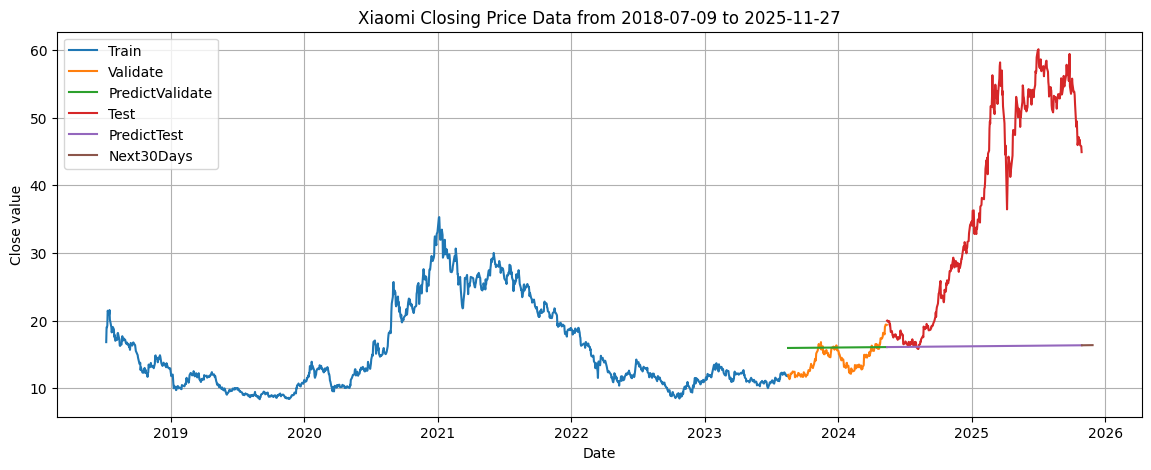
### 5.1.1. Tỉ lệ 7:1:2



###### *Hình 5.1. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình Linear Regression tỷ lệ 7:1:2*

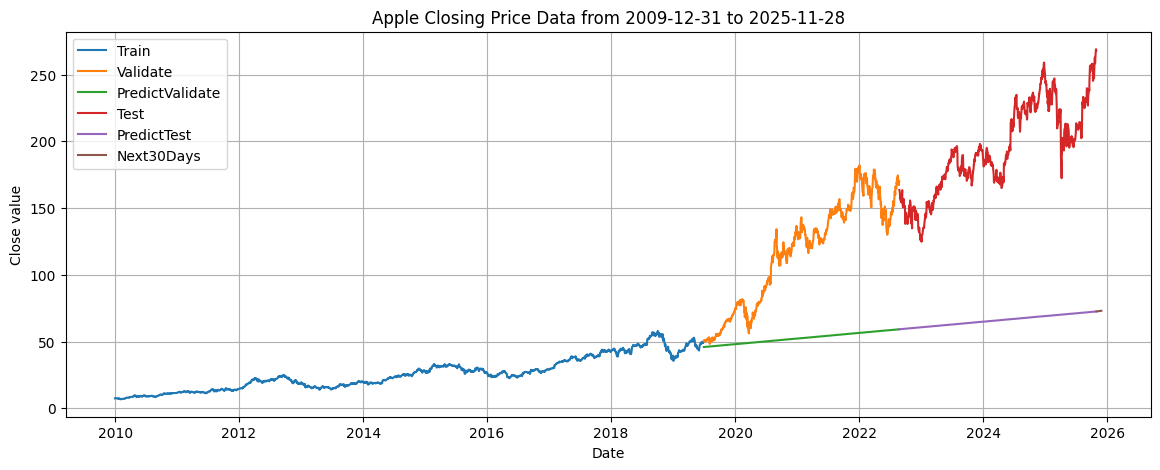


###### *Hình 5.2. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình Linear Regression tỷ lệ 7:1:2*

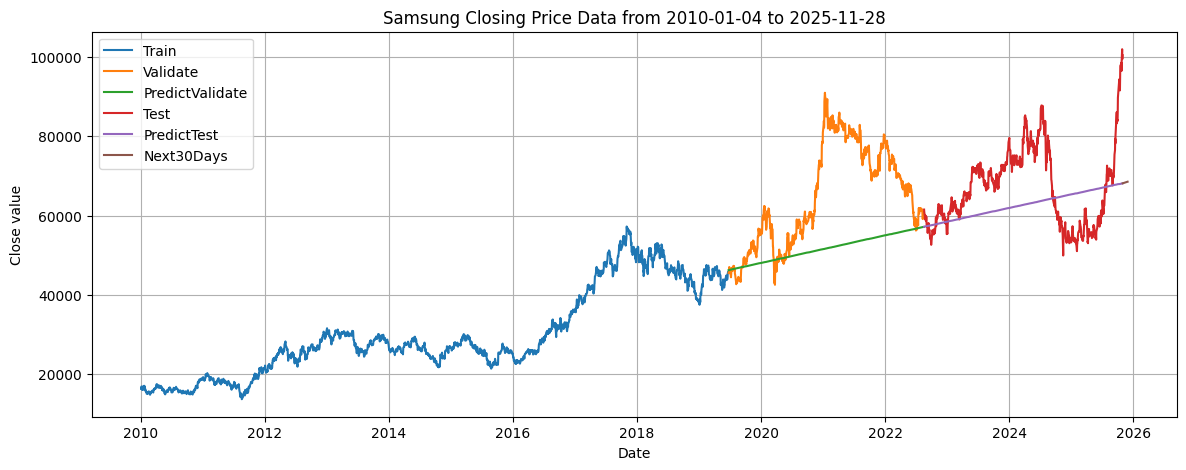


###### *Hình 5.3. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình Linear Regression tỷ lệ 7:1:2*

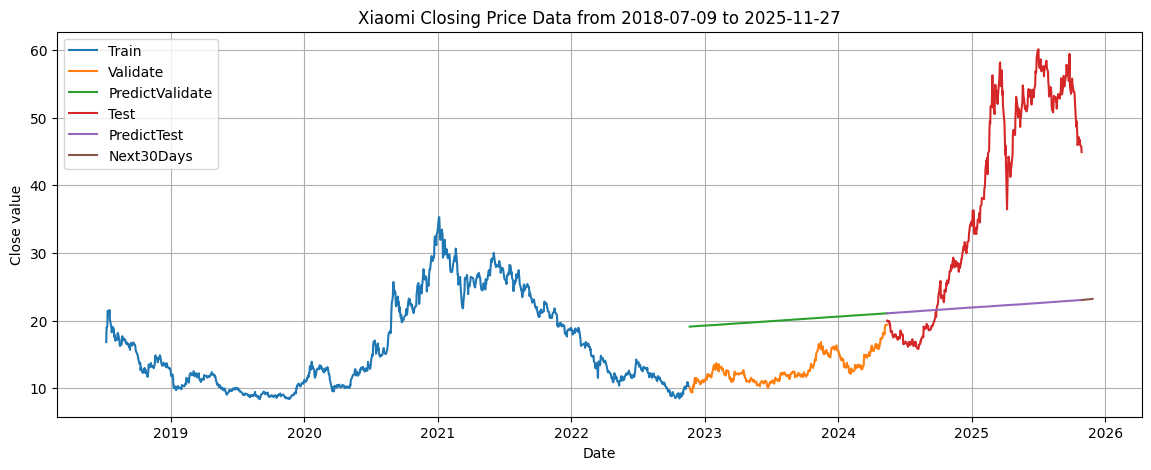
### 5.1.2. Tỉ lệ 6:2:2



###### *Hình 5.4. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình Linear Regression tỷ lệ 6:2:2.*



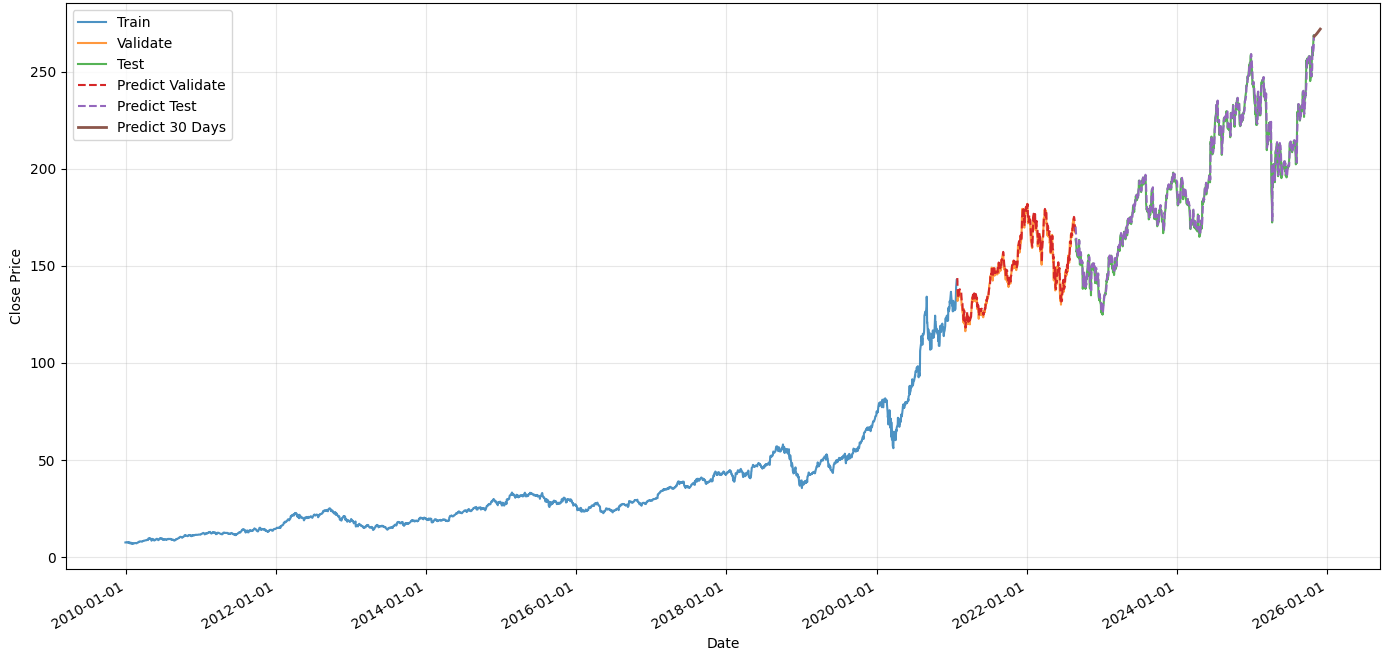
###### *Hình 5.5. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình Linear Regression tỷ lệ 6:2:2*



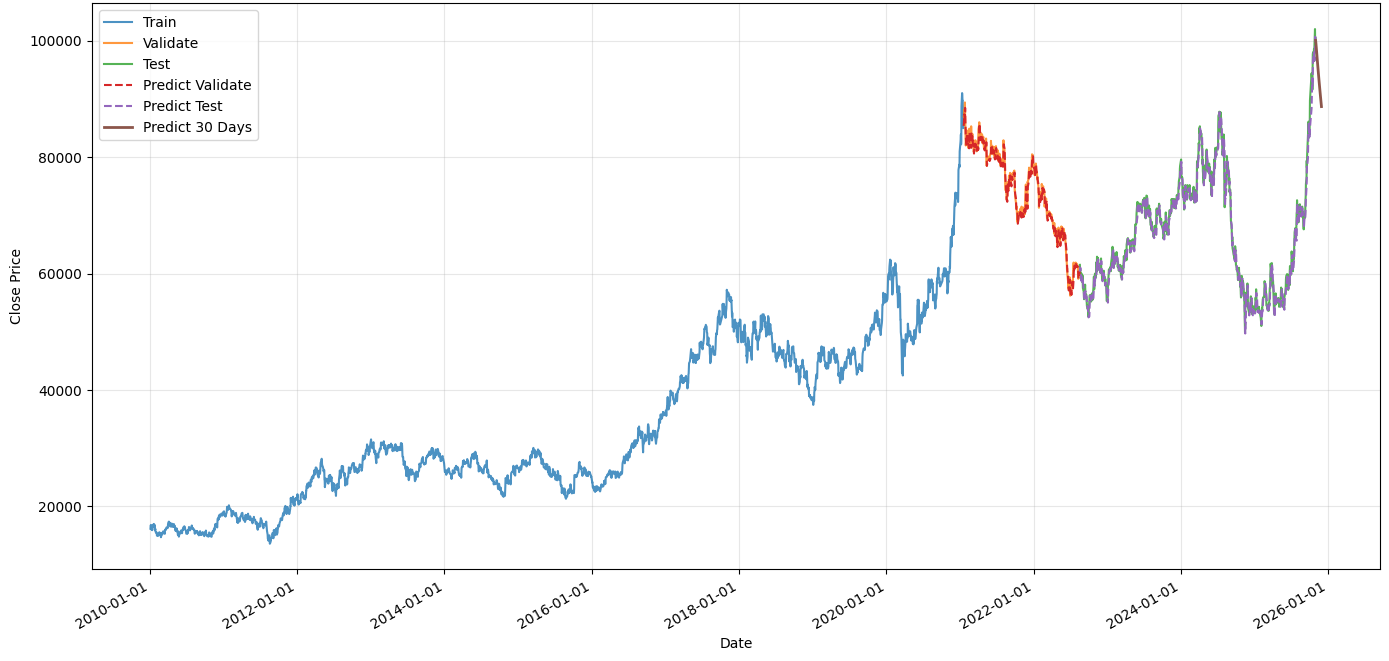
###### *Hình 5.6. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình Linear Regression tỷ lệ 6:2:2*

## 5.2. Gate Recurrent Unit (GRU)

### 5.3.1. Tỉ lệ 7:1:2

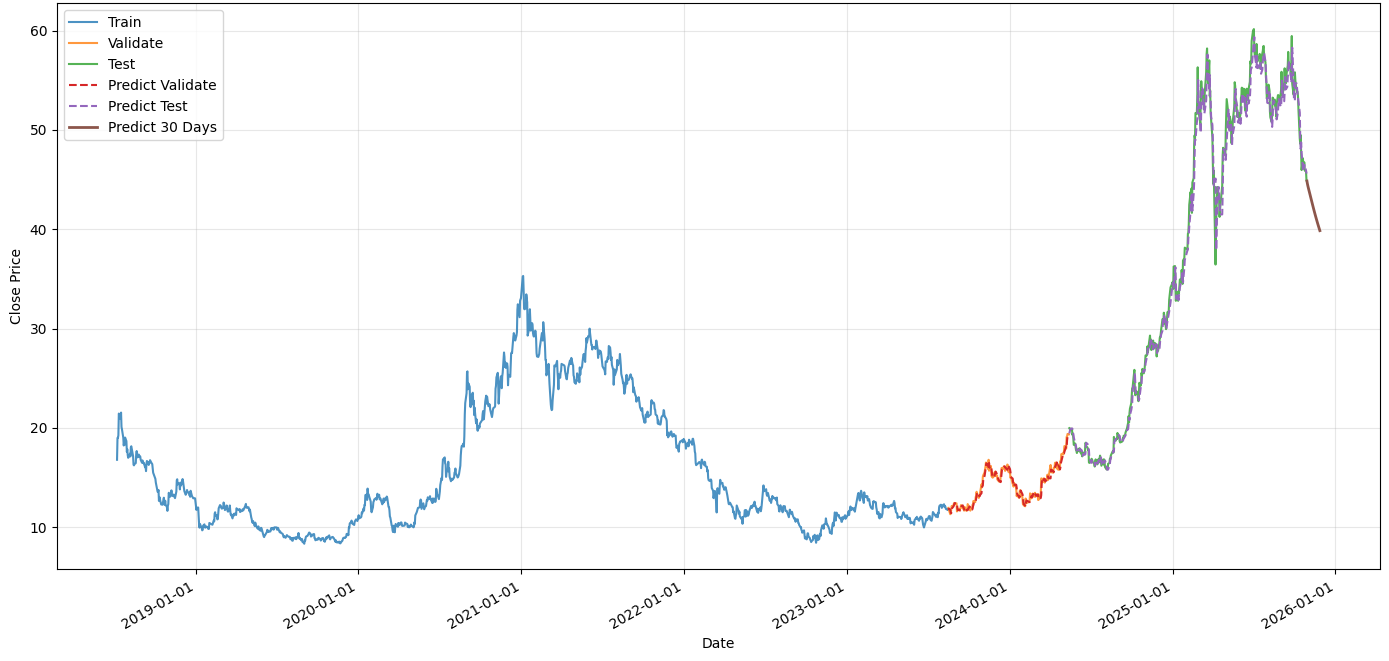


###### *Hình 5.7. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 7:1:2*



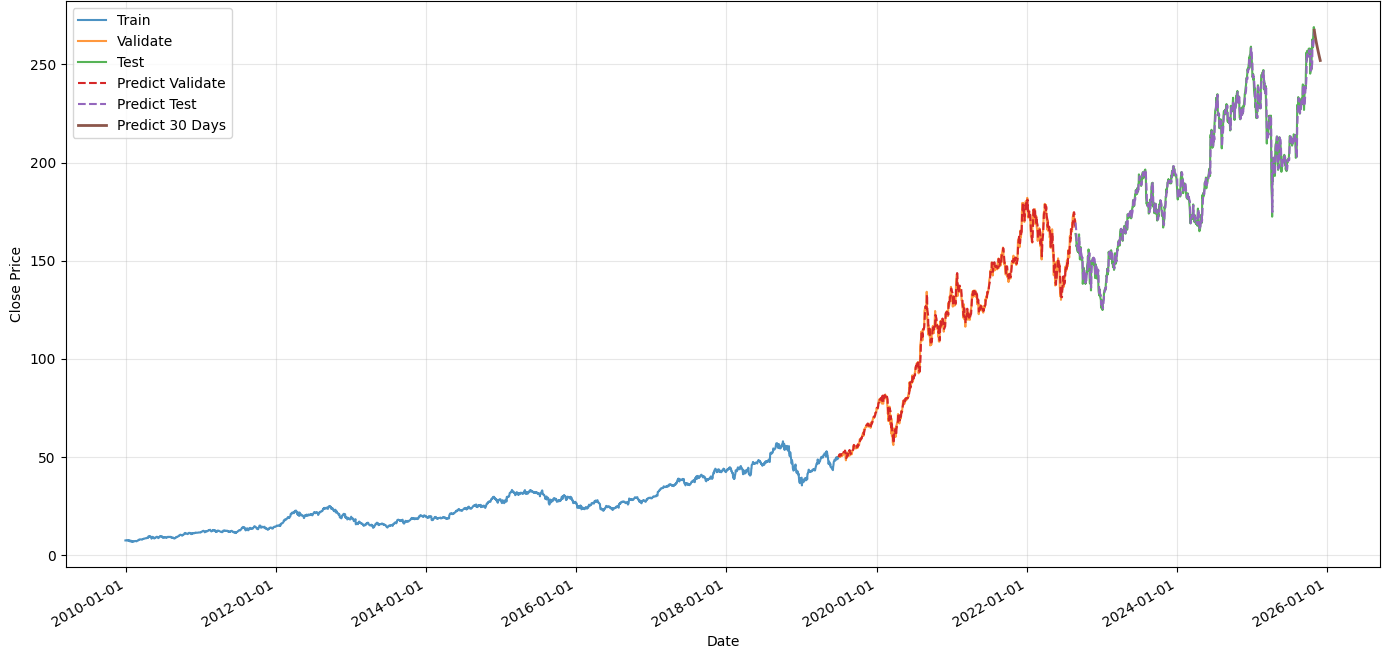
###### *Hình 5.8. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 7:1:2*

