Tạp chí Khoa học Đại học Cần Thơ
Phần D: Kinh tế và Pháp luật
ISSN 1859-2333 | e-ISSN 2815-5599

DOI:10.22144/ctujos.2023.232

ĐÁNH GIÁ HIỆU SUẤT MÔ HÌNH PHỨC HỢP LSTM-GRU: NGHIÊN CỨU ĐIỂN HÌNH VỀ DỰ BÁO CHỈ SỐ ĐO LƯỜNG XU HƯỚNG BIẾN ĐỘNG GIÁ CỔ PHIẾU TRÊN SÀN GIAO DỊCH CHỨNG KHOÁN HỒ CHÍ MINH

Trần Đăng Tuyên*
Trường Kinh tế, Trường Đại học Cần Thơ
**Tác giả liên hệ (Corresponding author): tuyenm4522033@gstudent.ctu.edu.vn*

Thông tin chung (Article Information)

Nhận bài (Received): 14/08/2023
Sửa bài (Revised): 11/09/2023
Duyệt đăng (Accepted): 13/09/2023

Title: *Performance evaluation of LSTM-GRU Hybrid models: A case study on forecasting stock price volatility trends on the Ho Chi Minh Stock Exchange*

Author(s): *Tran Dang Tuyen**

Affiliation(s): *Can Tho University*

TÓM TẮT

Thị trường chứng khoán là một hệ thống chuyển động phi tuyến rất phức tạp và quy luật biến động của nó bị ảnh hưởng bởi rất nhiều yếu tố, vì vậy việc dự đoán chỉ số giá cổ phiếu là một nhiệm vụ rất khó khăn. Mô hình mạng nơ-ron với bộ nhớ ngắn hạn định hướng dài hạn (LSTM), mạng nơ-ron hồi tiếp với nút cổng (GRU) và các phức hợp được thiết kế bằng ngôn ngữ lập trình Python với các gói phụ trợ có sẵn, cho thấy kết quả dự báo với độ chính xác cao, hiệu suất của mô hình LSTM-GRU Hybrid cho kết quả tốt nhất. Thông qua mô hình LSTM-GRU Hybrid, nghiên cứu dự báo xu hướng biến động chỉ số VNIndex 100 ngày tiếp theo cho kết quả chỉ số VNIndex có xu hướng tăng. Điều đó gián tiếp chỉ ra rằng thị trường chứng khoán Việt Nam có dấu hiệu khởi sắc trở lại cùng với các chính sách mới của Chính phủ.

Từ khoá: *Dự báo, GRU, HoSE, LSTM, máy học, VNIndex*

ABSTRACT

The stock market is a highly complex non-linear system, and its volatility is influenced by numerous factors, making stock price prediction a challenging task. Long Short Term Memory networks (LSTM) model, Gated Recurrent Unit (GRU) model, and their hybrids are designed using the Python programming language with available supporting packages, and they demonstrate high accuracy in forecasting. The LSTM-GRU Hybrid model performs the best among the complex models. Through the LSTM-GRU Hybrid model, the study predicts the trend of the VNIndex for the next 100 days, indicating a rising trend. This indirectly suggests a potential resurgence in the Vietnamese stock market, driven by new government policies.

Keywords: *Forecasting, GRU, HoSE, LSTM, machine learning, VNIndex*

1. GIỚI THIỆU

VNIndex là một chỉ số thị trường thể hiện xu hướng biến động giá cổ phiếu niêm yết trên sàn giao dịch chứng khoán Hồ Chí Minh *(HoSE). Dựa vào chỉ số này người ta có thể biết được cụ thể quy mô cũng như giá trị các cổ phiếu tại thời điểm hiện tại, so với các mức giá trị tính theo ngày cơ sở là 28/07/2000.* Vấn đề đo lường xu hướng chỉ số VNIndex cho biết một cách khách quan sự tăng trưởng hay suy thoái của nền kinh tế hiện tại, đồng thời mô tả sự dịch chuyển của nền kinh tế. Nếu nền kinh tế có sự tái cơ cấu lại các ngành, giá cổ phiếu từng ngành sẽ có sự thay đổi, kéo theo chỉ số chứng khoán bị tác động trực tiếp.

Dự báo chính xác xu hướng biến động của giá cổ phiếu thật sự là thách thức vì giá cổ phiếu chứa đựng rất nhiều thông tin không rõ ràng, nhiều yếu tố nhiễu và còn phụ thuộc tâm lý đám đông (Chen & Hao, 2017; Chung & Shin, 2020). Ngoài ra, giá cổ phiếu bị ảnh hưởng bởi các chính sách của Chính phủ, cũng như thông tin về tình hình chính trị, kinh tế - xã hội, thậm chí cả thị trường chứng khoán Hoa Kỳ cũng tác động đến thị trường chứng khoán trong nước. Tuy nhiên, hiện nay dự báo giá cổ phiếu thu hút được nhiều sự quan tâm nghiên cứu (Yu & Yan, 2020). Các nghiên cứu đi theo hai hướng đó là phân tích cơ bản và phân tích kỹ thuật. Phân tích cơ bản, giả sử rằng bất kỳ cổ phiếu riêng lẻ có giá trị nội tại phụ thuộc vào tiềm năng thu nhập của giá cổ phiếu như chất lượng quản lý, triển vọng ngành và triển vọng kinh tế. Phân tích cơ bản sử dụng các phương pháp tài chính xác định giá thực tế của cổ phiếu cao hơn hoặc thấp hơn giá trị nội tại của nó và tạo ra dự đoán về giá trị tương lai (Nti et al., 2020); trong khi phân tích kỹ thuật hoặc biểu đồ giả định rằng sự thay đổi giá cổ phiếu có sự lặp lại xu hướng trong quá khứ. Giá cổ phiếu phản ánh các yếu tố như xu hướng, biến động kinh tế, tâm lý thị trường và kết quả kinh doanh. Do đó phân tích kỹ thuật chỉ chú trọng sự biến động của giá để đưa ra chiến lược phù hợp. Phân tích kỹ thuật sử dụng các thống kê hoặc các phương pháp toán học dựa vào lịch sử giao dịch, khối lượng giao dịch để xác định xu hướng dự báo (Lin et al., 2011). Phân tích cơ bản phản ánh giá trị nội tại của cổ phiếu nhưng không phản ánh sự thay đổi liên tục của cổ phiếu (Yun et al., 2021). Báo cáo tài chính của công ty, bảng cân đối kế toán, báo cáo kết quả hoạt động kinh doanh thường theo khoảng thời gian dài, vì vậy chúng không phù hợp để dự báo chiều biến động thường xuyên của giá cổ phiếu (Song et al., 2019; Yun et al., 2021). Phân tích và dự báo chuỗi thời gian đã được nghiên cứu chuyên sâu hơn 40 năm. Một trong những phương pháp truyền thống để dự báo chiều biến động của cổ phiếu là mô hình Trung bình di động tích hợp tự hồi quy (ARIMA) được sử dụng để nghiên cứu các quá trình thay đổi theo thời gian. Tuy nhiên, một hạn chế của ARIMA là xu hướng tự nhiên của nó tập trung vào các giá trị trung bình của chuỗi dữ liệu quá khứ. Do đó, vẫn còn khó khăn để nắm bắt một quá trình thay đổi nhanh chóng (Hong, 2021). Hỗ trợ hồi quy vectơ (SVR) đã được áp dụng thành công để dự đoán chuỗi thời gian, nhưng nó cũng có những nhược điểm như thiếu phương pháp có cấu trúc để xác định một số tham số chính của mô hình. Tuy nhiên, dữ liệu cổ phiếu là phi tuyến, không cố định thậm chí có yếu tố mùa vụ, vì vậy khó đáp ứng các giả định của mô hình. Sự phát triển của máy học và trí tuệ nhân tạo đã mở ra hướng nghiên cứu cho dự báo giá hoặc chiều biến động của cổ phiếu mà không đòi hỏi các điều kiện chặt chẽ về dữ liệu. Các nghiên cứu dự báo giá cổ phiếu chủ yếu đang sử dụng lịch sử giá giao dịch mà ít nghiên cứu sử dụng các chỉ báo kỹ thuật để dự báo chiều biến động của chỉ số chứng khoán tại Việt Nam. Đây là khoảng trống cần nghiên cứu nhằm kiểm tra tính hiệu quả của các chỉ báo kỹ thuật ứng dụng mô hình học máy. Các mô hình học máy phổ biến nhất hiện nay như mạng nơ-ron hồi quy (RNN - Recurrent Neural Network), mạng nơ-ron với bộ nhớ ngắn hạn định hướng dài hạn (LSTM - Long Short Term Memory networks), mạng nơ-ron hồi tiếp với nút cổng (GRU – Gated Recurrent Unit), mạng nơ-ron chuyển đổi (Transformer Neural Network), và các dạng phức hợp của chúng. Trong những năm gần đây do có cấu trúc linh hoạt, các mô hình máy học ngày càng được sử dụng nhiều hơn trong dự đoán chuỗi thời gian. Cụ thể, RNN một trong những mô hình máy học, được thiết lập nhằm giải quyết vấn đề trên bằng các kết nối thần kinh tái phát. Tuy nhiên, đối với bất kỳ kiến trúc RNN tiêu chuẩn nào, ảnh hưởng của một đầu vào nhất định lên các lớp ẩn và cuối cùng là đầu ra mạng thần kinh sẽ phân rã hoặc nổ tung theo cấp số nhân khi quay vòng các kết nối lặp đi lặp lại. Để giải quyết vấn đề này LSTM, GRU và các phức hợp đã được thiết kế mang tính cách mạng bằng cách thay đổi cấu trúc của các tế bào thần kinh ẩn trong RNN truyền thống. Ngày nay, nghiên cứu và ứng dụng của mô hình LSTM, GRU và các phức hợp để dự đoán thị trường chứng khoán còn rất hạn chế. Một vài nghiên cứu lý thuyết về hai mô hình trên và các phức hợp của nó như sau: Shejul et al., (2023), với nghiên cứu "Dự đoán giá cổ phiếu bằng GRU, SimpleRNN và LSTM" cho thấy kết quả dự có độ chính xác cao trước những biến bất thường của thị trường chứng khoán; Kanzari et al. (2023) với nghiên cứu "Dự đoán bất ổn tài chính vĩ mô - Tâm lý có liên quan như thế

