KẾT HỢP MÔ HÌNH HỌC MÁY VÀ MÔ HÌNH THỐNG KÊ TRONG DỰ BÁO CHUỖI THỜI GIAN: TRƯỜNG HỢP LẠM PHÁT TẠI VIỆT NAM GIAI ĐOẠN 2000 - 2021

COMBINING MACHINE LEARNING AND STATISTICAL MODELS IN TIME SERIES FORECASTING: CASE OF INFLATION IN VIETNAM PERIOD 2000 - 2021

Ngày nhận bài: 28/02/2022

Ngày chấp nhận đăng: 25/03/2022

Nguyễn Hương Ly✉, Hoàng Thị Thu Hà

TÓM TẮT

Dự báo chuỗi thời gian là bài toán hết sức quan trọng trong hoạt động sản xuất, kinh doanh và hoạch định chính sách. Ở Việt Nam, nhiều nghiên cứu đã sử dụng các mô hình thống kê và mô hình học sâu một cách độc lập để dự báo các chuỗi thời gian như: lượng vốn đầu tư nước ngoài, chỉ số chứng khoán, chỉ số giá tiêu dùng, … Tuy nhiên việc kết hợp các mô hình trên trong dự báo các biến số kinh tế đang còn khá ít ở Việt Nam. Bài viết nhằm xác định kết hợp tối ưu giữa các mô hình học sâu và mô hình học máy truyền thống khi dự báo chỉ số lạm phát của Việt Nam trong giai đoạn 2000 - 2021.

Từ khóa: Dự báo chuỗi thời gian, mô hình học sâu, mô hình học máy truyền thống, mô hình kết hợp, lạm phát.

ABSTRACT

Time series forecasting is a very important problem in production, business and policy making. In Vietnam, many studies have used statistical models and deep learning models independently to forecast time series such as: amount of foreign investment, stock index, consumer price index, etc. However, the combination of the above models in forecasting economic variables is still quite rare in Vietnam. The article aims to determine the optimal combination between deep learning models and traditional machine learning models in forecasting the inflation index of Vietnam in the period 2000 - 2021.

Keywords: Time series forecasting, deep learning models, traditional machine learning models, association models, inflation..

1. Giới thiệu

Dự báo chuỗi thời gian là quá trình phân tích chuỗi thời gian bằng việc sử dụng các phương pháp thống kê và mô hình hóa để dự báo. Một số mô hình học máy truyền thống thường được sử dụng để dự báo chuỗi thời gian có thể cho kết quả tốt. Tuy nhiên, chúng có những nhược điểm như là hiệu suất bị tác động do thiếu các yếu tố đặc trưng, không thể nhận ra những biến động phức tạp trong dữ liệu và dự báo trong dài hạn sẽ không chính xác.

Một trong số đó là mô hình tự hồi quy tích hợp trung bình trượt (ARIMA). Ngay từ ngày đầu, mô hình này đã được sử dụng phổ biến trong nhiều lĩnh vực như thống kê, ước lượng và dự báo (Thomas, 1983) vì người ta nhận thấy mô hình có nhiều ưu điểm (Box, 1970; Jarrett, 1991). Tính chính xác cao khi dự báo đồng thời loại bỏ được hiện tượng đa cộng tuyến trong mô hình là ưu điểm lớn nhất của ARIMA. Tuy nhiên, mô hình này cũng có một số nhược điểm, đó là: khó khăn trong việc thiết lập mô hình chuẩn từ nhóm các mô hình tiềm năng; phương pháp dự báo mang tính chủ quan; mô hình lí thuyết và các mối quan hệ cấu trúc không khác so với một

Nguyễn Hương Ly, Trường Đại học Kinh tế Quốc dân
Hoàng Thị Thu Hà, Trường Đại học Thương mại

số mô hình dự báo đơn giản (Thomas, 1983); dữ liệu phải tuân theo các giả thiết của mô hình hồi quy tuyến tính. Khắc phục những hạn chế của các mô hình học máy truyền thống, các mô hình học sâu (*deep learning*, DL) là một bước tiến đáng kể. Nhiều nghiên cứu cho thấy mô hình học sâu được áp dụng rộng rãi trong dự báo và đạt kết quả có độ chính xác cao.