###### 



###### *Hình 5.9. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 7:1:2*

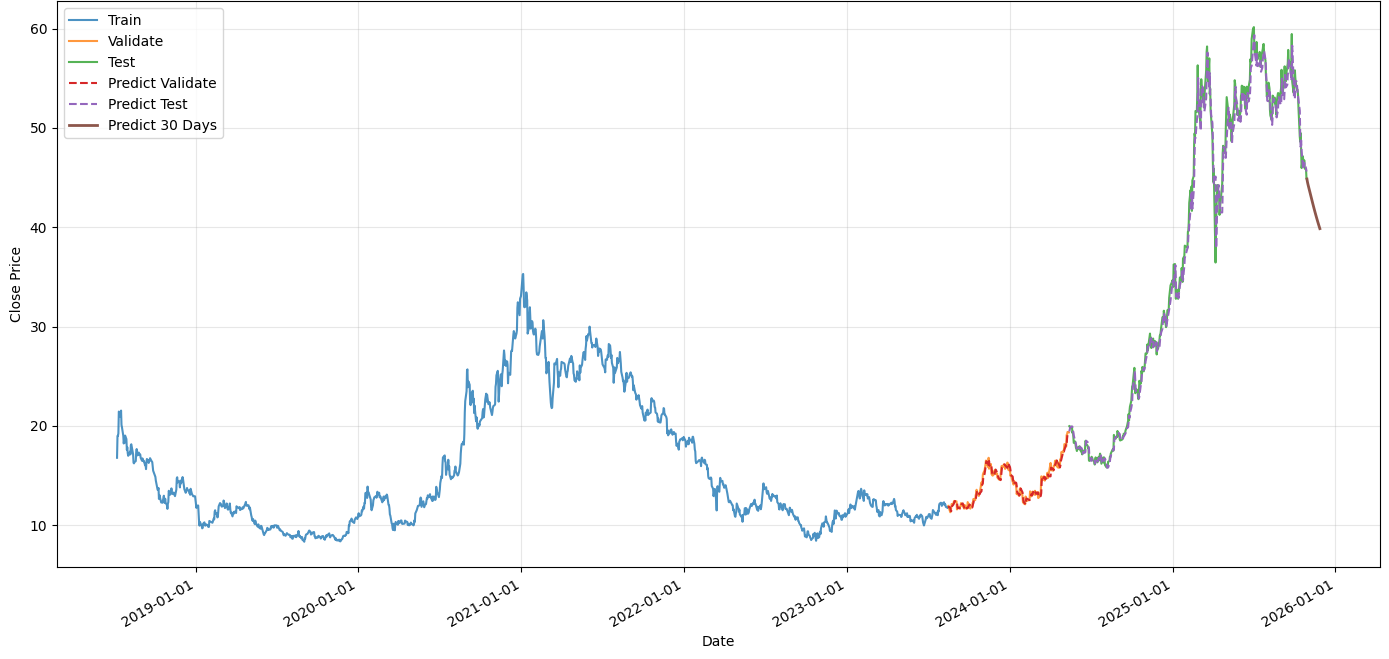
### 5.3.2. Tỉ lệ 6:2:2



###### *Hình 5.10. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 6:2:2*



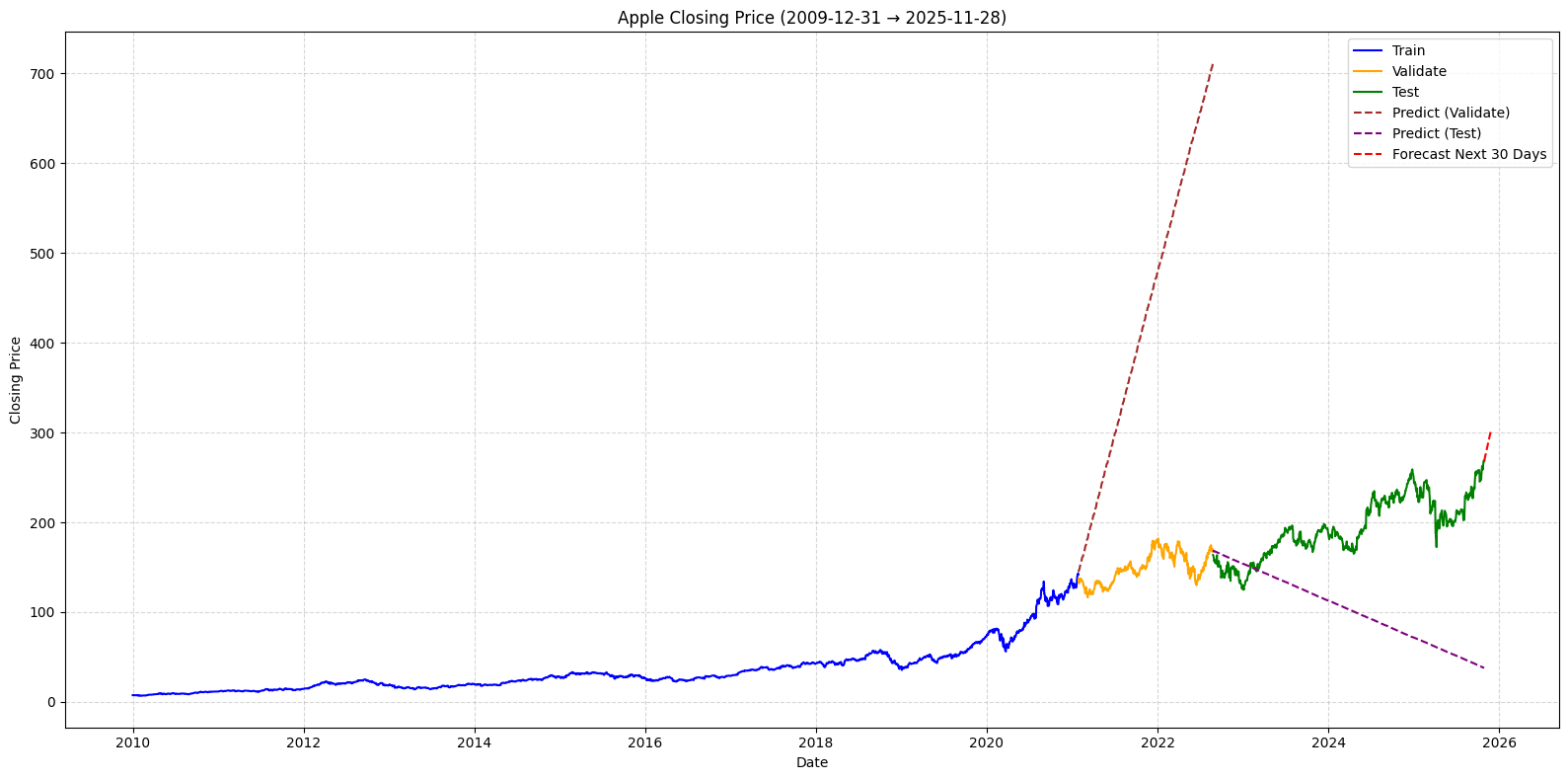
###### *Hình 5.11. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 6:2:2*



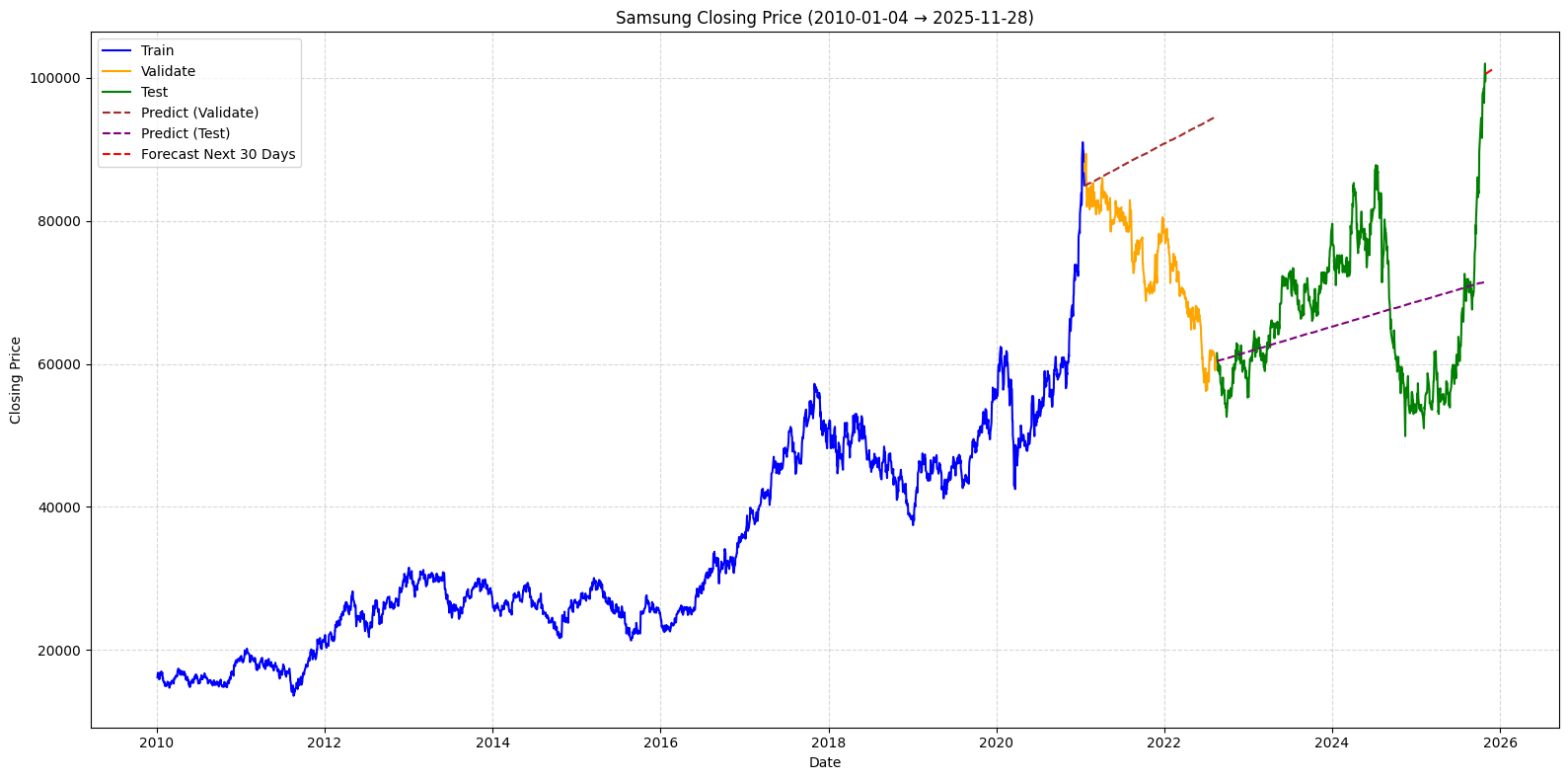
###### *Hình 5.12. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình Gated Recurrent Unit tỷ lệ 6:2:2*