nào? Bằng các mạng nơ-ron hồi quy." Kết quả cho thấy mô hình LSTM có thể dự đoán chính xác xu hướng của thị trường chứng khoán, tuy nhiên mô hình GRU chạy và đào tạo nhanh hơn LSTM, nhưng LSTM chính xác hơn; Salimath et al. (2021), dự đoán giá cổ phiếu cho thị trường chứng khoán Ấn Độ bằng sử dụng hai mô hình LSTM và GRU, dự đoán được thực hiện trên giá đóng cửa cổ phiếu của 25 công ty ở Ấn Độ. Kết quả chỉ ra rằng GRU hoạt động tốt hơn tất cả các mạng khác liên quan đến 25 công ty này; Liu et al. (2019) ứng dụng mô hình GRU-LSTM chính quy trong dự đoán giá cổ phiếu. Trong bài báo này, dựa trên các mô hình hiện có, họ đã đề xuất mô hình mạng nơ-ron GRU-LSTM phức hợp và áp dụng nó vào dự báo ngắn hạn về giá đóng cửa của hai cổ phiếu. Kết quả thử nghiệm cho thấy mô hình đề xuất của họ vượt trội so với các mô hình mạng GRU và LSTM hiện có trong dự đoán chuỗi thời gian chứng khoán. Sau cùng phải nói nghiên cứu của Hossain et al. (2018), nghiên cứu ứng dụng mô hình phức hợp GRU-LSTM để dự đoán chỉ số S&P500, họ cho thấy mô hình phức hợp cho hiệu suất khá tốt và ít phát sinh các lỗi cũng như giảm thiểu các hiện tượng quá tải của hệ thống cho quy mô dữ liệu lớn.

Mục tiêu của nghiên cứu này đóng góp vào việc nâng cao hiệu quả và tính chính xác của mô hình dự báo thị trường chứng khoán thông qua việc ứng dụng và đánh giá hiệu suất của các mô hình LSTM, GRU và các phức hợp của chúng sẽ giúp cho nhà đầu tư, các tổ chức tài chính cũng như nhà nghiên cứu có cái nhìn sâu hơn về tính hiệu quả của mô hình máy học trong việc dự báo. Đồng thời với kết quả dự báo xu hướng biến động của chỉ số VNIndex gián tiếp hỗ trợ các nhà quản lý quỹ, các tổ chức tài chính và cá nhân trong quản lý tài sản, các nhà quản lý và hoạch định chính sách đưa ra quyết định trong lĩnh vực tài chính sẽ đồng bộ với các chính sách kinh tế vĩ mô khác. Một thị trường chứng khoán lành mạnh và năng động giúp định giá quỹ cạnh tranh, cho phép các doanh nghiệp phát triển. Một thị trường chứng khoán lành mạnh cũng tạo thành nền tảng của một thị trường tương lai mạnh mẽ và một thị trường quyền chọn sôi động.

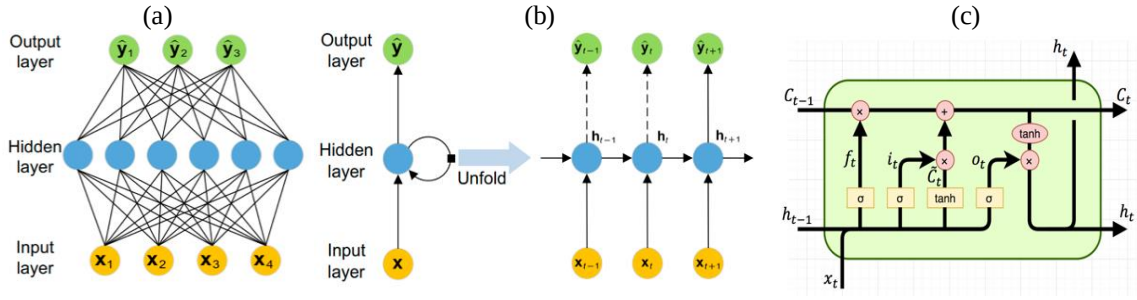
Mô hình nghiên cứu là mạng nơ-ron LSTM, GRU và các phức hợp. Dữ liệu nghiên cứu là chỉ số VNIndex từ ngày 14/7/2008 đến 14/7/2023 với số quan sát là 3747.

2. MÔ HÌNH VÀ PHƯƠNG PHÁP NGHIÊN CỨU

Mô hình

Mô hình LSTM

Mạng nơ-ron nhân tạo (ANN) được xây dựng như một lớp các mô hình học máy có thể loại bỏ những hạn chế của các thuật toán học truyền thống với lập trình dựa trên quy tắc (LeCun et al., 2015). ANN có thể được phân thành hai loại chính mạng nơ-ron truyền thẳng (FFNN - FeedForward Neural Networks) và RNN. FFNN thường bao gồm một lớp đầu vào, một lớp đầu ra và các lớp ẩn (nếu cần thiết). Mỗi lớp bao gồm một số tế bào thần kinh và hàm kích hoạt. Một sơ đồ đơn giản của FFNN được minh họa trong Hình 1(a). Trong khi FFNN, không có kết nối giữa các tế bào thần kinh trong cùng một lớp và tất cả các tế bào thần kinh cũng không thể được kết nối giữa các lớp, có nghĩa là thông tin chảy theo một hướng từ lớp đầu vào qua các lớp ẩn (nếu có) đến lớp đầu ra. FFNN được sử dụng rộng rãi trong các lĩnh vực khác nhau như phân loại dữ liệu, nhận dạng đối tượng và xử lý hình ảnh. Tuy nhiên, do cấu trúc bên trong của chúng nên FFNN không phù hợp để xử lý các phụ thuộc lịch sử.



Hình 1. Minh họa khối bộ nhớ FFNN, RNN và LSTM

Ghi chú: (a) FFNN, (b) RNN, (c) LSTM

(Nguồn: Hua et al., 2019)

RNN như một loại ANN khác, tương tự như FFNN trong cấu trúc của các lớp thần kinh, nhưng cho phép các kết nối giữa các tế bào thần kinh trong cùng một lớp ẩn. Một hình minh họa của RNN có thể được quan sát ở phía bên trái của Hình 1(b). Ngoài ra, phía bên phải của Hình 1(b) là dạng mở rộng của mô hình RNN, chỉ ra rằng RNN tính toán đầu ra của thời điểm hiện tại từ đầu vào của thời điểm hiện tại tại $x_t$và trạng thái ẩn của thời điểm trước h_{i-1}. Do đó, RNN cho phép thông tin đầu vào lịch sử được lưu trữ ở trạng thái bên trong của mạng và nó có khả năng ánh xạ tất cả dữ liệu đầu vào lịch sử đến đầu ra cuối cùng. Về mặt lý thuyết, RNN có thẩm quyền xử lý các phụ thuộc tầm xa như vậy. Tuy nhiên, trên thực tế RNN dường như không thể hoàn thành nhiệm vụ. Hiện tượng này đã được khám phá sâu bởi Hochreiter and Schmidhuber (1997), họ đã giải thích một số lý do khá cơ bản tại sao việc học như vậy có thể khó khăn.

Theo Hình 1(b), khi cho một chuỗi vectơ đầu vào $x = (x_1, \dots, x_t)$, kiến trúc RNN tính chuỗi vectơ ẩn $h = (h_1, \dots h_t)$ và chuỗi vectơ đầu ra $y = (\hat{y}_1, \dots \hat{y}_t)$ bằng cách lập các phương trình (1) và (2) sau từ 1 đến t.

$$h_t = \mathcal{H}(\mathcal{W}_{xh}x_t + \mathcal{W}_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = \mathcal{W}_{hy}h_t + b_y \quad (2)$$

Trong đó, $\mathcal{W}$ biểu thị ma trận trọng số (ví dụ: $\mathcal{W}_{hh}$ biểu thị ma trận trọng số ẩn), b biểu thị vectơ thiên vị (ví dụ: b_h biểu thị vectơ thiên vị ẩn) và $\mathcal{H}$ biểu thị hàm kích hoạt mà lớp RNN sử dụng.

Mạng nơ-ron với bộ nhớ ngắn hạn định hướng dài hạn thường chỉ được gọi là LSTM, là một RNN đặc biệt phù hợp để học các phụ thuộc dài hạn. Phần quan trọng giúp tăng cường khả năng của LSTM để mô hình hóa các phụ thuộc dài hạn là một thành phần được gọi là khối bộ nhớ (Hochreiter & Schmidhuber, 1997). Như minh họa trong Hình 1(c), khối bộ nhớ là một mạng con được kết nối thường xuyên có chứa các mô-đun chức năng được gọi là ô nhớ (memory cell) và các cổng (gate). Ô nhớ chịu trách nhiệm ghi nhớ trạng thái thời gian của mạng lưới nơ-ron và các cổng được hình thành bởi các đơn vị nhân có nhiệm vụ kiểm soát mô hình luồng thông tin. Theo các chức năng thực tế tương ứng, các cổng này được phân loại là cổng đầu vào (input gate), cổng đầu ra (output gate) và cổng quên (forget gate). Cổng đầu vào kiểm soát lượng thông tin mới chảy vào ô nhớ, trong khi cổng quên kiểm soát lượng thông tin của ô nhớ vẫn còn trong ô nhớ hiện tại thông qua kết nối lặp lại và cổng đầu ra kiểm soát lượng thông tin được sử dụng để tính toán kích hoạt đầu ra của khối bộ nhớ và tiếp tục chảy vào phần còn lại của mạng nơ-ron.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

$$\varphi(x) = \left(\frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}\right) \quad (4)$$

Trước khi đi qua các chi tiết của LSTM, xét hàm sigmoid, phương trình (3) và hàm tiếp tuyến, phương trình (4) còn gọi là hàm *tanh*, được sử dụng làm hàm kích hoạt trong ANN. Miền của cả hai hàm là trường số thực, nhưng giá trị trả về cho hàm sigmoid nằm trong khoảng từ 0 đến 1, trong khi hàm *tanh* nằm trong khoảng từ -1 đến 1. Hình 1(c) giải thích chi tiết cách thức LSTM hoạt động. Bước đầu tiên là quyết định loại thông tin nào sẽ bị xóa khỏi trạng thái ô nhớ, được thực hiện bởi một lớp sigmoid (tức là cổng quên). Bước tiếp theo là quyết định thông tin mới nào sẽ được lưu trữ trong trạng thái ô nhớ. Thao tác này có thể được chia thành hai bước con: Đầu tiên, một lớp sigmoid (tức là cổng đầu vào) xác định những gì sẽ được cập nhật và một lớp *tanh* tạo ra một vectơ các giá trị ứng cử viên mới $\tilde{C}_t$ có thể được thêm vào trạng thái ô nhớ, nơi chỉ số dưới *t* biểu thị thời điểm hiện tại. Tiếp theo, hai phần này được kết hợp để kích hoạt cập nhật trạng thái ô nhớ. Để cập nhật trạng thái ô nhớ cũ C_{t-1}vào trạng thái ô nhớ mới C_t, trước tiên chúng ta có thể nhân các phần tử tương ứng của $\boldsymbol{C}_{t-1}$ và phần ra của lớp cổng quên (tức là f_t), giống như cơ chế lãng quên trong não người, và sau đó thêm $i_t * \tilde{C}_t$, với i_t biểu thị đầu ra của cổng đầu vào và $*$ biểu thị phép nhân các phần tử theo cặp trong hai ma trận hoặc mảng có cùng kích thước, gọi tắt là phép nhân Hadamard. Bước cuối cùng là quyết định những gì cho đầu ra, đó là hiện thực hóa bằng phép nhân Hadamard giữa giá trị thu được từ một hàm *tanh* của C_t và đầu ra của một lớp sigmoid (tức là cổng đầu ra) o_t. Thông qua sự hợp tác giữa ô nhớ và cổng, LSTM có một khả năng mạnh mẽ để dự đoán chuỗi thời gian với sự phụ thuộc dài hạn.

Sự khác biệt giữa LSTM và RNN nằm ở trạng thái ẩn. Trong khi RNN sử dụng trạng thái ẩn thông thường, LSTM sử dụng ô nhớ cụ thể được minh họa ở Hình 1(c) chứa nhiều tham số hơn và đơn vị hệ thống cổng điều khiển luồng thông tin. Hệ thống cổng này làm cho LSTM có thể khắc phục điểm yếu RNN, chẳng hạn như vấn đề bộ nhớ dài và vấn đề "gradient biến mất" (vanishing gradient) so với RNN gốc, do đó làm cho hiệu suất LSTM tốt hơn RNN. Phương trình mà ô nhớ LSTM sử dụng được viết dưới dạng phương trình đệ quy như sau:

$$i_t = \sigma(\mathcal{W}_i[h_{t-1}, x_t] + b_i) \quad (5)$$

$$f_t = \sigma(\mathcal{W}_f[h_{t-1}, x_t] + b_f) \quad (6)$$

$$\check{C}_t = \varphi(\mathcal{W}_{\check{c}}[h_{t-1}, x_t] + b_{\check{c}}) \quad (7)$$

$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \quad (8)$$

$$o_t = \sigma(\mathcal{W}_o[h_{t-1}, x_t] + b_o) \quad (9)$$

$$h_t = o_t * \varphi(c_t) \quad (10)$$

Trong đó σ biểu thị hàm sigmoid, φ biểu thị hàm tanh; $\mathcal{W}_i, \mathcal{W}_f, \mathcal{W}_{\check{c}}, \mathcal{W}_o$ biểu thị ma trận trọng số của $i_t, f_t, \check{C}_t, o_t$; $b_i, b_f, b_{\check{c}}, b_o$ biểu thị các vectơ thiên vị của $i_t, f_t, \check{C}_t, o_t$; $\check{C}_t$là một đề xuất mới của ô nhớ; x_t giá trị đầu vào tại thời điểm t; h_{t-1}là vectơ ẩn của trạng thái t-1 và i_t, f_t, C_t, o_t, h_t lần lượt là cổng vào, cổng quên, ô nhớ, cổng ra và vectơ ẩn trên trạng thái t.

Mô hình GRU

Mô hình GRU hay còn gọi là mạng nơ-ron hồi tiếp với nút cổng, được giới thiệu bởi Cho et al. (2014). GRU có thể xem là một biến thể thế hệ mới của LSTM, về cấu trúc GRU đơn giản chỉ với 02 cổng: Cổng khởi tạo (reset gate) và cổng cập nhật (update gate). Với thiết kế đơn giản như vậy mô hình GRU giảm số lượng tham số so với LSTM, điều này làm cho nó nhanh hơn trong việc huấn luyện và dự báo. GRU phù hợp với tập dữ liệu nhỏ, đòi hỏi nhanh hơn trong quá trình huấn luyện và giảm thiểu nguy cơ quá tải.

Nguyên tắc hoạt động của GRU: Cổng khởi tạo r_t kiểm soát mức độ hợp nhất thông tin trạng thái hiện tại với thông tin trạng thái trước đó và cổng cập nhật z_t (sự kết hợp của cổng đầu vào và cổng quên) kiểm soát xem thông tin trạng thái tại thời điểm trước đó có được giữ lại hay không và mức độ lưu giữ ở trạng thái hiện tại. Nếu đầu ra tại thời điểm trước đó là h_{t-1} và đầu vào tại thời điểm hiện tại là x_t, quá trình tính toán đơn vị lớp ẩn h_t dựa trên GRU được viết dưới dạng phương trình đệ quy như sau:

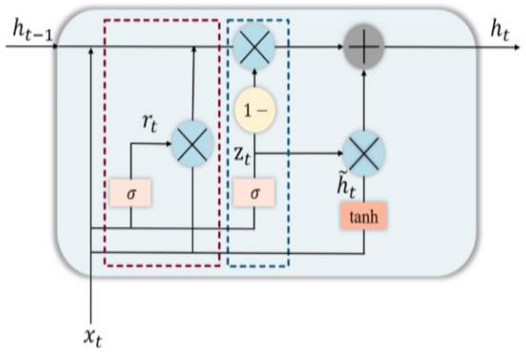
$$r_t = \sigma(\mathcal{W}_r[h_{t-1}, x_t] + b_r) \quad (11)$$

$$z_t = \sigma(\mathcal{W}_z[h_{t-1}, x_t] + b_z) \quad (12)$$

$$\check{h}_t = \varphi(\mathcal{W}_{\check{h}}[r_t * h_{t-1}, x_t] + b_{\check{h}}) \quad (13)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \check{h}_t \quad (14)$$

Trong đó $\mathcal{W}_r, \mathcal{W}_z, \mathcal{W}_{\check{h}}$ lần lượt là ma trận trọng lượng của $r_t, z_t, \check{h}_t$. $b_r, b_z, b_{\check{h}}$ biểu thị các vectơ thiên vị của $r_t, z_t, \check{h}_t$ tương ứng. Đầu tiên, trạng thái lớp ẩn không thể giữ lại tại thời điểm hiện tại được điều khiển bởi cổng đặt lại. Thứ hai, trạng thái lớp ẩn còn lại tại thời điểm hiện tại được xác định bởi cổng cập nhật. Cuối cùng, trạng thái ứng cử viên lớp ẩn được tính theo trạng thái lớp ẩn của thời điểm trước đó và đầu vào của thời điểm hiện tại, và trạng thái lớp ẩn chuyển tiếp của đầu ra thời điểm hiện tại thu được bằng cách kết hợp đầu ra của thời điểm trước đó. GRU được đào tạo để có khả năng thực hiện một số công việc dự báo, có thể làm giảm nguy cơ quá tải do số lượng tham số ít.



Hình 2. Minh họa khối nhớ mô hình GRU

(Nguồn: Liu et al.,2019)

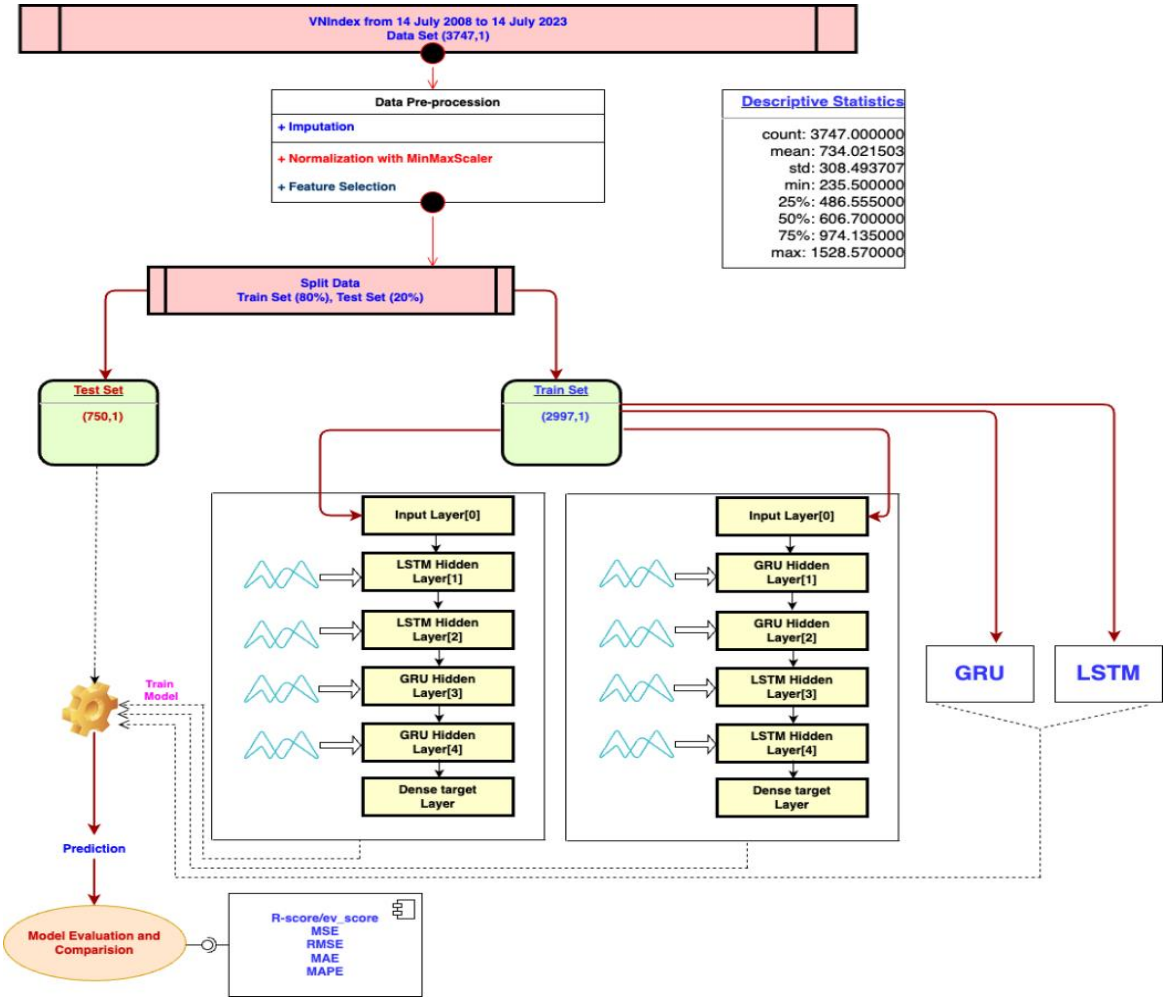
Mô hình GRU-LSTM phức hợp

Theo thiết kế ở Hình 3, mô hình phức hợp của nghiên cứu này là sử dụng 02 lớp ẩn đầu tiên của mô hình LSTM hoặc GRU làm 02 lớp ẩn đầu vào của một mô hình tổng quát. Bài nghiên cứu này sử dụng thuật ngữ LSTM-GRU Hybrid chỉ cách sử dụng 02 lớp ẩn của LSTM làm đầu vào của mô hình và GRU-LSTM Hybrid là trường hợp ngược lại.