## 5.3. Autoregressive Integrated Moving Average (ARIMA)

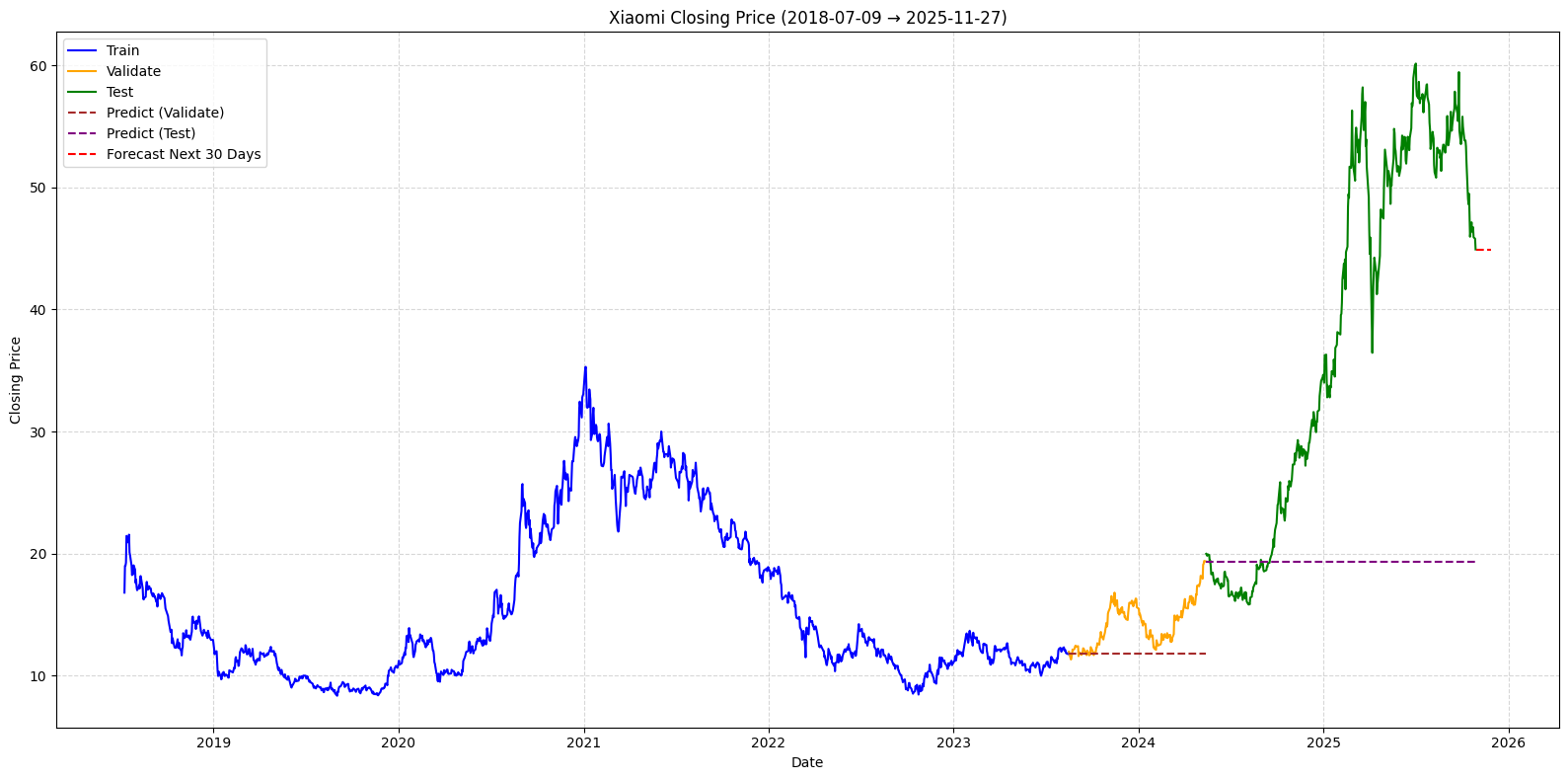
### 5.3.1. Tỉ lệ 7:1:2



###### *Hình 5.13. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình ARIMA tỷ lệ 7:1:2*

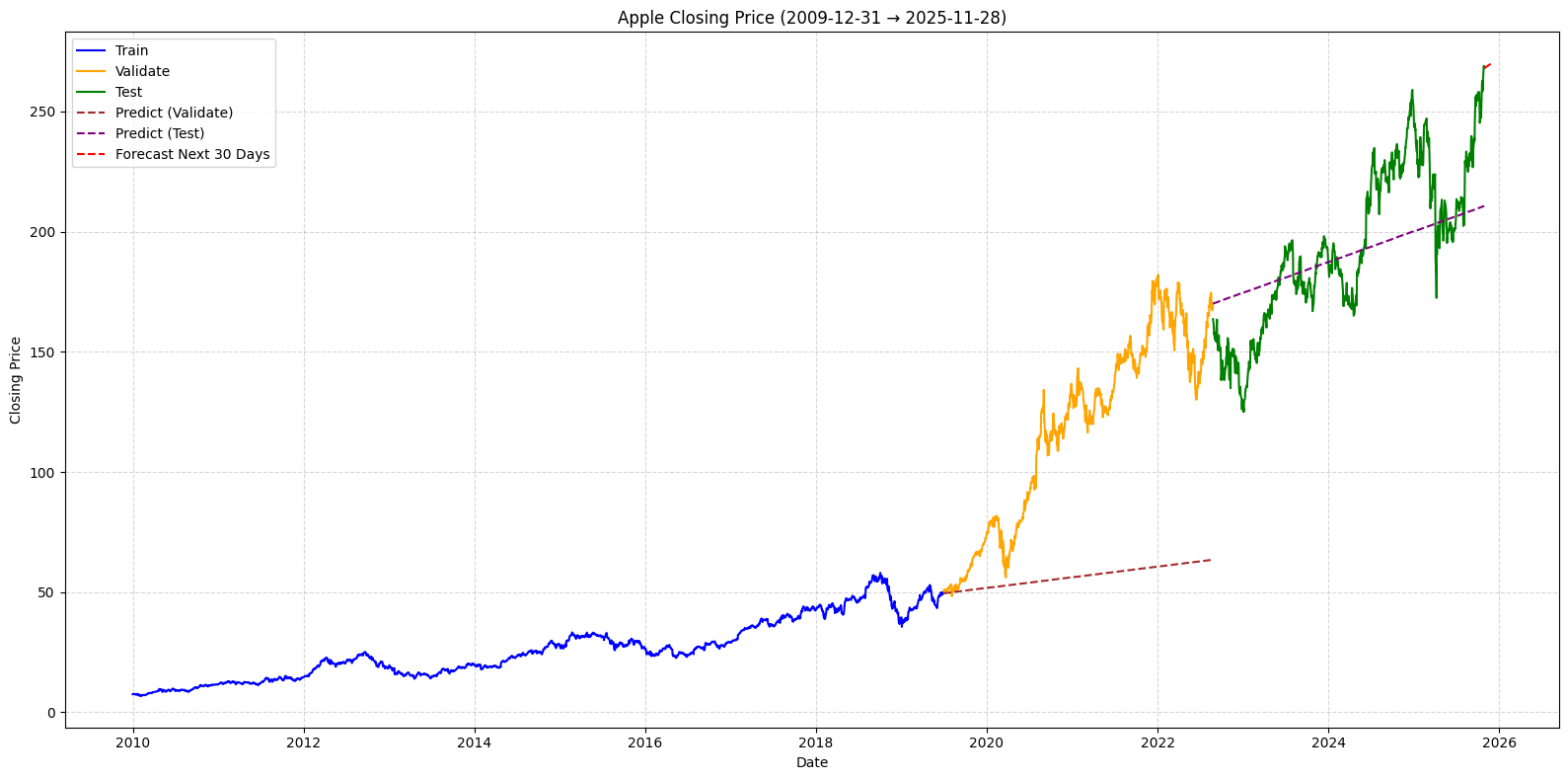


###### *Hình 5.14. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình ARIMA tỷ lệ 7:1:2*



###### *Hình 5.15. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình ARIMA tỷ lệ 7:1:2*

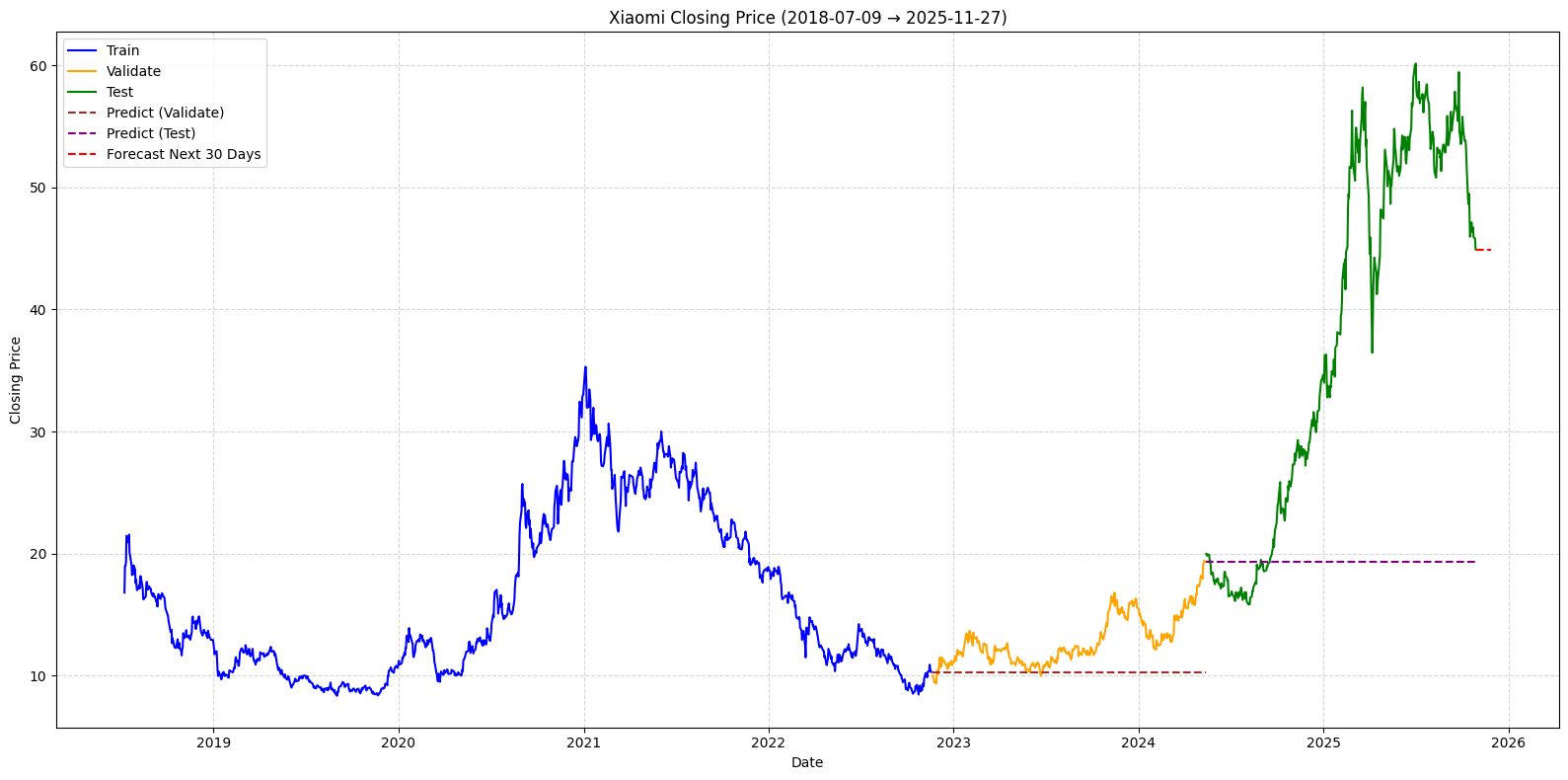
### 5.3.2. Tỉ lệ 6:2:2



###### *Hình 5.16. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình ARIMA tỷ lệ 6:2:2*



###### *Hình 5.17. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình ARIMA tỷ lệ 6:2:2*



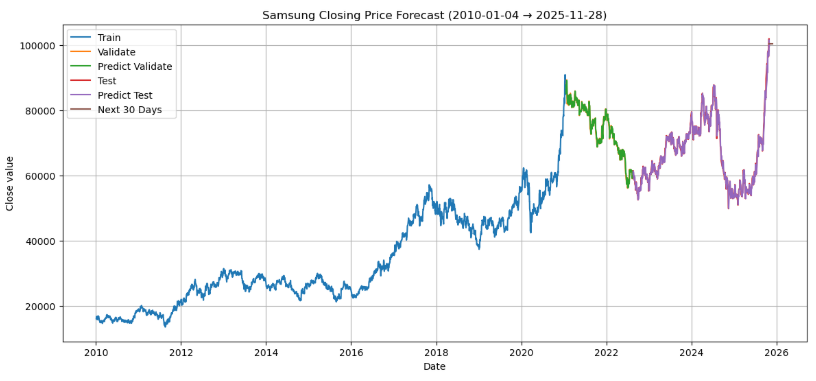
###### *Hình 5.18. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình ARIMA tỷ lệ 6:2:2*

## 5.4. Exponential Smoothing (ETS)

### 5.4.1. Tỉ lệ 7:1:2



###### *Hình 5.19. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình ETS tỷ lệ 7:1:2*

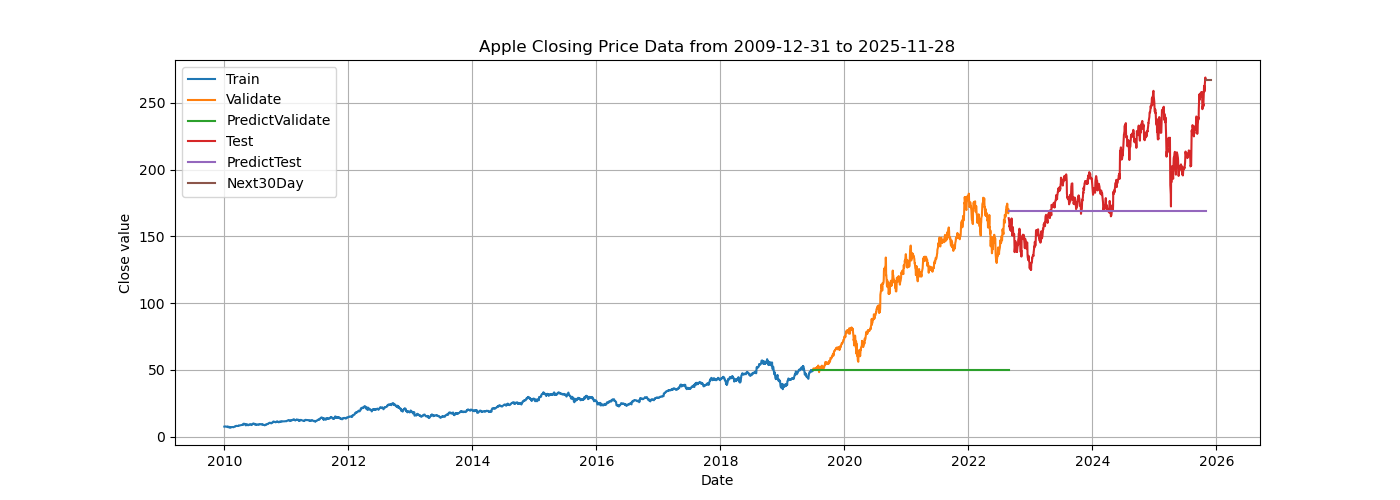


###### *Hình 5.20. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình ETS tỷ lệ 7:1:2*

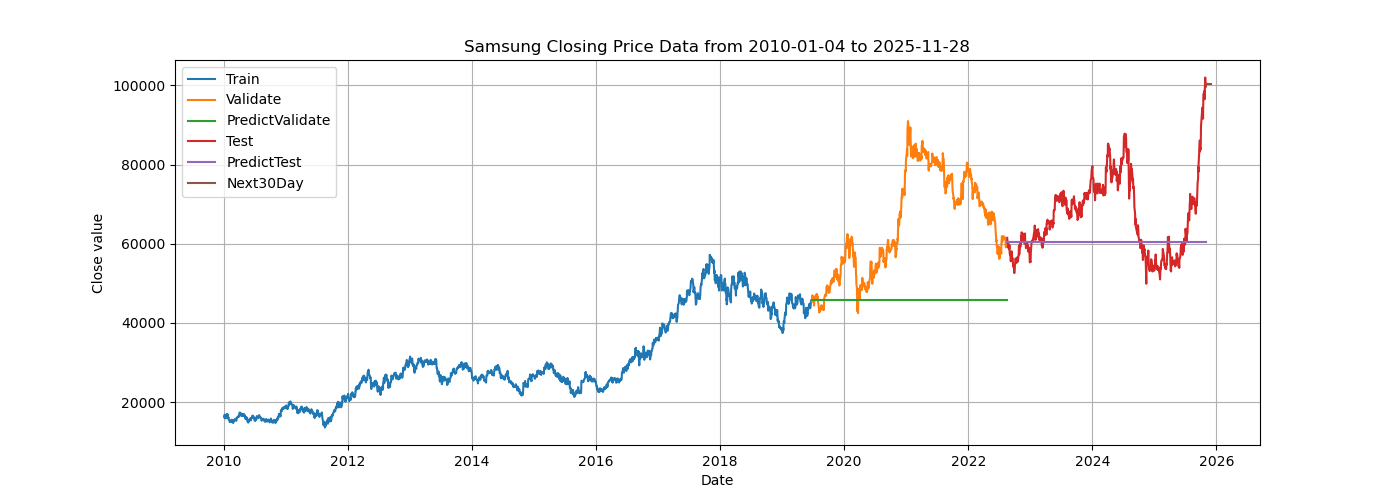


###### *Hình 5.21. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình ETS tỷ lệ 7:1:2*

### 5.4.2. Tỉ lệ 6:2:2



###### *Hình 5.22. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình ETS tỷ lệ 6:2:2*



###### *Hình 5.23. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình ETS tỷ lệ 6:2:2*



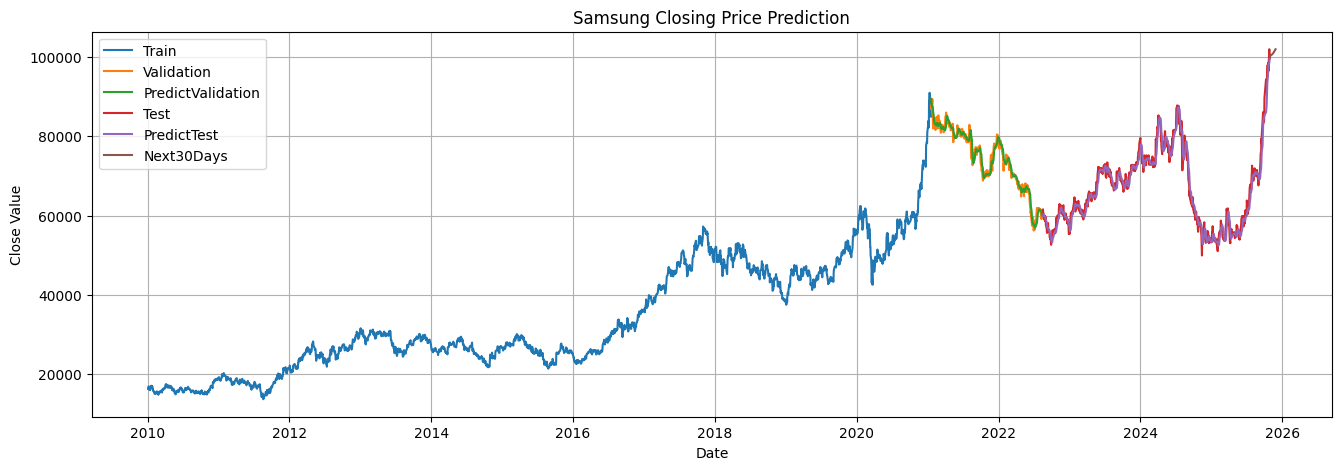
###### *Hình 5.24. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình ETS tỷ lệ 6:2:2*