Tối ưu hóa các tham số

Để tối ưu hóa tốc độ cập nhật của các tham số mô hình LSTM và GRU, tốc độ học tập cần được điều chỉnh. Sau khi xem xét nhiều thuật toán tối ưu hóa dựa trên độ dốc, thuật toán Ước tính khoảnh khắc thích ứng (Adam - Adaptive Moment Estimation) đã được chứng minh là đạt được hiệu suất tổng thể tốt hơn trong các ứng dụng thực tế. Thuật toán Adam (Kingma et al., 2014) là kết hợp các ưu điểm của thuật toán Ada Grad và RMS Prop (Duchi et al., 2011) được sử dụng để ước tính thời điểm bậc nhất và ước tính thời điểm bậc hai của gradient để điều chỉnh động tốc độ học tập của từng tham số, giúp cập nhật tham số ổn định hơn và chiếm ít tài nguyên lưu trữ hơn (Daneshvar et al., 2022). Thuật toán Adam không chỉ lưu trữ trung bình bình phương các gradient trước đó mà còn lưu cả giá trị trung bình mô-men m_t. Các giá trị $m_t$và v_t được tính bởi công thức (15) và (16), trong đó β_1 và β_2 là

các trọng số không âm, thường được chọn là β_1 = 0,9 và β_2= 0,999, g_t là gradient của hàm mất mát tại bước t. Nếu khởi tạo m_t và v_t là các vectơ 0, các giá trị này có khuynh hướng nghiêng về 0, đặc biệt là khi β_1 và β_2 xấp xỉ bằng 1. Do vậy, để khắc phục, các giá trị này được ước lượng bằng công thức (17) và (18)



Hình 3. Lưu đồ quy trình nghiên cứu mô hình máy học

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t \quad (15)$$

$$v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2 \quad (16)$$

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t} \quad (17)$$

$$\widehat{v_t} = \frac{v_t}{1 - \beta_2^t} \quad (18)$$

$$\theta_{t+1} = \theta_t - \frac{\widehat{m_t}}{\sqrt{\widehat{v_t} + \varepsilon}} * \eta \quad (19)$$

Trong công thức (19), θ_{t+1}, θ_tlà ma trận trọng số lần lược tại bước t và t+1; ε là một số dương khá bé nhằm tránh trường hợp mẫu số bằng 0, thường bằng 10^{-8} và η là số dương được gọi là tốc độ học tập của mô hình.

Các chỉ số đánh giá

Báo cáo này sử dụng hệ số R^2 ($r2\_score$ - R-squared score), hệ số giải thích phương sai (ev_score – Explained Variance Regression Score) để đánh giá về độ phù hợp của mô hình trong học máy, công thức (20) và (21). Giá trị $r2\_score$ và ev_score đều nằm trong khoảng từ 0 đến 1 và cho kết quả gần tương đương nhau, nếu giá trị của nó càng gần 1 thì mô hình dự báo càng hiệu quả.

$$r2_{score} = \frac{\sum_{i=1}^{n}|y_i - \bar{y}|^2 - \sum_{i=1}^{n}|y_i - \hat{y}_i|^2}{\sum_{i=1}^{n}|y_i - \bar{y}|^2} \quad (20)$$

$$ev\_score = \left[\frac{n\sum_{i=1}^{n} y_i\hat{y}_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} \hat{y}_i}{\sqrt{[n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2]\, x\, [n\sum_{i=1}^{n} \hat{y}_i^2]}}\right]^2 \quad (21)$$

Tiếp theo, báo cáo này sử dụng Gamma Deviance và Poisson Deviance để tính toán độ sai lệch trung bình giữa giá trị dự đoán của mô hình và giá trị thực tế trong bài toán dự đoán với phân phố Gamma và Poisson để so sánh hiệu suất giữa các mô hình. Ngoài ra, báo cáo này còn sử dụng sai số bình phương trung bình (MSE), sai số bình phương trung bình gốc (RMSE), sai số tuyệt đối trung bình (MAE) và sai số phần trăm tuyệt đối trung bình (MAPE) để đánh giá hiệu quả của mô hình. Các công thức được mô tả như sau:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2 \quad (22)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2} \quad (23)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (24)$$

$$MAPE = \sum_{i=1}^{n}\left|\frac{|y_i - \hat{y}_i|}{y_i}\right| x \frac{100}{n} \quad (25)$$

$$Gamma\ Deviance = -2\sum_{i=1}^{n}[y_i \log(\hat{y}_i) - \hat{y}_i - y_i \log(y_i) + \Gamma(y_i)] \quad (26)$$

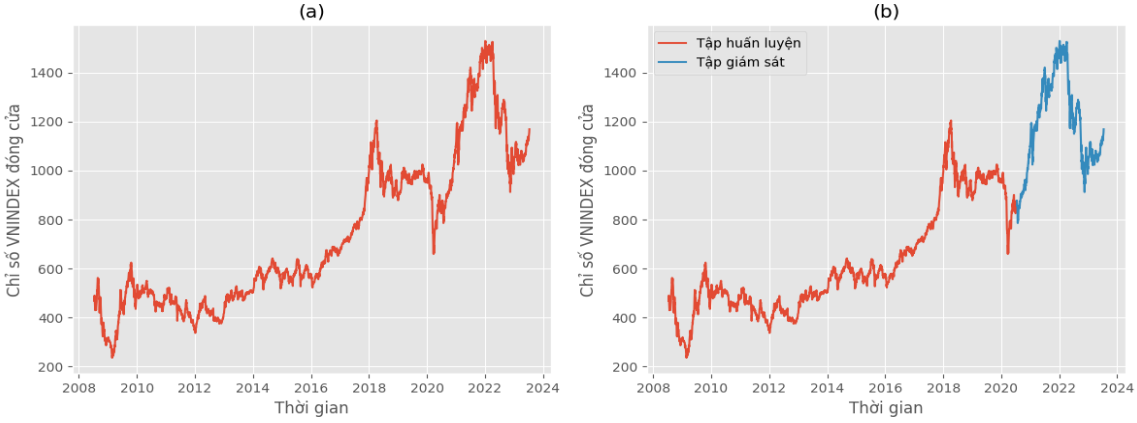
$$Poisson\ Deviance = 2\sum_{i=1}^{n}\left[y_i \log\left(\frac{y_i}{\hat{y}_i}\right) + \hat{y}_i - y_i\right] \quad (27)$$

Trong đó, n là tổng số quan sát tham gia thử nghiệm; y_i là giá trị thực của ngày giao dịch thứ i; $\hat{y}_i$ là giá trị dự đoán của giá giao dịch ngày thứ i; $\bar{y}$ là giá trị trung bình của giao dịch thực tế; $\Gamma(.)$ hàm gamma.

Phương pháp nghiên cứu

Thu thập dữ liệu

Nghiên cứu này chọn giá trị chỉ số VNIndex giao dịch lúc đóng cửa hàng ngày của thị trường chứng khoán HoSE làm biến số ủy quyền của giá giao dịch chứng khoán làm bộ dữ liệu nghiên cứu chính. Tất cả dữ liệu được lấy từ cơ sở dữ liệu của M.S Fusion Media Ltd (https://www.investing.com/indices/vn-historical-data) và kéo dài từ ngày 14 tháng 7 năm 2008 đến 14 tháng 7 năm 2023 với tổng số 3747 bộ dữ liệu. Trong mô hình độc lập LSTM, GRU và các phức hợp của nó, bộ dữ liệu huấn luyện chứa 2997 quan sát, chiếm 80% tập dữ liệu gốc và bộ dữ liệu giám sát chứa 750 quan sát, chiếm 20% cuối cùng của tập dữ liệu gốc, cụ thể biểu đồ đặc tả xu hướng biến động chỉ số VNIndex trước - Hình 4(a) và sau khi chia tập dữ liệu chính - Hình 4(b).



Hình 4. Biểu đồ xu hướng biến động chỉ số VNIndex từ ngày 14/7/2008 đến 14/7/2023

Mô hình đào tạo

Quy trình xây dựng của mô hình dự đoán LSTM, GRU và các phức hợp của nó được tối ưu hóa bởi thuật toán Adam như sau:

Giai đoạn 1: Tiền xử lý dữ liệu

Bước 1: Chuẩn hóa dữ liệu

Do tính không đồng đều về kích thước số của dữ liệu thô nên phải hiệu chỉnh lại bộ dữ liệu về một

phạm vi nhất định, nghiên cứu này hạn chế tất cả các giá trị trong phạm vi từ 0 đến 1, nhằm nâng cao hiệu quả học tập của dữ liệu trong quá trình học tập và dự đoán, giảm lỗi, và đảm bảo hiệu quả của gradient gốc. Phương pháp chuẩn hóa tối thiểu-tối đa (min-max) được sử dụng để cung cấp dữ liệu chuỗi thời gian giá giao dịch chứng khoán bằng cách sử dụng công thức chuẩn hóa min-max (Guo et al., 2020) như sau:

$$Z_i = \frac{x_i - Min(x)}{Max(x) - Min(x)} \quad (28)$$

Trong đó, Z_i là giá trị sau khi được chuẩn hóa ở trạng thái i và có giá trị từ 0 đến 1, x_i là giá trị quan sát ở trạng thái i trong tập dữ liệu x với i=1…n, $Min(x)$ và $Max(x)$ biểu thị giá trị nhỏ nhất và lớn nhất của tập dữ liệu x tương ứng.

Bước 2:

Chuyển đổi dữ liệu thành dữ liệu có thể được sử dụng cho việc huấn luyện có giám sát. Dựa trên các đặc điểm dữ liệu của mạng nơ-ron LSTM và GRU, dữ liệu được chuẩn hóa sẽ được chia thành 02 tập, sử dụng trực tiếp cho việc máy học gọi tắt là tập huấn luyện (training set) và tập giám sát (test set). Do các điểm mẫu dữ liệu, nghiên cứu này trực tiếp áp dụng phương pháp dự đoán lùi một bước. Ngoài ra, sau khi huấn luyện nhiều lần về mô hình và điều chỉnh tham số, các nghiên cứu trước thấy rằng kích thước bước thời gian (time step) nên được giảm càng nhiều càng tốt khi có ít mẫu dữ liệu (Su et al., 2021). Kích thước bước thời gian quyết định số bước thời gian trong một chuỗi dữ liệu thời gian mà mô hình sẽ sử dụng để thực hiện dự báo. Do mục tiêu của mô hình, tính chất của dữ liệu và tài nguyên máy tính hiện có nghiên cứu này chọn kích thước bước thời gian là 50.

Giai đoạn 2: Huấn luyện mô hình

Ở giai đoạn này, kích thước lớp ẩn và số lượng tối ưu hóa tế bào nơ-ron đã được thực hiện để tạo ra mô hình LSTM, GRU và các phức hợp của nó được sử dụng trong dự đoán chỉ số VNIndex. Số lượng tế bào nơ-ron cao làm tăng thời gian tính toán và yêu cầu bộ nhớ. Mặt khác, một số lượng tế bào thần kinh thấp gây ra sự thiếu phù hợp. Khi số lượng lớp tăng lên trong các thuật toán học sâu, mô hình học tốt hơn, nhưng sự lan truyền ngược đến các lớp đầu tiên ít hơn (Szandała, 2021). Các thông số khác cho mô hình được phát triển là tốc độ học tập, hàm kích hoạt, trình tối ưu hóa, số lần mô hình được huấn luyện (epoch), kích thước lô và hàm mất.

Trong nghiên cứu này, mô hình mạng nơ-ron ba lớp được sử dụng để đào tạo các mô hình đơn và mạng nơ-ron năm lớp cho mô hình phức hợp, kích thước ô nhớ (batch size) là 50 và một lớp mạng nơ-ron thông thường được thêm vào lớp ẩn để giảm kích thước của kết quả đầu ra. Hàm $tanh$ được sử dụng làm hàm kích hoạt trong quá trình đào tạo mô hình. Đồng thời, để giảm hiện tượng quá khớp (overfitting) hay còn gọi là mô hình được đào tạo quá tốt trên tập dữ liệu huấn luyện, đạt hiệu suất cao nhưng không tổng quát hóa tốt cho tập dữ liệu mới, tỷ lệ từ chối (dropout) của mỗi lớp nút mạng được đặt lần lược là 0,25, 0,35, 0,45 và 0,5. Do đó, nghiên cứu sử dụng hệ số MAE, phương trình (25) làm hàm lỗi và thuật toán Adam làm phương pháp cập nhật lặp lại của các tham số trọng lượng. Biểu thức của hàm mất (loss) được định nghĩa như sau:

$$loss = \frac{\sum_{i=1}^{L(m-L)} (p_i - y_i)^2}{L(m-L)} \quad (29)$$

Trong đó, p_i, y_i tương ứng là giá trị dự đoán và giá trị thực tế; m là số lượng lớp của mô hình; L là số lượng cặp giá trị (p_i, y_i); $L(m-L)$ là tham số phụ thuộc vào số lượng lớp và số lượng mẫu dữ liệu. $L(m-L)$ biểu thị số lượng cặp giá trị (p_i, y_i) trong một mẻ hoặc tổng số lượng cặp giá trị (p_i, y_i) trong toàn bộ tập dữ liệu. Mục tiêu của hàm loss là tối thiểu hóa sai số giữa giá trị dự đoán và giá trị thực tế. Trong quá trình huấn luyện mô hình, tham số mô hình sẽ được điều chỉnh để giảm thiểu giá trị loss và cải thiện hiệu suất mô hình.

Sau cùng, xem xét giới hạn của các điểm lấy mẫu dữ liệu, số lần huấn luyện và các kích thước kết hợp hàng loại để đào tạo mô hình, nghiên cứu cho số lần huấn luyện lần lượt là 50 và 100. Do mục tiêu nghiên cứu cần một lớp đầu ra duy nhất (dense) để nối kết các mối nơ-ron với tất cả các nơ-ron trong lớp đó với lớp sau đó, cũng như giảm thiểu nguy cơ quá tải của hệ thống nên đặt Dense=1.

Giai đoạn 3: Áp dụng mô hình dự đoán

Mô hình được đào tạo ở trên sử dụng để mô phỏng xu hướng biến động chỉ số VNIndex của thị trường chứng khoán Hồ Chí Minh sẽ được sử dụng để dự đoán trên tập dữ liệu giám sát và dự báo tiếp theo.

Giai đoạn 4: Đánh giá hiệu suất mô hình

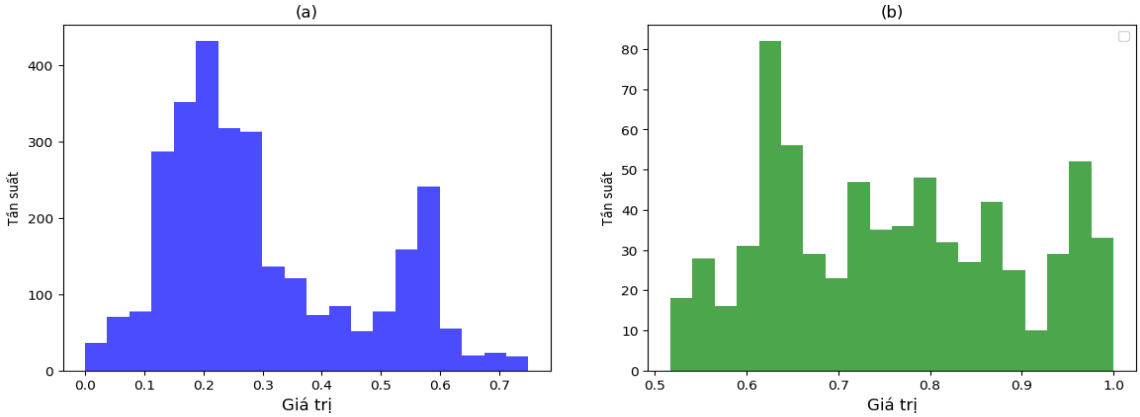
Sử dụng các chỉ số thống kê như $r2\_score$, ev_score, Gamma Deviance, Poisson Deviance, MSE, RMSE, MAE và MAPE để đánh giá độ phù hợp, hiệu suất và hiệu quả của mô hình.

Giai đoạn 5: Ứng dụng mô hình có hiệu suất tốt nhất để dự báo xu hướng của thị trường chứng khoán với 100 ngày tiếp theo.

Trong nghiên cứu này, các mô hình máy học và cơ sở đề xuất các siêu tham số được triển khai bằng cách sử dụng các gói của ngôn ngữ lập trình Python, bao gồm Keras (phiên bản 2.6.0), gói phụ trợ Tensorfow và Scikit-learn (Pedregosa et al., 2011; Chollet, 2015; Abadi et al., 2016), matplotlib và seaborn để trực quan hóa kết quả (Hunter, 2007; Waskom, 2021), Papermill, thuật toán tối ưu Bayesian (de Freitas et al., 2016). Khối lượng công việc nặng hơn được chạy trên Google Colab được trang bị GPU Tesla T4 của NVIDIA và VRAM GDDR5 16GB. Phần còn lại của các mô hình được thực thi trên máy tính có cấu hình 02 CPU Intel (R) Xeon (R) X5670 @ 2.93 GHz 2.93 GHz , RAM 96,0 GB.

3. KẾT QUẢ VÀ THẢO LUẬN

Do yêu cầu về thuật toán, nghiên cứu thực hiện chuẩn hóa tập dữ liệu nghiên cứu cho mô hình bằng phương pháp Min-Max. Hình 5(a) cho thấy các giá trị của tập huấn luyện hội tụ nhanh ở hai phân đoạn (0,1:0,3) và (0,5:0,7), điều này cho thấy việc chia tập dữ liệu huấn luyện là tương đối phù hợp đồng thời giảm thiểu các vấn đề về gradient khi huấn luyện; Hình 5(b) cho thấy việc các giá trị hội tụ ở phân đoạn (0,5:1,0). Phân bố trên hai tập dữ liệu phù hợp với yêu cầu là trong khoảng (0:1,0) và không có hiện tượng mất dữ liệu; Việc phân chia hai tập dữ liệu cũng được đảm bảo thành hai phần riêng biệt là tập huấn luyện dùng huấn luyện mô hình, tập giám sát dùng để kiểm tra hiệu suất dự báo.



Hình 5. Biểu đồ phân bố của tập dữ liệu sau chuẩn hóa

Ghi chú: (a) Tập huấn luyện, (b) Tập giám sát

Bằng phương pháp thực nghiệm điều chỉnh lần lược số nơ-ron từ 32, 64 và 128 cho các mô hình LSTM, GRU và mô hình phức hợp, nghiên cứu nhận thấy số nơ-ron và các tham số cài đặt như Bảng 1 và Bảng 2 là phù hợp với tập dữ liệu nghiên cứu. Tuy nhiên, nghiên cứu này cũng cho thấy số nơ-ron của mô hình phức hợp cài đặt ở mức tối đa 64 là phù hợp và các tham số của mô hình LSTM-GRU Hybrid là thấp nhất so với 03 mô hình còn lại.