## 5.5. Long Short-Term Memory (LSTM)

### 5.5.1. Tỉ lệ 7:1:2



###### *Hình 5.25. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình LSTM tỷ lệ 7:1:2*

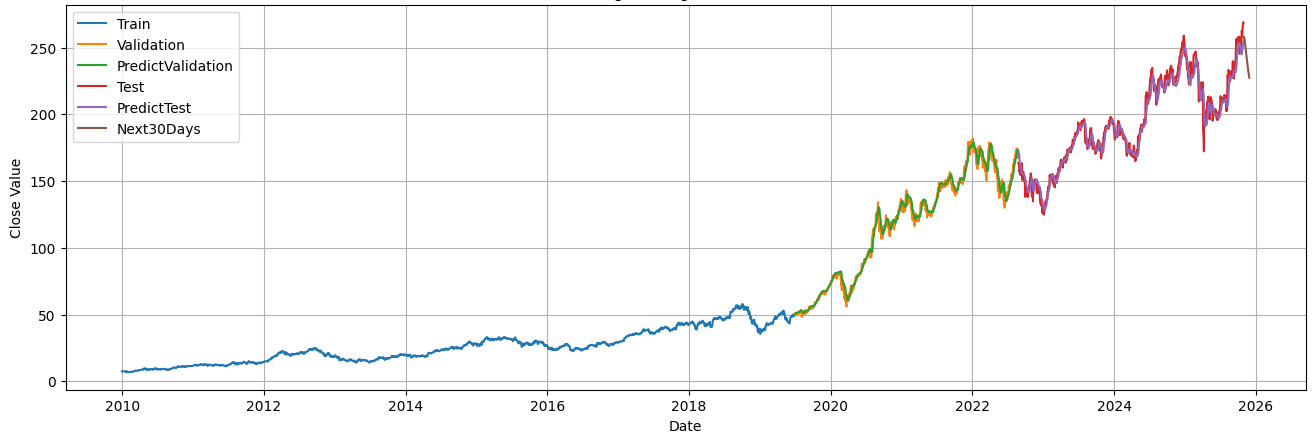


###### *Hình 5.26. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình LSTM tỷ lệ 7:1:2*

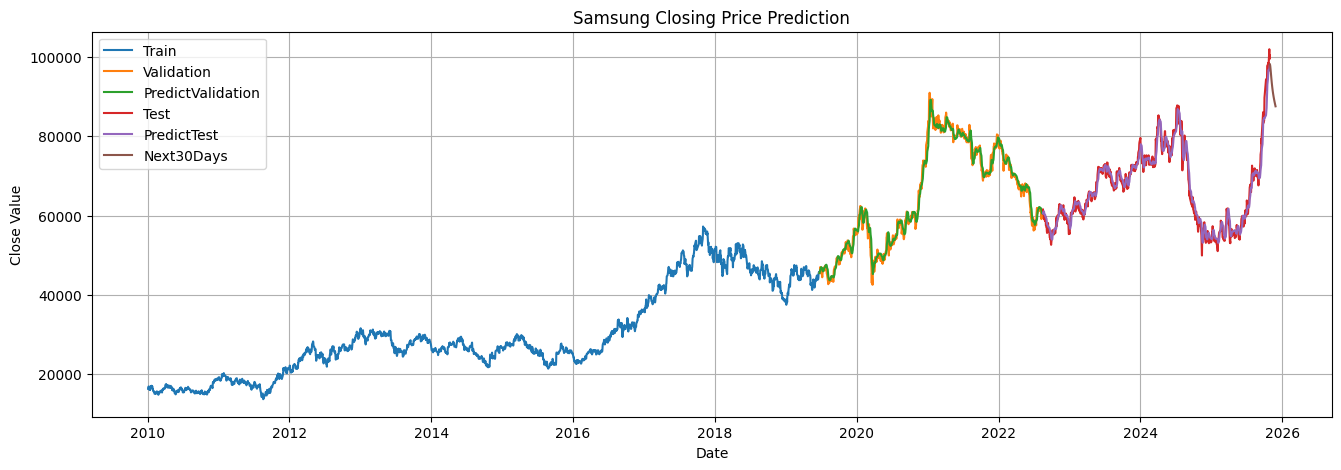


###### *Hình 5.27. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình LSTM tỷ lệ 7:1:2*

### 5.5.2. Tỉ lệ 6:2:2



###### *Hình 5.28. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình LSTM tỷ lệ 6:2:2*



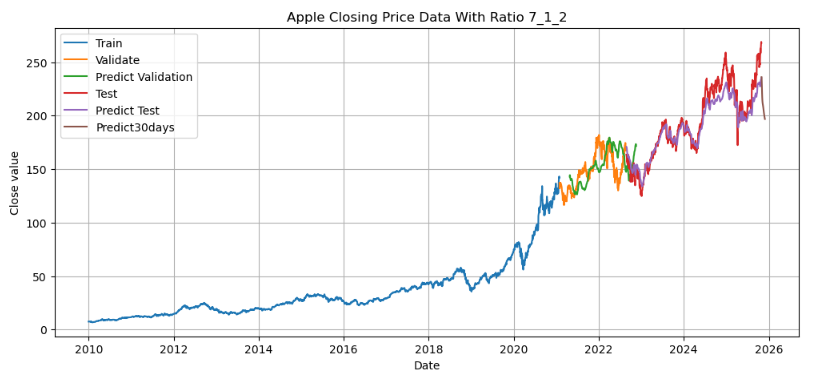
###### *Hình 5.29. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình LSTM tỷ lệ 6:2:2*



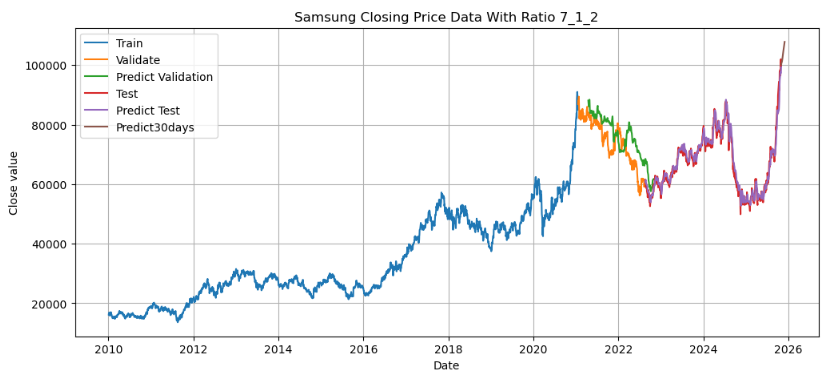
###### *Hình 5.30. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình LSTM tỷ lệ 6:2:2*

## 5.6. Recurrent Neural Network (RNN)

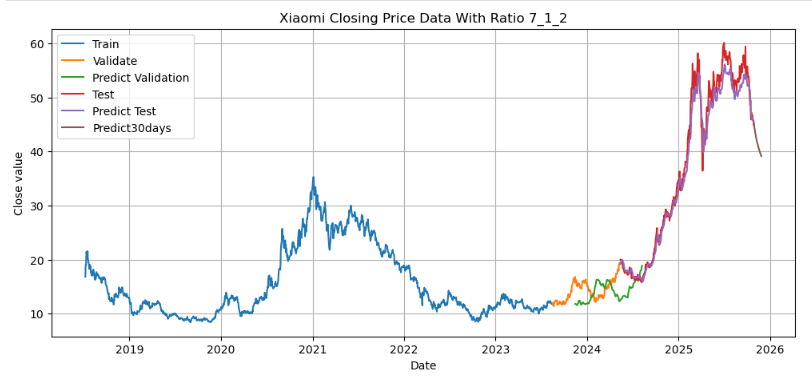
### 5.6.1. Tỉ lệ 7:1:2



###### *Hình 5.31. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình RNN tỷ lệ 7:1:2*

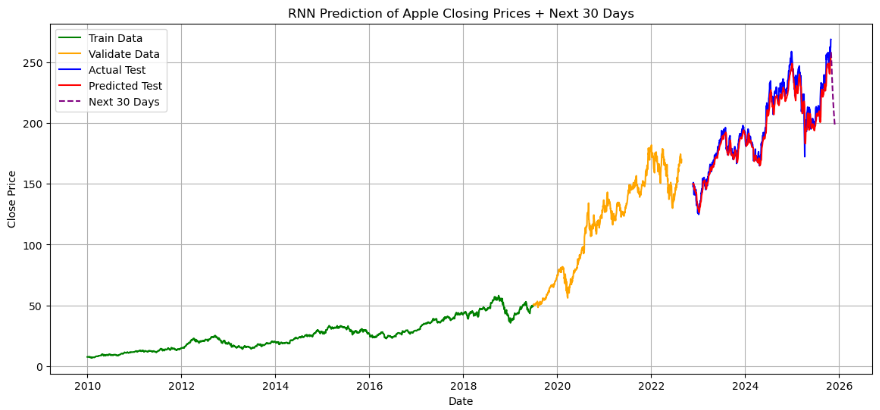


###### *Hình 5.32. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình RNN tỷ lệ 7:1:2*

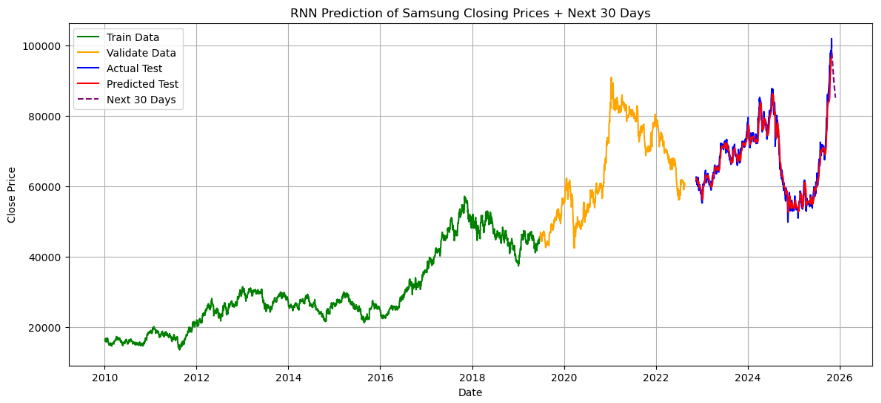


###### *Hình 5.33. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình RNN tỷ lệ 7:1:2*

### 5.6.2. Tỉ lệ 6:2:2



###### *Hình 5.34. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình RNN tỷ lệ 6:2:2*



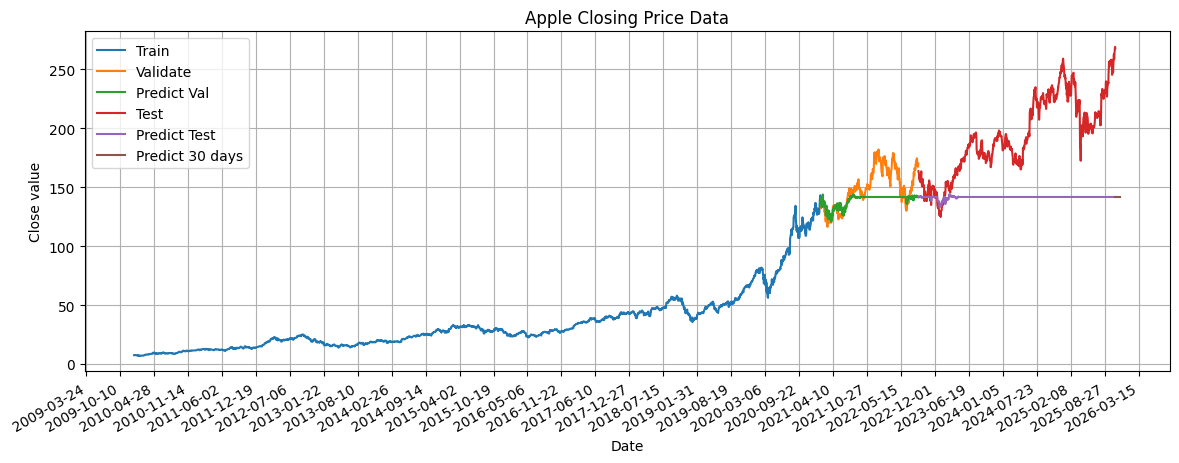
###### *Hình 5.35. Biểu đồ giá đóng cửa của Samsung từ 04/01/2010 đến 28/11/2025 trên mô hình RNN tỷ lệ 6:2:2*



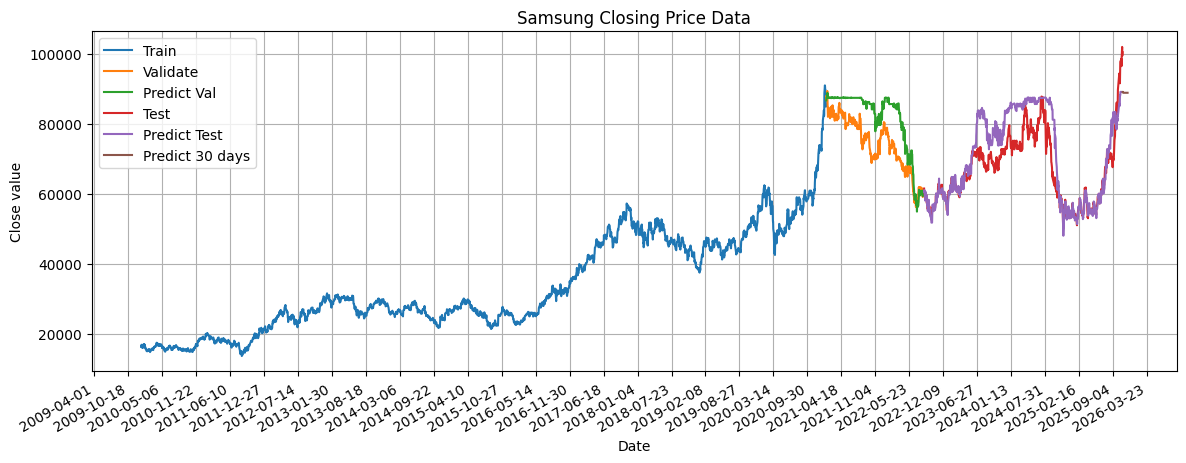
###### *Hình 5.36. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình RNN tỷ lệ 6:2:2*