Xét về tỷ lệ từ chối của mỗi lớp nút mạng, nghiên cứu cũng nhận thấy nếu giảm nhỏ hơn 0,25 thì hệ thống máy tính bị hiện tượng quá tải; nhưng khi tăng lên càng cao thì dẫn đến hiện tượng mất dữ liệu dẫn đến kết quả dự báo sau cùng lại kém hiệu quả, không đáp ứng yêu cầu của nghiên cứu này

Bảng 1 và Bảng 2 cho thấy tổng các tham số của các mô hình giảm nhanh qua lần lược các mô hình LSTM, GRU, GRU-LSTM Hybrid và LSTM-GRU Hybrid. Trong đó LSTM có tổng số tham số lớn nhất và LSTM-GRU Hybrid có tổng tham số bé nhất. Việc giảm số tham số và chia nhỏ các lớp đầu vào nối tiếp, giảm thiểu nguy cơ quá tải hệ thống máy tính, giúp phương pháp máy học ngày càng được sử dụng rộng rãi hơn.

Bảng 1. Tóm tắt các thông số mô hình đơn

(a)			(b)		
Layer (type)	***Output Shape***	***Param #***	***Layer (type)***	***Output Shape***	***Param #***
lstm (LSTM)	(None, 50, 128)	66560	gru (GRU)	(None, 50, 128)	50304
lstm_1 (LSTM)	(None, 64)	49408	gru_1 (GRU)	(None, 64)	37248
dropout (Dropout)	(None, 64)	0	dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 1)	65	dense (Dense)	(None, 1)	65
Total params: 116033			*Total params: 87617*		
Trainable params: 116033			*Trainable params: 87617*		
Non-trainable params: 0			*Non-trainable params: 0*		

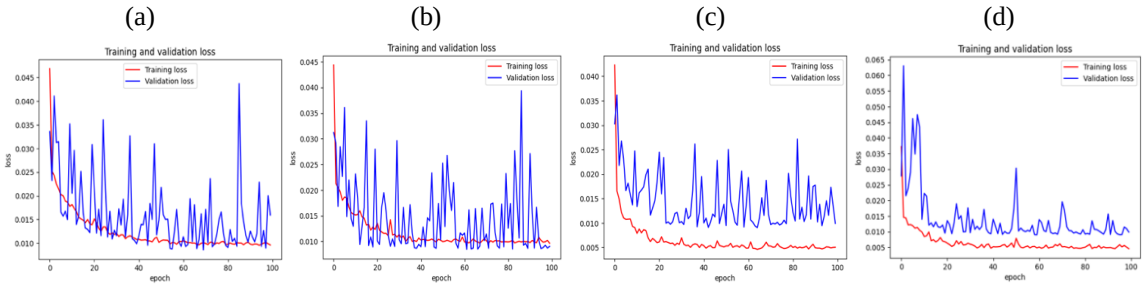
Ghi chú: (a) Mô hình GRU, (b) Mô hình LSTM

Bảng 2. Tóm tắt các thông số mô hình phức hợp

(a)			(b)		
Layer (type)	***Output Shape***	***Param #***	***Layer (type)***	***Output Shape***	***Param #***
gru (GRU)	(None, 50, 64)	12864	lstm (LSTM)	(None, 50, 64)	16896
gru_1 (GRU)	(None, 50, 32)	9408	lstm_1 (LSTM)	(None, 50, 32)	12416
lstm (LSTM)	(None, 64)	24832	gru (GRU)	(None, 64)	18816
lstm_1 (LSTM)	(None, 32)	12416	gru_1 (GRU)	(None, 32)	9408
dense (Dense)	(None, 1)	33	dense (Dense)	(None, 1)	33
Total params: 59553			*Total params: 57569*		
Trainable params: 59553			*Trainable params: 57569*		
Non-trainable params: 0			*Non-trainable params: 0*		

Ghi chú: (a) Mô hình GRU-LSTM, (b) LSTM-GRU

Trong quá trình đào tạo mô hình, sự hội tụ các giá trị giao dịch sau khi tiền xử lý dữ liệu được phân tích, như thể hiện ở Hình 6. Kết quả cho thấy giá trị của hàm tổn thất được đào tạo sau khi tiền xử lý dữ liệu giảm nhanh chóng. Nếu xét riêng cho các mô hình đơn cho thấy hội tụ sau 60 lần lặp, giá trị hàm tổn thất tương đương 0,010. Và các mô hình phức hợp hội tụ sau 40 lần lặp lại, giá trị hàm tổn thất tương đương 0,005, điều này cho thấy tất cả các mô hình phức hợp học nhanh hơn các mô hình đơn.



Hình 6. Biểu đồ so sánh của hàm loss

Ghi chú: (a) LSTM, (b) GRU, (c) GRU-LSTM, (d) LSTM-GRU

Qua thực nghiệm này, nghiên cứu cho thấy có thể giảm số lần huấn luyện cho các mô hình phức hợp từ 100 xuống 50 lần nhằm giảm thời gian thực thi trên máy tính, điều đó góp phần làm giảm các chi phí cho các nghiên cứu tiếp theo. Ngoài ra, sau lần huấn luyện thứ 80 Hình 6(a), (b) giá trị hàm mất mát trên tập kiểm tra có hiện tượng tăng bất thường, thậm chí tăng cao hơn số lần huấn luyện đầu tiên. Vấn đề này trái với quy luật của máy học và sẽ được làm rõ hơn ở các phần tiếp theo của nghiên cứu này.

Bảng 3. Đo lường độ chính xác trên tập huấn luyện

Thông số	*LSTM(0:18:50)*	*GRU(0:13:06)*	*GRU-LSTM(0:14:21)*	*LSTM-GRU(0:12:28)*
r2_score	0,99794	0,99866	0,99833	0,99868
ev_score	0,99801	0,99867	0,99868	0,99869
Gamma Deviance	0,00024	0,00017	0,00021	0,00017
Poisson Deviance	0,13842	0,09588	0,11958	0,09361
MSE	93,70231	60,83519	76,14630	59,88852
RMSE	9,68000	7,79969	8,72618	7,73877
MAE	7,16149	5,47041	6,34235	5,37542
MAPE	0,01681	0,00946	0,01093	0,00926

Quá trình huấn luyện được thực thi trên Google Colab, có thể nói đây là một hạn chế của nghiên cứu này. Thực tế nhóm nghiên cứu thực nghiệm nhiều lần trên Google Colab cho thời gian thực thi của các mô hình là không giống nhau, nguyên nhân có thể là do tốc độ đường truyền internet, số lượng người đang sử dụng máy chủ Google Colab. Tuy nhiên để ước lượng thời gian thực thi của từng mô hình, nhóm nghiên cứu thực thi lệnh cho bốn mô hình trên cùng một thời gian, cùng một đường truyền internet. Qua Bảng 3, thời gian (mang tính tương đối) của mô hình LSTM có thời gian thực thi chậm nhất, nguyên nhân là do mô hình có số tham số lớn nhất so với các mô hình còn lại. Tuy nhiên nghiên cứu cũng cho thấy ngoài số tham số ảnh hưởng đến thời gian thực thi mà yếu tố lớp nơ-ron, cụ thể hơn quá trình truyền dẫn giữa các lớp nơ-ron. Số lớp nơ-ron tăng lên, dù tham số có giảm lại thì quá trình thực thi mô hình trên hệ thống máy tính cũng bị chậm lại. Nhóm nghiên cứu đưa ra kết luận này là dựa vào áp dụng thực tế, chưa có đầy đủ cơ sở khoa học để kết luận chính xác, do vậy cần phải nghiên cứu sâu về khoa học máy tính và máy học mới đảm bảo tính chính xác. Kết quả đo lường ở Bảng 3 trên tập huấn luyện cho thấy tất cả mô hình đều cho độ phù hợp lớn hơn 99%, điều này làm cơ sở cho các thực nghiệm tiếp theo của nghiên cứu này. Với sự phù hợp trên, tiến hành đo lường các thông số thống kê khác, nghiên cứu nhận thấy mô hình LSTM-GRU Hybrid có các chỉ số đo lường thống kê tốt nhất.

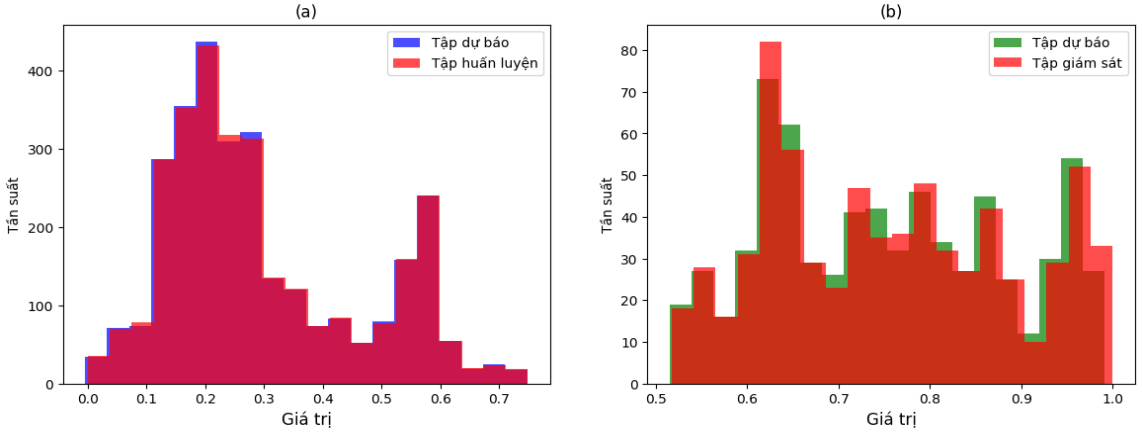
Bảng 4. Đo lường độ chính xác trên tập giám sát

Thông số	*LSTM*	*GRU*	*GRU-LSTM*	*LSTM-GRU*
r2_score	0,98028	0,99052	0,98949	0,99108
ev_score	0,99064	0,99061	0,98960	0,99136
Gamma Deviance	0,00039	0,00019	0,00021	0,00018
Poisson Deviance	0,47032	0,23077	0,25072	0,21672
MSE	578,51989	278,18633	308,34858	261,68583
RMSE	24,05244	17,37117	17,55986	16,17671
MAE	20,57606	11,78603	12,81945	11,69156
MAPE	0,01681	0,00979	0,01050	0,00970

Sau quá trình huấn luyện, nghiên cứu tiến hành đánh giá lại kết quả huấn luyện dựa trên tập dữ liệu giám sát, các thông số đo lường thống kê thể hiện ở Bảng 4. So sánh các thông số thống kê ở Bảng 3 với Bảng 4 cho thấy việc học và việc kiểm tra có một khoảng cách nhất định. Tuy nhiên, qua thông số đo lường thống kê cho thấy mô hình LSTM-GRU Hybrid cho kết quả tốt nhất và mô hình GRU cũng cho kết quả không kém hơn so với mô hình tốt nhất.