## 5.7. Gradient Boosted Tree (GBT)

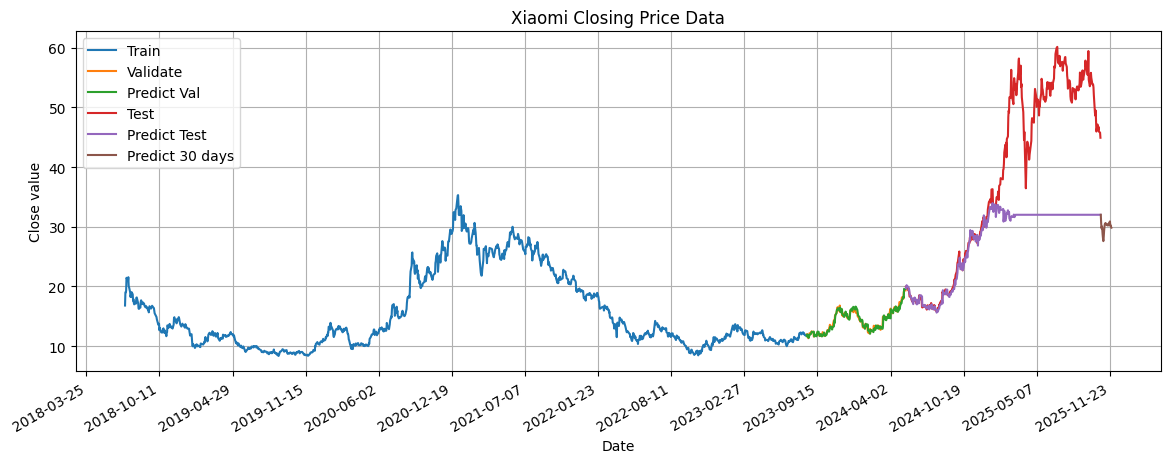
### 5.7.1. Tỉ lệ 7:1:2



###### *Hình 5.37. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình GBT tỷ lệ 7:1:2*

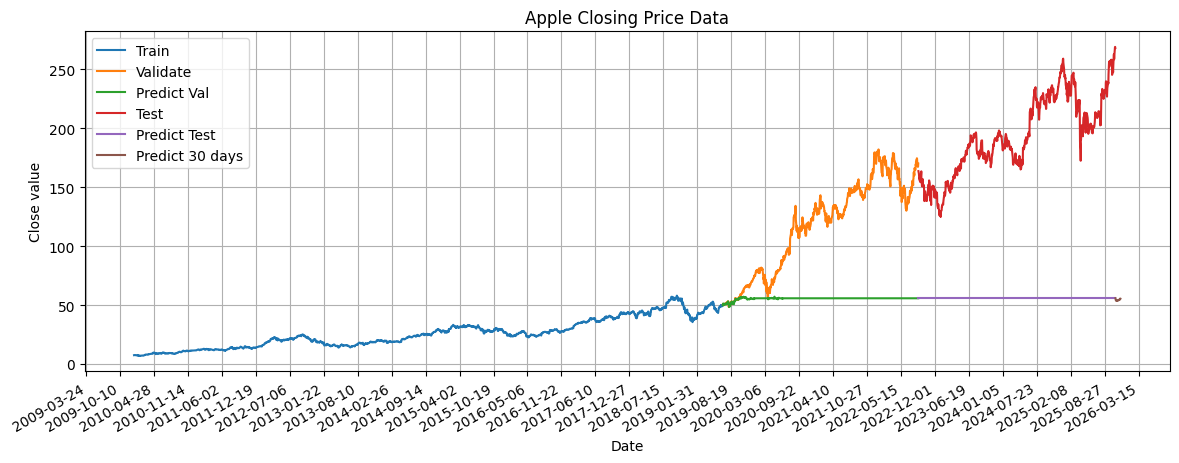


###### *Hình 5.38. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình GBT tỷ lệ 7:1:2*

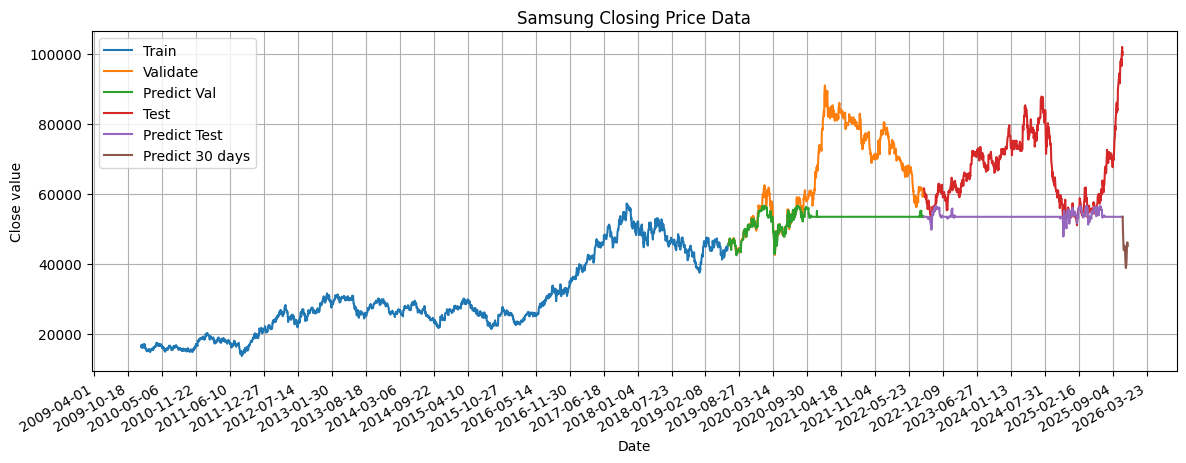


###### *Hình 5.39. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình GBT tỷ lệ 7:1:2*

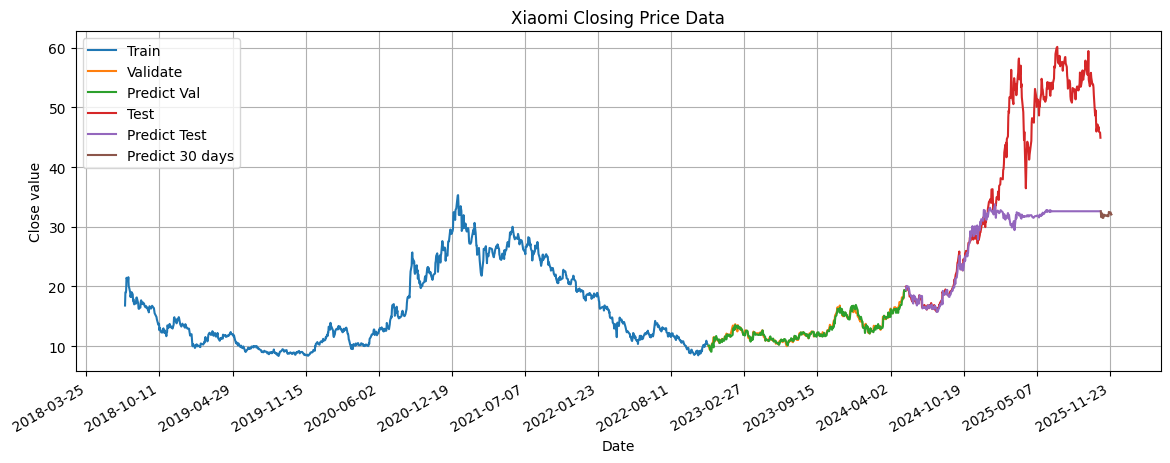
### 5.7.2. Tỉ lệ 6:2:2



###### *Hình 5.40. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình GBT tỷ lệ 6:2:2*



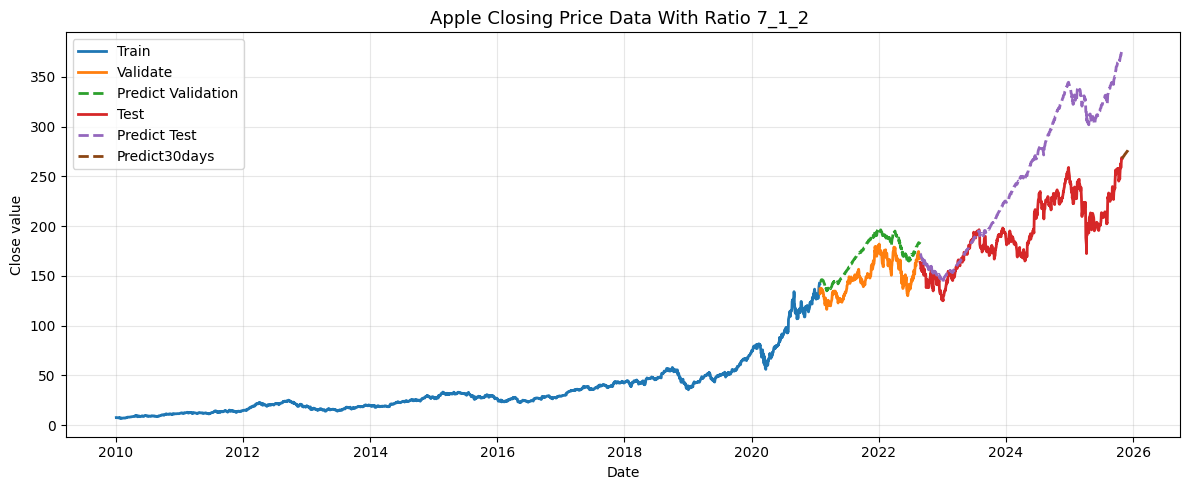
###### *Hình 5.41. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình GBT tỷ lệ 6:2:2*



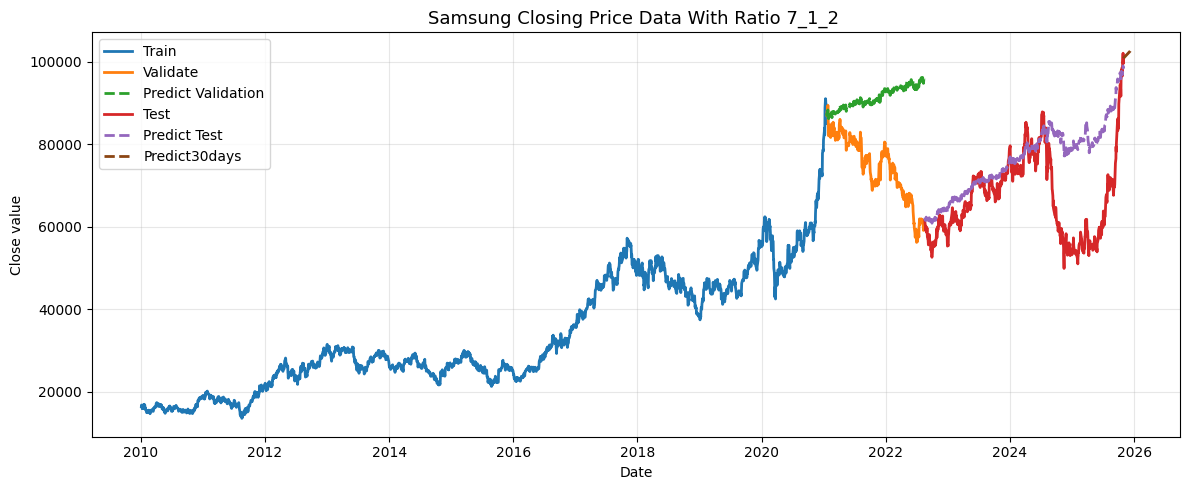
###### *Hình 5.42. Biểu đồ giá đóng cửa của Xiaomi từ 09/07/2018 đến 27/11/2025 trên mô hình GBT tỷ lệ 6:2:2*

## 5.8. Hidden Markov Model (HMM)

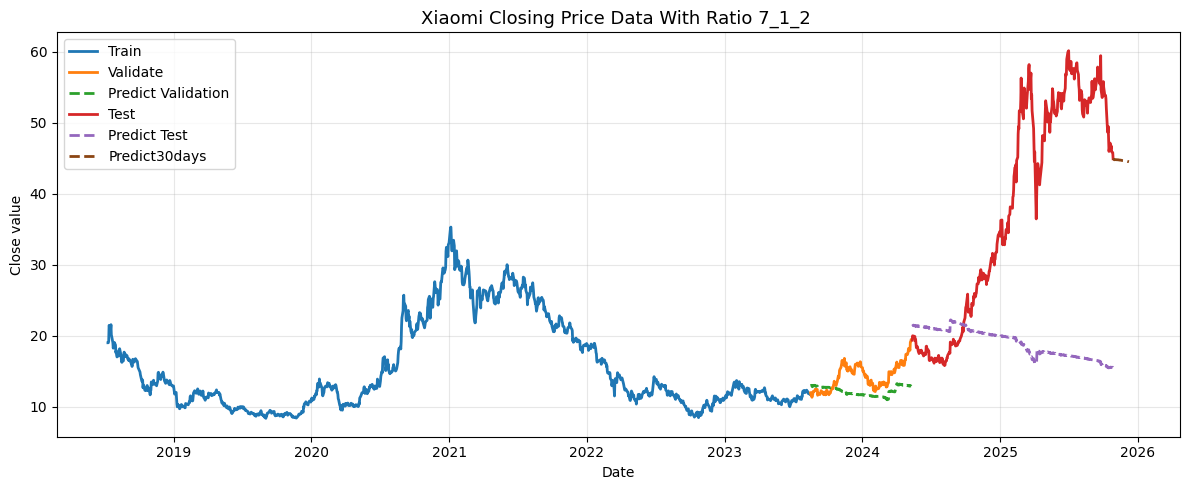
### 5.8.1. Tỉ lệ 7:1:2



###### *Hình 5.43.Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình HMM tỷ lệ 7:1:2*

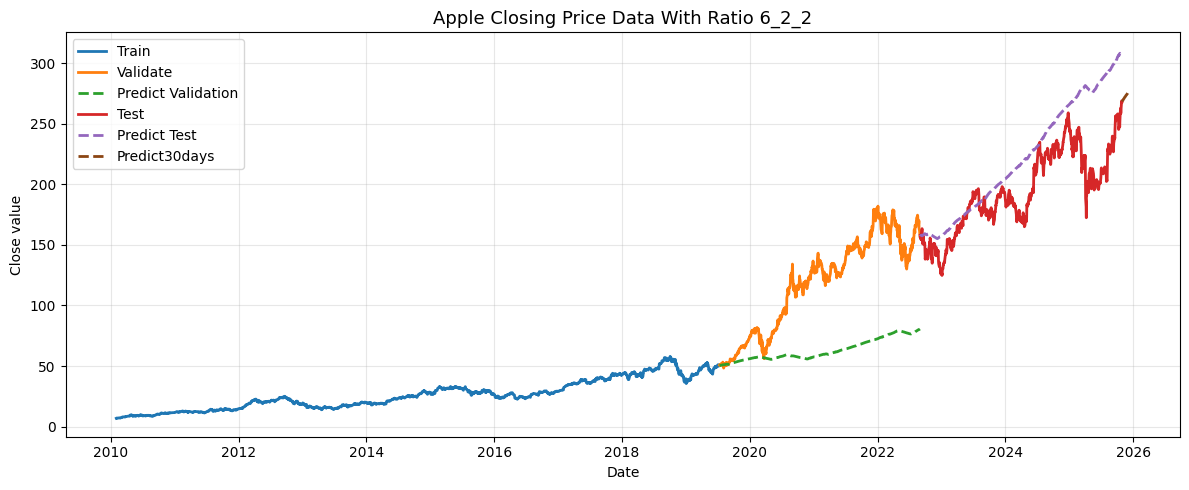


###### *Hình 5.44. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình HMM tỷ lệ 7:1:2*

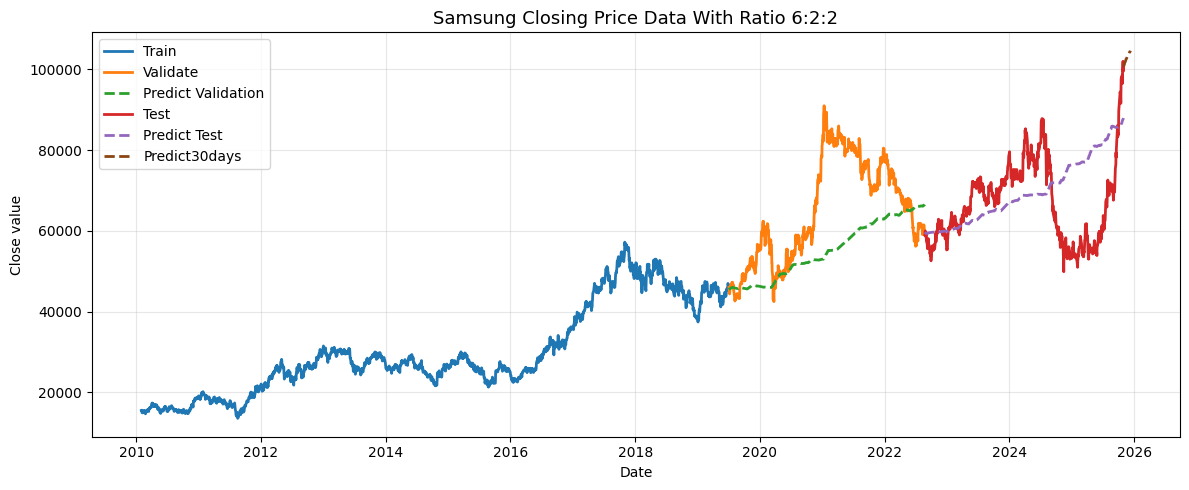


###### *Hình 5.45. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình HMM tỷ lệ 7:1:2*

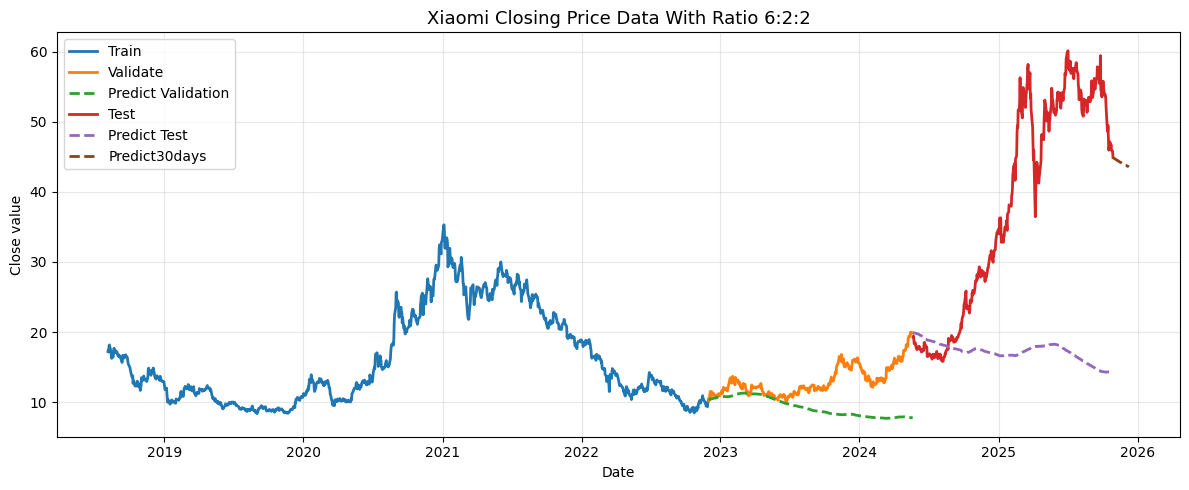
### 5.8.2. Tỉ lệ 6:2:2



###### *Hình 5.46. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình HMM tỷ lệ 6:2:2*



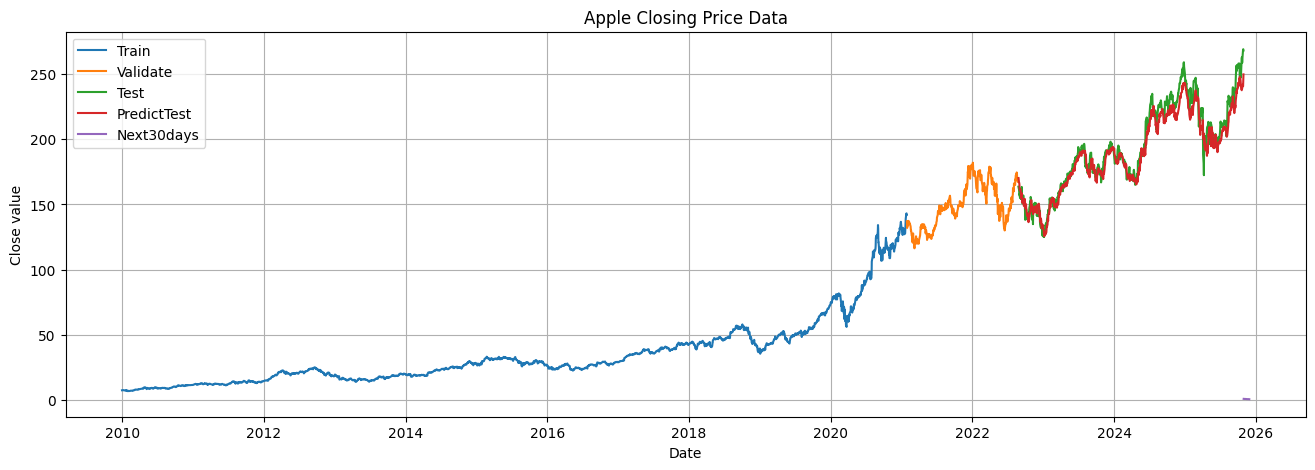
###### *Hình 5.47. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình HMM tỷ lệ 6:2:2*