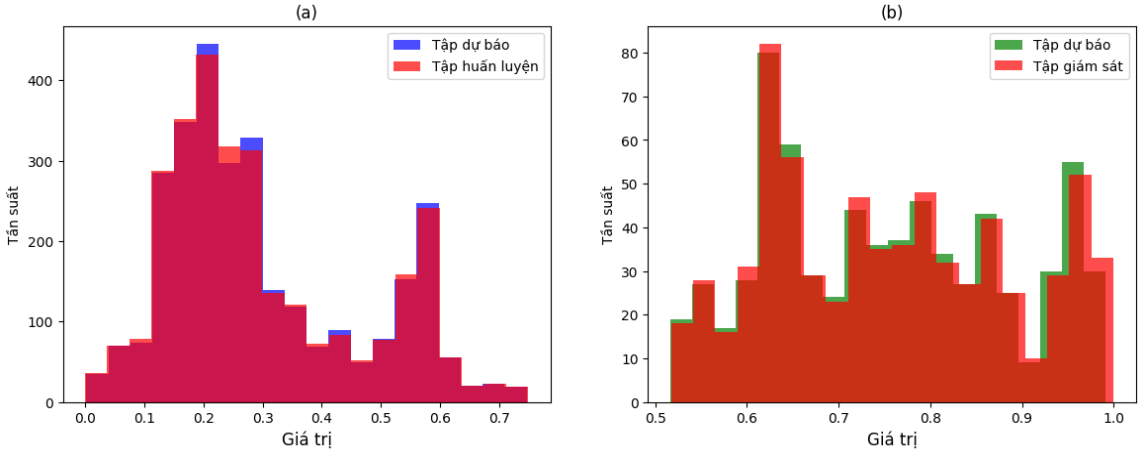
Hình 7(a) và Hình 8(a), các giá trị của tập dự báo trong tập dữ liệu huấn luyện cho thấy sự phân bố của chúng gần như tương khớp với nhau hoàn toàn. Tuy nhiên, quan sát ở Hình 9(a) cho thấy sự phân bố của tập dữ liệu dự báo có tính tương khớp không tốt hơn mô hình GRU. Sau khi quan sát Hình 7(b) và Hình 8(b), kết quả phân bố trên tập dữ liệu giám sát ở mô hình GRU có nhiều khoảng dữ liệu chưa tương khớp hoàn toàn như mô hình LSTM-GRU Hybrid. Vấn đề đặt ra là "tại sao quá trình học rất tốt, nhưng khi kiểm tra lại cho kết quả kém hơn?" rất nhiều nhà nghiên cứu về máy học cho rằng: Nguyên nhân của vấn đề trên là do hiện tượng "quá khớp", và được giải thích rõ bởi Alpaydin (2014). Kết hợp kết quả quan sát hiện tượng này ở Hình 6(b) và Hình 7(a), đối chiếu các nghiên cứu

của Alpaydin (2014) kết luận mô hình GRU xảy ra hiện tượng "quá khớp" là có cơ sở. Nghiên cứu này đã thực nghiệm các giải pháp khắc phục vấn đề trên như giảm số lần huấn luyện (epoch=80), giảm tỷ lệ từ chối (dropout=0,5) của mỗi lớp nút mạng nhưng kết quả cho thấy hiệu suất của mô hình không tăng dù có giải quyết được hiện tượng "quá khớp". Tuy nhiên, nghiên cứu này chưa thực hiện giải pháp chia nhỏ tập huấn luyện, từ đó huấn luyện từng phần và cuối cùng kết hợp các phần nhỏ thành một tập huấn luyện chung. Ngoài ra, Hình 7(b) và Hình 8(b), cho thấy sự phân bố các giá trị dự báo ở tập giám sát hội tụ về đoạn bé hơn 1,0 điều này kéo theo kết quả dự báo khuynh hướng thấp hơn giá trị thực tế.



Hình 7. Biểu đồ so sánh kết quả phân bố của mô hình GRU

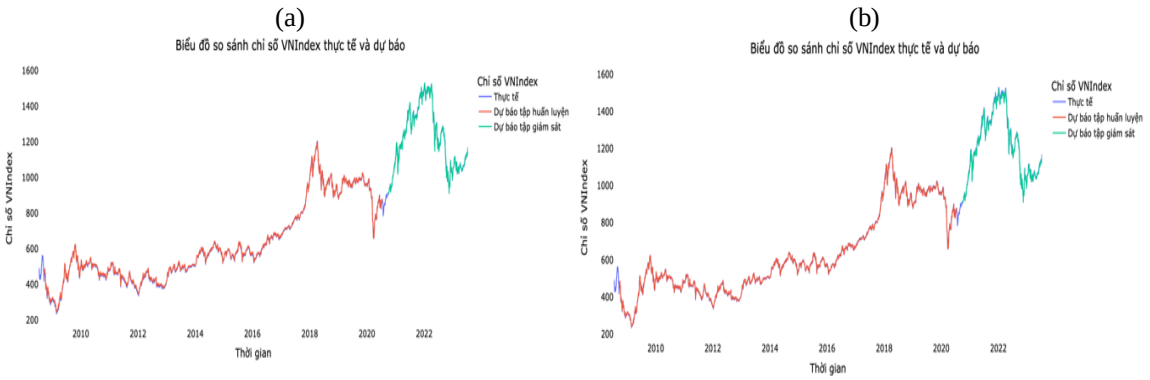
Ghi chú: (a) Tập huấn luyện, (b) Tập giám sát



Hình 8. Biểu đồ so sánh kết quả phân bố của mô hình LSTM-GRU

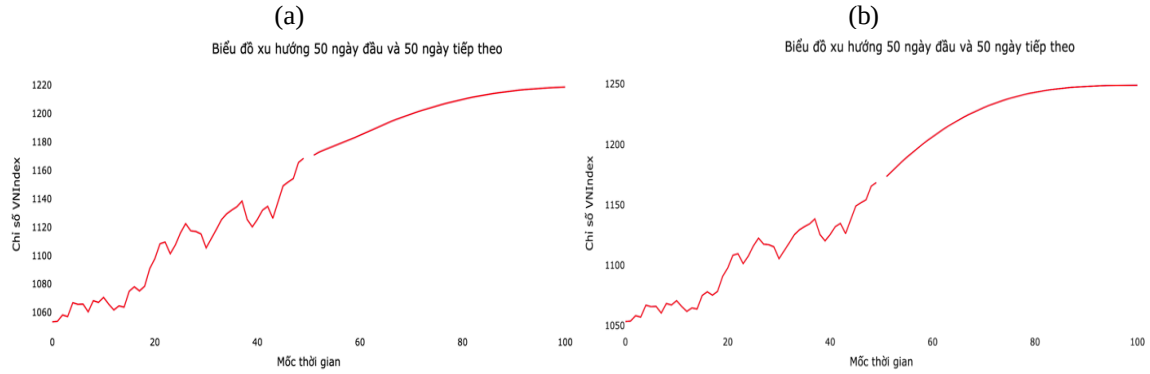
Ghi chú: (a) Tập huấn luyện, (b) Tập giám sát

Xem xét Hình 9, kết quả dự báo so với số liệu thực tế cho thấy số liệu dự báo theo sát xu hướng của dữ liệu thực tế. Trước hết, việc xem xét sự tương quan giữa kết quả thực tế trên tập dữ liệu huấn luyện và giám sát cho thấy có sự tương đồng đáng kể giữa hai tập dữ liệu dự báo và thực tế, điều này có thể chỉ ra rằng mô hình có khả năng tổng quát hóa tốt. Thứ hai, việc xét hiện tượng "quá khớp" cả hai mô hình cho thấy kết quả dự báo trên tập huấn luyện đều có khả năng xảy ra hiện tượng quá khớp. Tuy nhiên, khi quan sát trên tập giám sát của mô hình GRU cho thấy có hiện tượng học tốt nhưng ứng dụng không tốt xảy ra, qua kết quả đo lường độ chính xác của tập dữ liệu giám sát của mô hình này phải ánh đúng thực tế của mô hình GRU. Thứ ba, xét về sự biến đổi quá mức về độ lớn của hai tập dự liệu dự báo và thực tế cho thấy không xảy ra hiện tượng này và kết quả dự báo rất ổn định. Sau cùng, nhận xét về sự tương quan về thời gian, cũng như khả năng tổng quát hóa của mô hình nghiên cứu là hoàn toàn hợp lý.



Hình 9. Biểu đồ so sánh kết quả thực tế trên tập dữ liệu huấn luyện và giám sát

Ghi chú: (a) LSTM-GRU, (b) GRU



Hình 10. Biểu đồ dự báo 100 ngày

Ghi chú: (a) LSTM-GRU Hybrid, (b) GRU

Theo kết quả đo lường các thống kê cho thấy cả hai mô hình trên không có sự khác biệt lớn, do vậy nghiên cứu này áp dụng đồng thời hai mô hình để dự báo xu hướng biến động của chỉ số VNIndex với thời gian dự báo liên tiếp là 50 ngày đầu tiên và 50 ngày kế tiếp.

Biểu đồ dự báo Hình 10 cả hai mô hình, cho thấy 50 ngày đầu tiên chỉ số VNIndex tăng nhanh, 50 ngày tiếp theo có xu hướng tăng chậm lại và sau đó đi ngang. Giá trị tới hạn của chỉ số VNIndex chỉ đạt ở mức 1220 đến 1250, điều này cho thấy hiệu quả các chính sách của Chính phủ còn nhiều hạn chế; đồng thời, chịu ảnh hưởng chung của sự khủng hoảng kinh tế toàn cầu đến nền kinh tế của Việt Nam là khó tránh khỏi, nhất là tình hình xung đột Nga – Ucraina có khuynh hướng kéo dài như hiện nay. Nghiên cứu này được kiểm chứng thực tế cho 25 ngày đầu tiên của dự báo phù hợp với xu hướng thực tế trên thị trường chứng khoán hiện nay.

4. KẾT LUẬN

Nghiên cứu này đã xem xét khả năng ứng dụng và đánh giá hiệu suất của các mô hình đơn LSTM, GRU và các mô hình phức hợp của nó, trong việc dự báo xu hướng biến động giá cổ phiếu trên sàn giao dịch chứng khoán Hồ Chí Minh. Nhằm mục đích khám phá khả năng của các mô hình này trong việc ứng dụng vào lĩnh vực tài chính và chứng khoán. Nghiên cứu đã tiến hành một loạt thực nghiệm trên tập dữ liệu lịch sử của chỉ số đo lường xu hướng biến động giá cổ phiếu. Kết quả cho thấy, mô hình LSTM-GRU Hybrid đã cho ra các dự báo có độ chính xác cao và phản ánh chính xác hướng biến động của thị trường chứng khoán. Điều này góp phần khẳng định khả năng ứng dụng của mô hình phức hợp trong việc dự báo biến động giá cổ phiếu trên thị trường chứng khoán ở Việt Nam hiện nay. Đặc biệt, khả năng học và bắt chước mô hình dự báo giúp mô hình LSTM-GRU Hybrid vượt trội trong việc ứng dụng vào dự báo xu hướng biến động giá cổ phiếu.

Tuy nhiên, nghiên cứu cũng nhận thấy rằng việc chuẩn bị dữ liệu, lựa chọn siêu tham số và kiểm tra hiệu suất mô hình vẫn còn là những thách thức, cần có sự cân nhắc và kiểm tra kỹ lưỡng để đảm bảo tính chính xác và tin cậy của kết quả dự báo. Bên cạnh đó, một số vấn đề cần phải nghiên cứu thêm như: máy học trong việc nghiên cứu tâm lý đám đông, các chính sách của Chính phủ, các vấn đề khác trên thế giới ảnh hưởng đến thị trường chứng khoán của Việt Nam hiện nay. Song song, mô hình GRU cho kết quả tương đối tốt, nhưng gặp phải hiện tượng quá khớp và nghiên cứu cũng chưa đề xuất các giải pháp khắc phục phù hợp nhất.

Kết quả dự báo cho thấy các chính sách Chính phủ đã có tác động đến thị trường chứng khoán nói riêng và tình hình kinh tế chính trị trong nước ở thời điểm hiện tại. Một số chính sách phổ biến hiện nay có thể xem xét như: Một là, chính sách về tiền tệ, Chính phủ đã điều chỉnh lãi suất cơ bản, tăng giảm cung tiền và thực hiện các biện pháp khác đã ảnh hưởng gián tiếp đến sự biến động của thị trường chứng khoán. Hai là, chính sách fiskal, Chính phủ đã điều chỉnh các chính sách liên quan đến thuế, chi phí và ngân sách đã tác động đến hoạt động kinh doanh và tình hình kinh tế tổng thể, từ đó ảnh hưởng đến thị trường chứng khoán. Ba là, chính sách quản lý tài sản và đầu tư, Chính phủ đã thông qua các chính sách quản lý đầu tư công và tài sản quốc gia đã tạo điều kiện thuận lợi cho hoạt động kinh doanh và đầu tư, ảnh hưởng đến hiệu suất thị trường chứng khoán. Sau cùng, chính sách hỗ trợ và khuyến mãi, Chính phủ đã áp dụng các biện pháp hỗ trợ và khuyến mãi đặc biệt cho các ngành nông nghiệp, doanh nghiệp trong vấn đề chuyển đổi số nhằm thúc đẩy sự phát triển kinh tế nói chung. Các chính sách này có thể tác động đến tâm lý và quyết định của các nhà đầu tư, ảnh hưởng đến sự biến động và xu hướng của thị trường chứng khoán. Tuy nhiên, tác động của Chính phủ đến thị trường chứng khoán thường phức tạp và còn phụ thuộc vào nhiều yếu tố khác nhau như tình hình kinh tế tổng thể, tâm lý thị trường và sự biến đổi trong tài chính toàn cầu.

Trong tương lai, nghiên cứu này kỳ vọng mô hình LSTM-GRU Hybrid sẽ được áp dụng rộng rãi trong các nghiên cứu và ứng dụng thực tiễn cho lĩnh vực tài chính và chứng khoán. Các nghiên cứu tiếp theo có thể tập trung vào việc tối ưu hóa mô hình, thay thế hoặc nghiên cứu bổ sung các hàm kích hoạt khác nhằm nâng cao khả năng dự báo, giảm thiểu hiện tượng bão hòa của hàm sigmoid, cũng như giải quyết triệt để vấn đề "quá khớp" trong tập huấn luyện và thử nghiệm trên các dữ liệu thực tế khác nhau. Vấn đề số bước thời gian trong các mô hình máy học xử lý chuỗi thời gian nó sẽ xác định thời gian và phạm vi dự báo. Do đó, nghiên cứu cần phải xác định chính xác để đưa vào mô hình, khi đó mô hình mới có thể hiểu được cấu trúc của dữ liệu thời gian và học cách dự đoán trong tương lai dựa trên các chu kỳ và mẫu dữ liệu đã thấy. Việc xác định bước thời gian có liên quan đến việc xác định kích thước ô nhớ, điều này phụ thuộc rất lớn vào nguồn tài nguyên máy tính có thể thực hiện, dẫn đến việc quyết định loại hình dự báo ngắn hạn, trung hạn hay dài hạn.

Nhìn chung, nghiên cứu này đã mang lại những kiến thức quý báu về ứng dụng và đánh giá hiệu suất các mô hình máy học trong dự báo biến động giá cổ phiếu. Các kết quả và nhận định từ nghiên cứu này cung cấp cơ sở cho việc áp dụng mô hình trong thực tế, đóng góp vào sự phát triển của lĩnh vực tài chính và chứng khoán

TÀI LIỆU THAM KHẢO

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, L., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. Q. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. http://doi.org/10.48550/arXiv.1603.04467

Alpaydin, E. (2014). *Introduction to Machine Learning*. MIT Press.

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, 80, 340-355. http://doi.org/ 10.1016/j.eswa.2017.02.044

Cho, K., Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. http://doi.org/10.3115/v1/D14-1179

Chollet, F. (2015). Keras. https://github.com/fchollet/keras

Chung, H., & Shin, K. S. (2020). Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction. *Neural*

Computing and Applications, 32. http://doi.org/10.1007/s00521-019-04236-3

Daneshvar, A., Ebrahimi, M., Salahi, F., & Rahmaty, M. (2022). Brent crude oil price forecast utilizing deep neural network architectures. *Computational Intelligence and Neuroscience*. http://doi.org/1-13. 10.1155/2022/6140796

de Freitas, N., Shahriari, B., Swersky, K., Wang, Z., & Adams, R. P. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1), 148–175. https://doi.org/10.1109/jproc.2015.2494218

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12, 2121-2159.

Guo, C., Liu, G., & Chen, C. H. (2020). Air pollution concentration forecast method based on the deep ensemble neural network. *Wireless Communications and Mobile Computing*. https://doi.org/10.1155/2020/8854649

Hochreiter, S., & Schmidhuber, J. (1997), Long Short-Term Memory. *Neural Computation*, 9, 1735–1780. http://doi.org/10.1162/neco.1997.9.8.1735

Hong, W. C. (2021). Application of Seasonal SVR with Chaotic Immune Algorithm in Traffic Flow Forecasting. *Neural Computing and Applications*, 21, 583–593. http://doi.org/10.1007/s00521-010-0456-7

Hossain, M., Karim, R., Thulasiram, R., Bruce, N. D. B., & Wang, Y. (2018). Hybrid Deep Learning Model for Stock Price Prediction, 1837-1844. http://doi.org/10.1109/SSCI.2018.8628641

Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., & Zhang, H. (2019), Deep Learning with Long Short-Term Memory for Time Series Prediction. *IEEE Communications Magazine*, 1-6. https://doi.org/10.1109/MCOM.2019.1800155

Hunter, J. D. (2007). Matplotlib: a 2D graphics environment. *Computing in Science & Engineering*, 9, 90–95. https://doi.org/10.1109/MCSE.2007.55

Kanzari, D., Nakhli, M. S., Gaies, B., & Sahut, J. M. (2023). Predicting Macro-Financial Instability - How Relevant is Sentiment? Evidence from Long Short-Term Memory Networks, 65. http://doi.org/10.1016/j.ribaf.2023.101912

Kingma, D. P., & Ba, J. (2014). ADAM: A Method for Stochastic Optimization. *International Conference on Learning Representations*. http://doi.org/10.48550/arXiv.1412.6980

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521, 436-44. http://doi.org/ 10.1038/nature14539

Lin, X., Yang, Z., & Song, Y. (2011). Intelligent stock trading system based on improved technical analysis and Echo State Network. *Expert systems with Applications*, *38*(9), 11347-11354. http://doi.org/ 10.1016/j.eswa.2011.03.001

Liu, Y., Wang, Z., & Zheng, B. (2019). Application of Regularized GRU-LSTM Model in Stock Price Prediction. 1886-1890. http://doi.org/10.1109/ICCC47050.2019.9064035

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, *53*(4), 3007-3057. http://doi.org/10.1007/s10462-019-09754-z

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, V. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*. https://doi.org/10.48550/arXiv.1201.0490

Salimath, S., Chatterjee, T., Mathai, T., Kamble, P., & Kolhekar, M. (2021). Prediction of Stock Price for Indian Stock Market: A Comparative Study Using LSTM and GRU. http://doi.org/ 10.1007/978-3-030-88244-0_28

Shejul, A. A., Chaudhari, A., Dixit, B. A., & Lavanya, B. M. (2023). Stock Price Prediction Using GRU, SimpleRNN and LSTM. *Lecture Notes in Electrical Engineering*, 959, 529–535.

Song, Y., Lee, J. W., & Lee, J. (2019). A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction. *Applied Intelligence*, 49, 897-911. http://doi.org/10.1007/s10489-018-1308-x

Su, Z., Xie, H., & Han, L. (2021). Multi-factor RFG-LSTM algorithm for stock sequence predicting. *Computational Economics*, 57(4), 1041-1058. http://doi.org/ 10.1007/s10614-020-10008-2

Szandała, T. (2021). Review and comparison of commonly used activation functions for deep neural networks. http://doi.org/10.1007/978-981-15-5495-7_11

Waskom, M. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software*, 6, 3021. http://doi.org/10.21105/joss.03021

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. *Neural Computing and Applications*, 32, 1609-1628. http://doi.org/10.1007/s00521-019-04212-x

Yun, K. K., Yoon, S. W., & Won, D. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Systems with Applications*, 186, 115716. http://doi.org/10.1016/j.eswa.2021.115716