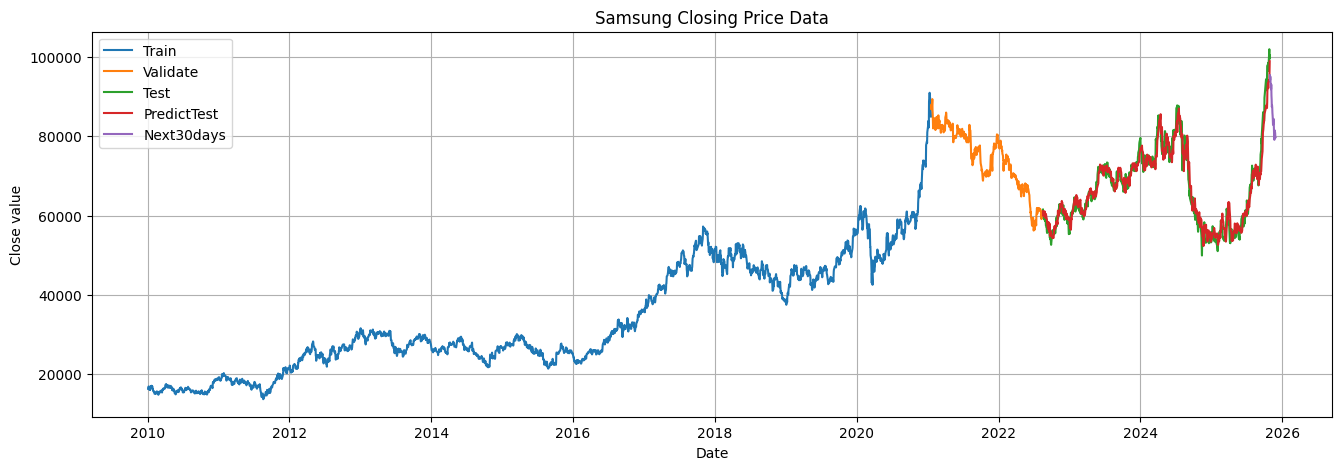
###### *Hình 5.48. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình HMM tỷ lệ 6:2:2*

## 5.9. Deep Neural Network (DNN)

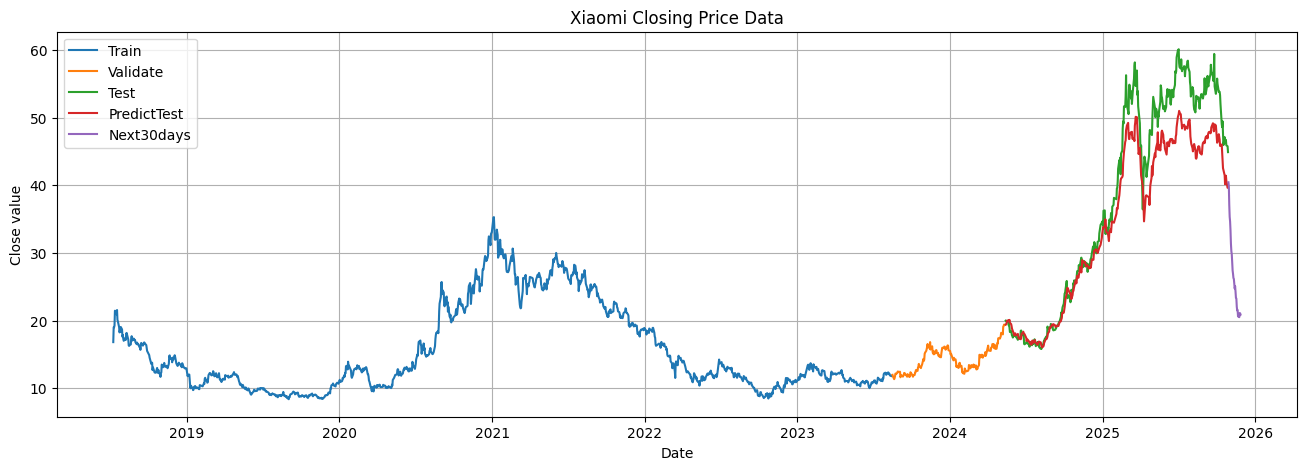
### 5.9.1. Tỉ lệ 7:1:2



###### *Hình 5.49. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình DNN tỷ lệ 7:1:2*

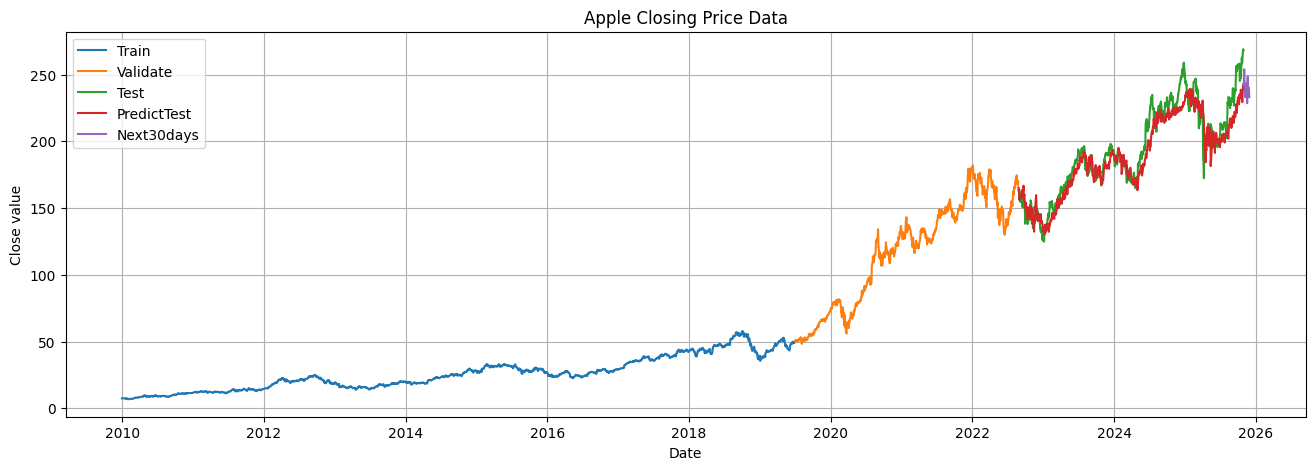


###### *Hình 5.50. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình DNN tỷ lệ 7:1:2*

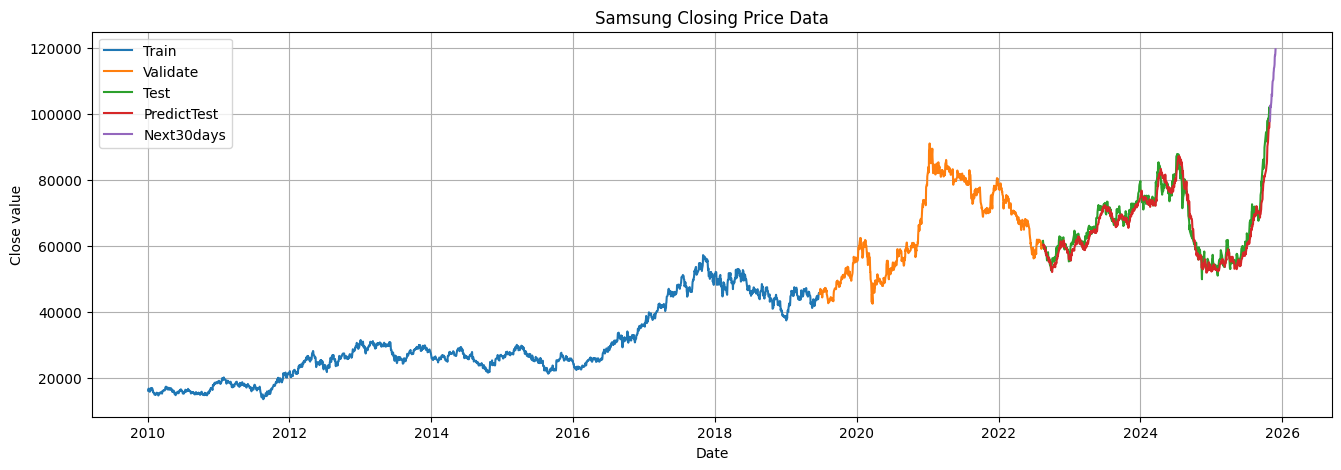


###### *Hình 5.51. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình DNN tỷ lệ 7:1:2*

### 5.9.2. Tỉ lệ 6:2:2



###### *Hình 5.52. Biểu đồ giá đóng cửa của Apple từ 31/12/2009 đến 28/11/2025 trên mô hình DNN tỷ lệ 6:2:2*



###### *Hình 5.53. Biểu đồ giá đóng cửa của Samsung từ 4/1/2010 đến 28/11/2025 trên mô hình DNN tỷ lệ 6:2:2*



###### *Hình 5.54. Biểu đồ giá đóng cửa của Xiaomi từ 9/7/2018 đến 27/11/2025 trên mô hình DNN tỷ lệ 6:2:2*

## 5.10. Đánh giá thuật toán

### 5.10.1. Đánh giá trên tập dữ liệu Validate

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | | Apple | | Samsung | | Xiaomi | |
| Tỷ lệ chia | | 7:1:2 | 6:2:2 | 7:1:2 | 6:2:2 | 7:1:2 | 6:2:2 |
| Mô hình | Độ đo |  | | | | | |
| LR | MAPE | 0.48 | 0.49 | 0.21 | 0.18 | 0.16 | 0.59 |
| RMSE | 73.44 | 73.45 | 18651.07 | 17025.95 | 2.54 | 7.39 |
| MSLE | 0.43 | 0.63 | 0.08 | 0.07 | 0.03 | 0.20 |
| GRU | MAPE | 1.415748 | 1.58104 | 0.9461421 | 1.854709 | 3.285213 | 2.0316688 |
| RMSE | 2.7969416 | 2.56799 | 927.55681 | 1530.686 | 0.5667300 | 0.370249 |
| MSLE | **0.00033044** | **0.0004737** | **0.0001656** | **0.0005329** | **0.0012851** | **0.000665** |
| ARIMA | MAPE | 1.8239 | 0.4574 | 0.2275 | 0.2708 | 0.1602 | 0.1841 |
| RMSE | 318.34 | 69.97 | 18623.34 | 23020.98 | 3.0726 | 3.2439 |
| MSLE | 1.0743 | 0.5290 | 0.0565 | 0.1504 | 0.0427 | 0.0568 |
| ETS | MAPE | **0.01371** | 0.16162308 | 0.01321 | 0.2685497 | 0.019686 | 0.0133425 |
| RMSE | 2.708315 | 41.268402 | 1212.914 | 22886.0992 | 0.38014 | 0.2592114 |
| MSLE | 0.00032 | 0.04459592 | 0.00032 | 0.148085 | 0.0006 | 0.00033 |
| LSTM | MAPE | **0.0205073** | **0.0261955** | **0.0272397** | **0.0262819** | **0.0407223** | **0.0394789** |
| RMSE | **0.0197407** | **0.0248025** | **0.0215868** | **0.0207897** | **0.0299066** | **0.0287405** |
| MSLE | **0.00012955** | **0.0002005** | **0.0001755** | **0.000161997** | **0.0003076** | **0.0002863** |
| RNN | MAPE | 0.03993 | 0,0049 | 0.018001 | 0.0283 | 0.02967 | 0.0393 |
| RMSE | 6.02004 | 6.4077 | 1638.278 | 2704.047 | 0.54734 | 2.21 |
| MSLE | 0.001688 | 0.001041 | 0.00049 | 0.001417 | 0.00126 | 0.002266 |
| GBT | MAPE | 0.08 | 0.46 | 0.10 | 0.17 | 0.02 | 0.02 |
| RMSE | 16.78 | 72.54 | 8392.49 | 16729.42 | 0.42 | 0.40 |
| MSLE | 0.01 | 0.59 | 0.01 | 0.07 | 0.0007 | 0.0008 |
| HMM | MAPE | 0.143747 | 0.411731 | 0.245936 | 0.148809 | 0.157340 | 0.262150 |
| RMSE | 22.406232 | 63.200710 | 19624.5605 | 13934.268122 | 2.737745 | 4.845230 |
| MSLE | 0.019393 | 0.39227568 | 0.061637 | 0.04462672 | 0.033852 | 0.15394398 |
| DNN | MAPE | 0.03 | 0.14 | 0.02 | 0.06 | 0.03 | 0.03 |
| RMSE | 4.40 | 6.65 | 1713.73 | 2280.37 | 0.48 | 0.48 |
| MSLE | 0.002 | 0.02 | 0.0008 | 0.005 | 0.001 | 0.001 |

###### *Bảng 5.1. Đánh giá các mô hình trên tập dữ liệu Validate*

Bảng 5.1 ghi nhận các giá trị độ đo **MAPE, RMSE và MSLE** của các mô hình đã được trình bày ở Chương 4 trên tập Validation của ba bộ dữ liệu **Apple, Samsung và Xiaomi** theo hai tỷ lệ train:validate:test là **7:1:2** và **6:2:2**.  
 Trong cả ba bộ dữ liệu, các giá trị tốt nhất (nhỏ nhất) của từng độ đo được in đậm và đánh dấu bằng màu đỏ nhằm làm nổi bật mô hình có hiệu suất cao nhất trong từng trường hợp.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Apple** | | | | | |
| **6:2:2** | | | **7:1:2** | | |
| **MAPE** | **RMSE** | **MSLE** | **MAPE** | **RMSE** | **MSLE** |
| RNN | LSTM | LSTM | ETS | LSTM | LSTM |
| LSTM | GRU | GRU | LSTM | ETS | ETS |
| DNN | RNN | RNN | DNN | GRU | GRU |
| ETS | DNN | DNN | RNN | DNN | RNN |
| HMM | ETS | ETS | GBT | RNN | DNN |
| ARIMA | ARIMA | HMM | HMM | GBT | GBT |
| GBT | GBT | ARIMA | LR | HMM | HMM |
| LR | LR | GBT | GRU | LR | LR |
| GRU | HMM | LR | ARIMA | ARIMA | ARIMA |

###### *Bảng 5.2. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Apple trên tập Validate*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Samsung** | | | | | |
| **6:2:2** | | | **7:1:2** | | |
| **MAPE** | **RMSE** | **MSLE** | **MAPE** | **RMSE** | **MSLE** |
| LSTM | LSTM | LSTM | ETS | LSTM | GRU |
| RNN | GRU | GRU | RNN | GRU | LSTM |
| DNN | GBT | RNN | DNN | ETS | ETS |
| HMM | LR | DNN | LSTM | RNN | RNN |
| GBT | ETS | HMM | GBT | DNN | DNN |
| LR | DNN | GBT | LR | GBT | GBT |
| ETS | RNN | LR | ARIMA | LR | ARIMA |
| ARIMA | HMM | ETS | HMM | ARIMA | HMM |
| GRU | ARIMA | ARIMA | GRU | HMM | LR |

###### *Bảng 5.3. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Samsung trên tập Validate*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Xiaomi** | | | | | |
| **6:2:2** | | | **7:1:2** | | |
| **MAPE** | **RMSE** | **MSLE** | **MAPE** | **RMSE** | **MSLE** |
| LSTM | LSTM | RNN | ETS | LSTM | LSTM |
| RNN | RNN | LSTM | GBT | ETS | ETS |
| DNN | DNN | DNN | RNN | GBT | GBT |
| ARIMA | ARIMA | GBT | DNN | DNN | DNN |
| ETS | ETS | GRU | LSTM | RNN | RNN |
| GRU | GRU | ARIMA | LR | GRU | GRU |
| GBT | GBT | ETS | ARIMA | LR | ARIMA |
| LR | LR | LR | HMM | ARIMA | LR |
| HMM | HMM | HMM | GRU | HMM | HMM |

###### *Bảng 5.4. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Xiaomi trên tập Validate*

Đối với **Mean Absolute Percentage Error (MAPE)**, mô hình **ETS** cho giá trị tốt nhất trên bộ dữ liệu Apple ở tỉ lệ 7:1:2 (0.01371) và vẫn duy trì trong nhóm mô hình có sai số thấp ở tỉ lệ 6:2:2. Trên bộ dữ liệu Samsung, ETS tiếp tục là mô hình đạt MAPE tốt nhất ở tỉ lệ 7:1:2 (0.01321), trong khi mô hình LR cho kết quả thấp nhất ở tỉ lệ 6:2:2 (0.18). Đối với bộ dữ liệu Xiaomi, ETS và GBT là hai mô hình có giá trị MAPE thấp nhất (0.01334 và 0.02), thể hiện khả năng dự báo ổn định về mặt phần trăm sai lệch.

Đối với **Root Mean Squared Error (RMSE)**, mô hình **LSTM** là mô hình cho giá trị thấp nhất trên cả ba bộ dữ liệu và ở cả hai tỉ lệ train:validate:test. Cụ thể, LSTM chỉ đạt lần lượt 0.01974 – 0.02480 trên Apple, 0.02158 – 0.02079 trên Samsung và 0.02990 – 0.02874 trên Xiaomi, thấp hơn đáng kể so với tất cả các mô hình truyền thống và học sâu khác.

Tương tự đối với **Mean Squared Logarithmic Error (MSLE)**, mô hình **LSTM** tiếp tục là mô hình có giá trị thấp nhất trong hầu hết các trường hợp. Trên cả ba bộ dữ liệu Apple, Samsung và Xiaomi, LSTM đều đạt các giá trị MSLE rất nhỏ (dao động trong khoảng 0.00012 – 0.00030), cho thấy khả năng dự báo ổn định và ít bị ảnh hưởng bởi các biến động dữ liệu theo logarit.

### 5.10.2. Đánh giá trên tập dữ liệu Test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | | Apple | | Samsung | | Xiaomi | |
| Tỷ lệ chia | | 7:1:2 | 6:2:2 | 7:1:2 | 6:2:2 | 7:1:2 | 6:2:2 |
| Mô hình | Độ đo |  | | | | | |
| LR | MAPE | 0.51 | 0.65 | 0.12 | 0.13 | 0.48 | 0.40 |
| RMSE | 104.63 | 131.32 | 9924.33 | 10687.78 | 26.59 | 21.76 |
| MSLE | 0.53 | 1.13 | 0.02 | 0.02 | 0.72 | 0.37 |
| GRU | MAPE | 0.012725 | 0.013022 | 0.014110 | 2.014020 | 0.022631 | 0.027217 |
| RMSE | 3.47 | 3.53 | 1304.15 | 1273.55 | 1.30 | 1.59 |
| MSLE | **0.000330** | **0.000337** | **0.000372** | **0.000356** | **0.000903** | **0.001232** |
| ARIMA | MAPE | 0.4510 | 0.1012 | 0.1218 | 0.1234 | 0.4201 | 0.4201 |
| RMSE | 113.59 | 23.86 | 9871.75 | 11711.46 | 24.1260 | 24.1260 |
| MSLE | 0.7812 | 0.0151 | 0.0210 | 0.0286 | 0.5091 | 0.5091 |
| ETS | MAPE | 0.01228 | 0.161623 | 0.01321 | 0.122941 | 0.02133 | 0.01425 |
| RMSE | 3.37451 | 41.26840 | 1212.91 | 11648.428 | 1.21607 | 0.871397 |
| MSLE | 0.0003 | 0.04459 | 0.00032 | 0.028287 | 0.00081 | 0.00042 |
| LSTM | MAPE | **0.0205073** | **0.0261955** | **0.0272397** | **0.0262819** | **0.0407223** | **0.0394789** |
| RMSE | **0.0197407** | **0.0248025** | **0.0215868** | **0.0207897** | **0.0299066** | **0.0287405** |
| MSLE | **0.00012955** | **0.0002005** | **0.0001755** | **0.000161997** | **0.0003076** | **0.0002863** |
| RNN | MAPE | 0.073308 | 0.0249 | 0.02134 | 0.0283 | 0.037811 | 0.0277 |
| RMSE | 20.0144 | 6.7130 | 1847.902 | 2704.0475 | 2.169517 | 1.6173 |
| MSLE | 0.008771 | 0.001041 | 0.00074 | 0.001417 | 0.002214 | 0.001278 |
| GBT | MAPE | 0.25 | 0.70 | 0.07 | 0.18 | 0.21 | 0.21 |
| RMSE | 61.70 | 141.95 | 6650.01 | 16528.36 | 14.69 | 14.53 |
| MSLE | 0.12 | 1.51 | 0.01 | 0.07 | 0.12 | 0.11 |
| HMM | MAPE | 0.069604 | 0.148005 | 0.027546 | 0.155413 | 0.081801 | 0.468112 |
| RMSE | 11.197317 | 37.686774 | 1837.74561 | 12397.891443 | 1.629519 | 26.427831 |
| MSLE | 0.004556 | 0.02662210 | 0.000923 | 0.03299802 | 0.005712 | 0.69995673 |
| DNN | MAPE | 0.03 | 0.04 | 0.02 | 0.03 | 0.10 | 0.10 |
| RMSE | 7.91 | 11.22 | 2357.20 | 2706.62 | 5.42 | 5.42 |
| MSLE | 0.001 | 0.003 | 0.001 | 0.001 | 0.01 | 0.01 |

###### *Bảng 5.5. Đánh giá các mô hình trên tập dữ liệu Test*

Bảng 5.5 ghi nhận các giá trị độ đo **MAPE, RMSE và MSLE** của các mô hình đã được trình bày ở Chương 4 trên tập Test của ba bộ dữ liệu **Apple, Samsung và Xiaomi** theo hai tỷ lệ train:validate:test là **7:1:2** và **6:2:2**.

Trong cả ba bộ dữ liệu, các giá trị tốt nhất (nhỏ nhất) của từng độ đo được in đậm và đánh dấu bằng màu đỏ nhằm làm nổi bật mô hình có hiệu suất cao nhất trong từng trường hợp.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Apple** | | | | | |
| **6:2:2** | | | **7:1:2** | | |
| **MAPE** | **RMSE** | **MSLE** | **MAPE** | **RMSE** | **MSLE** |
| GRU | LSTM | LSTM | GRU | LSTM | LSTM |
| RNN | GRU | GRU | ETS | ETS | ETS |
| LSTM | RNN | RNN | LSTM | GRU | GRU |
| DNN | DNN | DNN | DNN | DNN | DNN |
| ARIMA | ARIMA | ARIMA | HMM | HMM | HMM |
| HMM | HMM | HMM | RNN | RNN | RNN |
| ETS | ETS | ETS | GBT | GBT | GBT |
| LR | GBT | LR | ARIMA | LR | LR |
| GBT | LR | GBT | LR | ARIMA | ARIMA |

###### *Bảng 5.6. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Apple trên tập Test*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Samsung** | | | | | |
| **6:2:2** | | | **7:1:2** | | |
| **MAPE** | **RMSE** | **MSLE** | **MAPE** | **RMSE** | **MSLE** |
| LSTM | LSTM | LSTM | ETS | LSTM | LSTM |
| RNN | GRU | GRU | GRU | GRU | ETS |
| DNN | DNN | DNN | DNN | ETS | GRU |
| ETS | RNN | RNN | RNN | RNN | RNN |
| ARIMA | ETS | LR | LSTM | HMM | HMM |
| HMM | ARIMA | ETS | HMM | DNN | DNN |
| GBT | HMM | ARIMA | GBT | GBT | GBT |
| LR | GBT | HMM | ARIMA | ARIMA | LR |
| GRU | LR | GBT | LR | LR | ARIMA |

*Bảng 5.7. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Samsung trên tập Test*

###### 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Xiaomi** | | | | | |
| **6:2:2** | | | **7:1:2** | | |
| **MAPE** | **RMSE** | **MSLE** | **MAPE** | **RMSE** | **MSLE** |
| ETS | LSTM | LSTM | ETS | LSTM | LSTM |
| GRU | GRU | GRU | GRU | ETS | GRU |
| RNN | RNN | RNN | RNN | GRU | ETS |
| LSTM | ETS | ETS | LSTM | HMM | HMM |
| DNN | DNN | DNN | HMM | RNN | DNN |
| GBT | GBT | GBT | DNN | DNN | RNN |
| LR | LR | LR | GBT | GBT | GBT |
| HMM | ARIMA | HMM | ARIMA | ARIMA | ARIMA |
| ARIMA | HMM | ARIMA | LR | LR | LR |

###### *Bảng 5.8. Xếp hạng các giá trị đánh giá đo được của các mô hình theo thứ tự tăng dần từ trên xuống cho bộ dữ liệu Xiaomi trên tập Test*

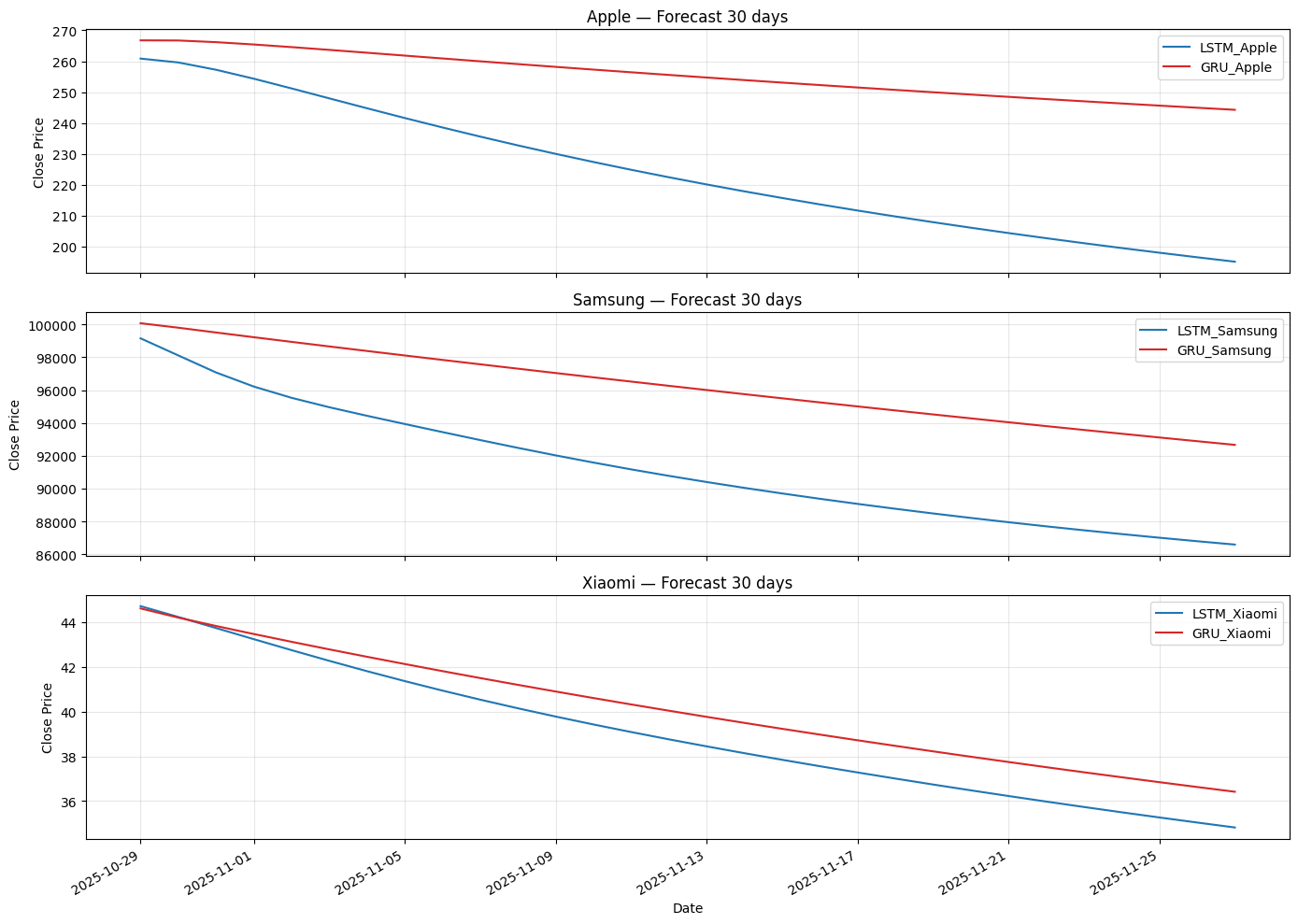
Đối với **Mean Absolute Percentage Error (MAPE)**, mô hình **GRU** đạt được giá trị thấp nhất trên bộ dữ liệu Apple ở cả hai tỉ lệ (0.012725 và 0.013022). Trên bộ dữ liệu Samsung, GRU tiếp tục là mô hình có sai số phần trăm thấp nhất ở tỉ lệ 7:1:2 (0.01411), trong khi ETS và ARIMA cho kết quả tốt hơn ở tỉ lệ 6:2:2. Trong khi đó, đối với bộ dữ liệu Xiaomi, mô hình ETS vượt trội với MAPE lần lượt là 0.02133 và 0.01425, thấp nhất trong tất cả các mô hình ở cả hai tỉ lệ.

Về **Root Mean Square Error (RMSE)**, **LSTM** tiếp tục là mô hình cho giá trị thấp nhất trên cả ba bộ dữ liệu và ở cả hai tỉ lệ train:validate:test. Cụ thể, LSTM chỉ đạt 0.01974 – 0.02480 trên Apple, 0.02158 – 0.02079 trên Samsung và 0.02990 – 0.02874 trên Xiaomi, thấp hơn đáng kể so với tất cả các mô hình truyền thống (ARIMA, ETS, LR) và mô hình học sâu khác (GRU, RNN, DNN).

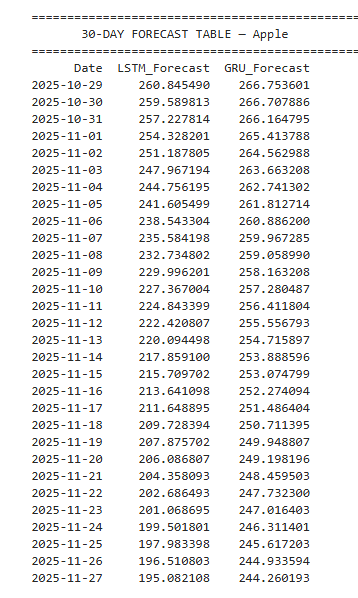
Đối với **Mean Squared Logarithmic Error (MSLE)**, LSTM cũng là mô hình cho kết quả thấp nhất trên hầu hết các trường hợp của ba bộ dữ liệu Apple, Samsung và Xiaomi. Các giá trị MSLE của LSTM dao động trong khoảng 0.00012 – 0.00030, cho thấy mô hình hoạt động ổn định và ít bị ảnh hưởng bởi biến động dữ liệu theo logarit. Mặc dù GRU cũng đạt MSLE khá thấp nhưng vẫn không vượt qua LSTM trong đa số trường hợp.

Tổng kết lại, từ phân tích các độ đo lỗi MAPE, RMSE và MSLE trên tập Test của ba bộ dữ liệu Apple, Samsung và Xiaomi, nhóm nhận thấy hai mô hình thể hiện hiệu quả vượt trội và ổn định nhất là **Long Short-Term Memory (LSTM)** và **Gated Recurrent Unit (GRU)**. Hai mô hình này đạt kết quả tốt nhất trên nhiều bộ dữ liệu và nhiều chỉ số đánh giá khác nhau, do đó được lựa chọn để dự đoán giá đóng cửa (Close) trong 30 ngày tiếp theo.

## 5.11. Dự đoán giá cổ phiếu trong 30 ngày tiếp theo

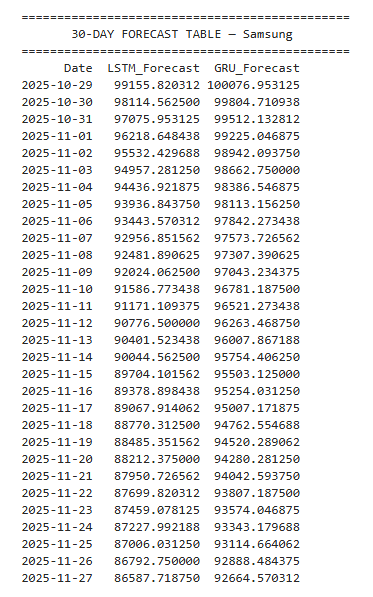


###### *Hình 5.55. Dự đoán giá cổ phiếu trong 30 ngày tiếp theo với tỉ lệ 6:2:2*



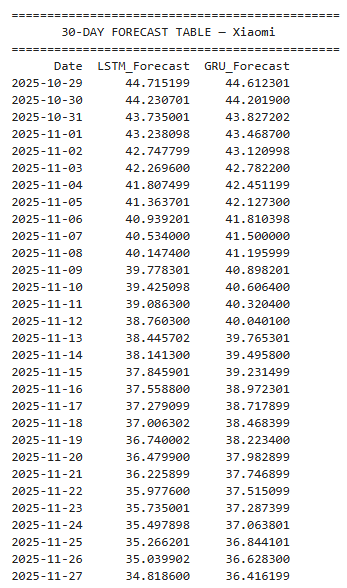
###### *Hình 5.56. Dự đoán 30 ngày của Apple trên tỷ lệ 6:2:2*

###### 

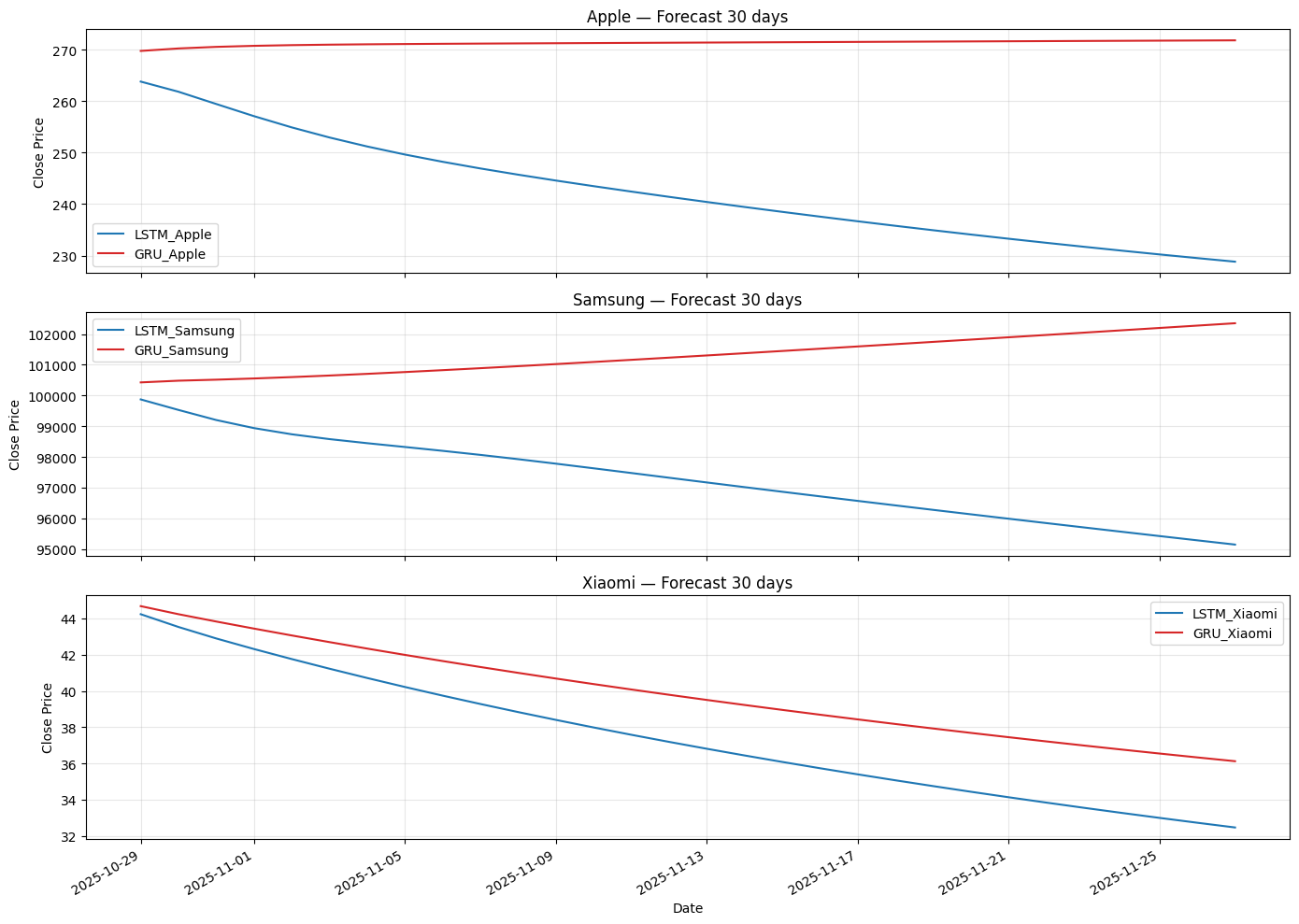


###### *Hình 5.57. Dự đoán 30 ngày của Samsung trên tỷ lệ 6:2:2*

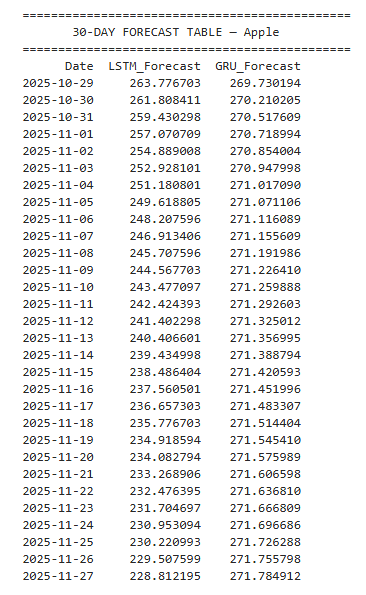
###### 



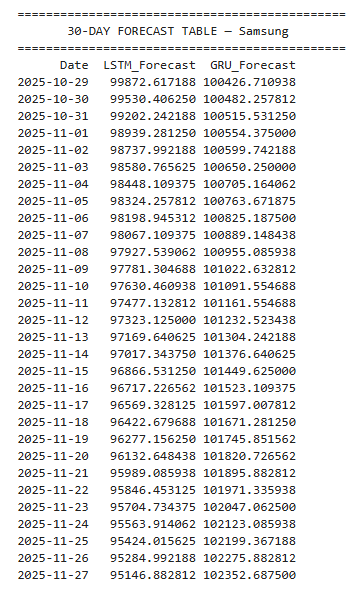
###### *Hình 5.58. Dự đoán 30 ngày của Xiaomi trên tỷ lệ 6:2:2*



###### *Hình 5.59. Dự đoán giá cổ phiếu trong 30 ngày tiếp theo với tỉ lệ 7:1:2*

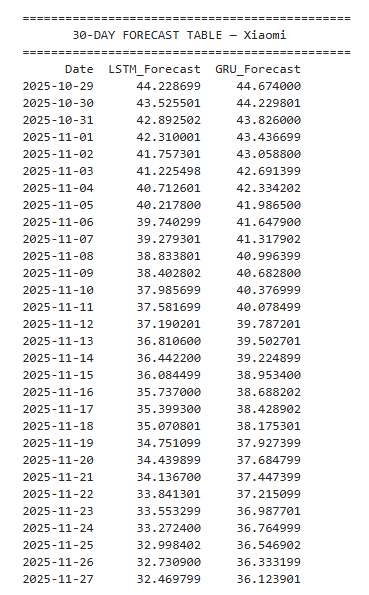


###### *Hình 5.60. Dự đoán của Apple trên tỷ lệ 7:1:2*



###### *Hình 5.61. Dự đoán 30 ngày của Samsung trên tỷ lệ 7:1:2*

###### 



###### *Hình 5.62. Dự đoán 30 ngày của Xiaomi trên tỷ lệ 7:1:2*

# KẾT LUẬN

Trên cơ sở mục tiêu nghiên cứu đã đề ra, chúng tôi đã tiến hành phân tích, xây dựng và đánh giá hiệu quả của các mô hình dự báo chuỗi thời gian trong bài toán dự đoán giá cổ phiếu, bao gồm: Linear Regression, ARIMA, ETS, GBT, HMM, RNN, DNN, LSTM và GRU. Quá trình thực nghiệm được tiến hành trên ba bộ dữ liệu thực tế của Apple Inc., Samsung Electronics Co., Ltd. và Xiaomi Corporation, với hai tỷ lệ chia dữ liệu khác nhau (7:1:2 và 6:2:2) nhằm đảm bảo tính khách quan và khả năng khái quát hóa của mô hình.

Kết quả đánh giá cho thấy, các mô hình học sâu thuộc nhóm mạng nơ-ron hồi quy, đặc biệt là LSTM (Long Short-Term Memory) và GRU (Gated Recurrent Unit), đạt hiệu suất dự báo vượt trội so với các mô hình truyền thống. Hai mô hình này thể hiện khả năng nắm bắt tốt các đặc trưng phi tuyến và mối quan hệ phụ thuộc theo thời gian trong dữ liệu giá cổ phiếu, qua đó mang lại kết quả dự báo ổn định và chính xác hơn. Đây cũng là hai mô hình được lựa chọn để thực hiện dự báo giá đóng cửa cổ phiếu trong 30 ngày tiếp theo, đáp ứng mục tiêu ứng dụng thực tiễn của nghiên cứu.

Tuy nhiên, nghiên cứu vẫn đối mặt với một số hạn chế nhất định, như sự biến động mạnh và tính khó đoán của thị trường tài chính, cũng như giới hạn về phạm vi dữ liệu đầu vào, chủ yếu tập trung vào chuỗi thời gian giá cổ phiếu mà chưa xem xét tác động của các yếu tố bên ngoài như tin tức, sự kiện kinh tế hoặc tâm lý nhà đầu tư.

Trong tương lai, chúng tôi nhận định rằng có thể triển khai cải thiện theo hướng mở rộng phạm vi dữ liệu, kết hợp thêm các đặc trưng phi cấu trúc (ví dụ: dữ liệu văn bản từ tin tức hoặc mạng xã hội) nhằm nâng cao độ chính xác của mô hình. Bên cạnh đó, việc tinh chỉnh và kết hợp các mô hình LSTM và GRU với các phương pháp hiện đại khác (như mô hình lai hoặc ensemble) cũng sẽ được xem xét như một hướng nghiên cứu tiềm năng, hướng tới việc xây dựng một hệ thống dự báo thông minh và đáng tin cậy hơn trong lĩnh vực phân tích tài chính doanh nghiệp.

# PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên thành viên  Nội dung   công việc | | Hoàng Lê Nhật Minh | Trịnh Thái Long | Đinh Nhật Luân | Cam Hồng Mạnh | Thượng Trung Nguyên | Chu Thị Cẩm Tú |
| Viết báo cáo | | x | x | x | x | x | x |
| Kiểm tra và chỉnh sửa | | x |  |  | x |  |  |
| Giới thiệu | |  |  |  | x |  | x |
| Phân tích dữ liệu | | x |  |  | x |  |  |
| Đánh giá kết quả thực nghiệm | | x |  |  |  |  |  |
| Giới thiệu độ đo | |  |  |  | x |  | x |
| Trực quan hóa dự đoán 30 ngày | | x |  |  |  |  |  |
| Kết luận | |  |  |  | x |  |  |
| Tìm hiểu  thuật toán | Linear Regression |  |  |  |  | x |  |
| GRU | x |  |  |  |  |  |
| ARIMA |  |  |  | x |  |  |
| ETS |  |  |  |  | x |  |
| LSTM | x |  |  |  |  |  |
| RNN |  |  | x |  |  |  |
| GBT |  | x |  |  |  |  |
| HMM |  |  |  |  |  | x |
| DNN |  | x |  |  |  |  |
| Làm slide báo cáo | |  |  |  |  | x |  |
| Đánh giá mức độ  hoàn thành | | 100% | 100% | 100% | 100% | 100% | 100% |

# TÀI LIỆU THAM KHẢO

[1] N. T. Bình, “Ảnh hưởng của các nhân tố vĩ mô đến lợi nhuận cổ phiếu: Thị trường chứng khoán Việt Nam,” 2023, Accessed: Nov. 07, 2025. [Online]. Available: https://digital.lib.ueh.edu.vn/handle/UEH/69069

[2] “Stock Price Prediction Using a Frequency Decomposition Based GRU Transformer Neural Network.” Accessed: Nov. 07, 2025. [Online]. Available: https://www.mdpi.com/2076-3417/13/1/222

[3] “Multi-feature stock price prediction by LSTM networks based on VMD and TMFG | Journal of Big Data | Full Text.” Accessed: Nov. 07, 2025. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-025-01127-4?utm\_source=chatgpt.com

[4] Q. Tang, T. Fan, R. Shi, J. Huang, and Y. Ma, “Prediction of financial time series using LSTM and data denoising methods,” Mar. 05, 2021, *arXiv*: arXiv:2103.03505. doi: 10.48550/arXiv.2103.03505.

[5] V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos, “A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks,” *Future Internet*, vol. 15, no. 8, p. 255, Aug. 2023, doi: 10.3390/fi15080255.

[6] A. Sunki, C. SatyaKumar, G. Surya Narayana, V. Koppera, and M. Hakeem, “Time series forecasting of stock market using ARIMA, LSTM and FB prophet,” *MATEC Web Conf.*, vol. 392, p. 01163, 2024, doi: 10.1051/matecconf/202439201163.

[7] N. T. T. Loan, D. Hung, and V. Van, “Application of ARIMA model and deep learning in forecasting stock price in Vietnam,” *Salud Cienc. Tecnol. - Ser. Conf.*, vol. 5, pp. 1320–1329, Jan. 2025, doi: 10.56294/sctconf20251320.

[8] “Prediction of Financial Time-series Data Using ARIMA Model and Machine Learning Algorithm | Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence.” Accessed: Nov. 07, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3745238.3745492

[9] Y. Gao, “A Comparative Study of ARIMA and ETS Models for Time Series Forecasting,” *Adv. Econ. Manag. Polit. Sci.*, vol. 149, pp. 196–201, Jan. 2025, doi: 10.54254/2754-1169/2024.19252.

[10] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “Statistical and Machine Learning forecasting methods: Concerns and ways forward,” *PLOS ONE*, vol. 13, no. 3, p. e0194889, thg 3 2018, doi: 10.1371/journal.pone.0194889.

[11] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “The M4 Competition: 100,000 time series and 61 forecasting methods,” *Int. J. Forecast.*, vol. 36, no. 1, pp. 54–74, Jan. 2020, doi: 10.1016/j.ijforecast.2019.04.014.

[12] S. Hansun and J. C. Young, “Predicting LQ45 financial sector indices using RNN-LSTM,” *J. Big Data*, vol. 8, no. 1, p. 104, July 2021, doi: 10.1186/s40537-021-00495-x.

[13] L. Zeng and J. Li, “Research on Stock Price Prediction Based on LSTM Neural Network Modeling,” *J. Innov. Dev.*, vol. 11, no. 3, pp. 6–18, June 2025, doi: 10.54097/8kp7bf18.

[14] Y. Zhang, “Stock Price Prediction Using ARIMA and LSTM Models: An Application to CSI 300 Closing Prices,” *Highlights Sci. Eng. Technol.*, vol. 88, pp. 286–293, Mar. 2024, doi: 10.54097/7dgx1c14.

[15] Z. Stankeviciute, E. Staniškienė, and J. Ramanauskaitė, “The Impact of Job Insecurity on Organisational Citizenship Behaviour and Task Performance: Evidence from Robotised Furniture Sector Companies,” *Int. J. Environ. Res. Public. Health*, vol. 18, p. 515, Jan. 2021, doi: 10.3390/ijerph18020515.

[16] R. Kumari and D. A. Kumari, “A MATHEMATICAL MODEL FOR THE IMPACT OF PHYSICAL EXERCISES AND NUTRITION ON DIABETIC PATIENTS,” vol. 12, no. 1, 2024.

[17] A. Adebiyi, A. Adewumi, and C. Ayo, *Stock price prediction using the ARIMA model*. 2014. doi: 10.1109/UKSim.2014.67.

[18] J. R. Reyes, “Nowcasting domestic liquidity in the Philippines using machine learning algorithms,” *Philipp. Rev. Econ.*, vol. 59, no. 2, pp. 1–40, Dec. 2022, doi: 10.37907/1ERP2202D.

[19] R. Tanvir, M. T. R. Shawon, and M. G. R. Alam, “DSE Stock Price Prediction using Hidden Markov Model,” Jan. 26, 2023, *arXiv*: arXiv:2302.08911. doi: 10.48550/arXiv.2302.08911.

[20] F. Correia, A. M. Madureira, and J. Bernardino, “Deep Neural Networks Applied to Stock Market Sentiment Analysis,” *Sensors*, vol. 22, no. 12, p. 4409, June 2022, doi: 10.3390/s22124409.

[21] “Using Market News Sentiment Analysis for Stock Market Prediction.” Accessed: Nov. 07, 2025. [Online]. Available: https://www.mdpi.com/2227-7390/10/22/4255

[22] “Yahoo Finance - Stock Market Live, Quotes, Business & Finance News.” Accessed: Nov. 07, 2025. [Online]. Available: https://finance.yahoo.com/

[23] LuanDinh, *DinhNhatLuan/IS403.Q12-*. (Nov. 07, 2025). Jupyter Notebook. Accessed: Nov. 07, 2025. [Online]. Available: https://github.com/DinhNhatLuan/IS403.Q12-

[24] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. doi: 10.3115/v1/D14-1179.

[25] “Understanding LSTM Networks -- colah’s blog.” Accessed: Nov. 07, 2025. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/?utm\_source=chatgpt.com

[26] <p>Han Peng and Z. Yang</p>, “An Empirical Study on Stock Price Forecasting Based on ARIMA Model,” *Front. Soc. Sci. Technol.*, vol. 4, no. 6, Aug. 2022, doi: 10.25236/FSST.2022.040605.

[27] A. Meyler, G. Kenny, and T. Quinn, “Forecasting irish inflation using ARIMA models,” Jan. 1999.

[28] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM,” Nov. 21, 2019, *arXiv*: arXiv:1911.09512. doi: 10.48550/arXiv.1911.09512.

[29] S. B. Kwarteng and P. A. Andreevich, “Comparative Analysis of ARIMA, SARIMA and Prophet Model in Forecasting,” *Res. Dev.*, vol. 5, no. 4, pp. 110–120, Oct. 2024, doi: 10.11648/j.rd.20240504.13.

[30] “Predictive Analytics in Finance Using the Arima Model....” Accessed: Nov. 07, 2025. [Online]. Available: https://reference-global.com/article/10.2478/sbe-2024-0042

[31] *Forecasting: Principles and Practice (3rd ed)*. Accessed: Nov. 07, 2025. [Online]. Available: https://otexts.com/fpp3/

[32] P. R. Winters, “Forecasting Sales by Exponentially Weighted Moving Averages,” *Manag. Sci.*, vol. 6, no. 3, pp. 324–342, Apr. 1960, doi: 10.1287/mnsc.6.3.324.

[33] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, vol. 385. in Studies in Computational Intelligence, vol. 385. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-24797-2.

[34] “Long Short-Term Memory | MIT Press Journals & Magazine | IEEE Xplore.” Accessed: Nov. 07, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6795963

[35] F. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” *Neural Comput.*, vol. 12, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.

[36] “deeplearningbook.org/contents/rnn.html.” Accessed: Nov. 07, 2025. [Online]. Available: https://www.deeplearningbook.org/contents/rnn.html

[37] “MAPE - Mean Absolute Percentage Error — Permetrics 2.0.0 documentation.” Accessed: Nov. 07, 2025. [Online]. Available: https://permetrics.readthedocs.io/en/latest/pages/regression/MAPE.html

[38] “RMSE - Root Mean Square Error — Permetrics 2.0.0 documentation.” Accessed: Nov. 07, 2025. [Online]. Available: https://permetrics.readthedocs.io/en/latest/pages/regression/RMSE.html

[39] “MSLE - Mean Squared Logarithmic Error — Permetrics 2.0.0 documentation.” Accessed: Nov. 07, 2025. [Online]. Available: https://permetrics.readthedocs.io/en/latest/pages/regression/MSLE.html