DL là một phần của học máy *(machine learning*, ML). Trong những năm gần đây, các kĩ thuật DL đã vượt trội so với các mô hình truyền thống. Đặc biệt, DL được ứng dụng thành công khi giải quyết các bài toán dự báo chuỗi thời gian bởi vì DL sử dụng cấu trúc nhiều lớp của các thuật toán được gọi là mạng nơ-ron giúp việc chuẩn bị dữ liệu nhanh hơn và có thể học các mẫu dữ liệu phức tạp hơn. Trong DL, các mạng nơ-ron hình thành nên hầu hết các mô hình đã được huấn luyện trước. Có ba kiểu mạng nơ-ron: mạng nơ-ron nhân tạo (ANN), mạng nơ-ron tích tụ (CNN) và mạng nơ-ron hồi quy (RNN).

Trong những năm gần đây, RNN được áp dụng trong nhiều lĩnh vực: nhận dạng giọng nói, mô hình hóa ngôn ngữ, … Tuy nhiên, việc truyền tải đầy đủ dữ liệu trong khoảng thời gian dài là một công việc rất khó khăn. Để khắc phục các vấn đề tiềm ẩn do sự biến mất của độ dốc trong RNN, các nhà khoa học Hochreiter, Schmidhuber và Bengio đã cải tiến RNN thành mạng bộ nhớ dài-ngắn hạn (Long Short-Term Memory, LSTM).

LSTM là phiên bản cải tiến của RNN, giúp cho việc ghi nhớ các dữ liệu trong quá khứ một cách dễ dàng hơn. Vấn đề độ dốc (*vanishing gradient*) bị triệt tiêu trong RNN đã phần nào được giải quyết ở đây. Mạng LSTM xây dựng mô hình bằng phương pháp lan truyền ngược. LSTM rất thích hợp cho việc phân loại, xử lí và dự đoán chuỗi thời gian có độ trễ không xác định. Sự khác biệt lớn nhất giữa RNN và LSTM là LSTM có thể lưu trữ thông tin trong khoảng thời gian dài, đây chính là điểm nổi trội của LSTM so với RNN cũng như các mạng nơ-ron khác. Tuy nhiên, vì thuật toán của LSTM phức tạp hơn RNN nên việc xử lí thông tin cần nhiều thời gian hơn.

Vì mỗi mô hình dự báo có đều có ưu và nhược điểm nên câu hỏi đặt ra là làm thế nào để kết quả dự báo chuỗi thời gian có độ chính xác cao nhất? Hiện nay, nhiều nghiên cứu nước ngoài đã sử dụng phương pháp dự báo chuỗi thời gian bằng cách kết hợp các mô hình học máy truyền thống và học sâu. Chẳng hạn, Liu và cộng sự (2017) đã sử dụng RNN để dự báo sự biến động của cổ phiếu. Kết quả là RNN tốt hơn MLP và máy véc tơ hỗ trợ (SVM). Bên cạnh đó, Gao, Chai và Liu (2017) đã sử dụng bộ dữ liệu về lịch sử giao dịch chỉ số S&P 500 trong 20 ngày và dự báo thị trường chứng khoán bằng bốn phương pháp khác nhau: trung bình trượt (MA), trung bình trượt hàm mũ (EMA), SVM) và LSTM. Kết quả cho thấy LSTM có độ chính xác cao nhất. Khi dự báo doanh số bán của các công ty bất động sản, Soy Temür, Akgün, và Temür (2019) dùng các mô hình ARIMA, LSTM và ARIMA-LSTM cùng với bộ số liệu gồm 124 tháng, từ tháng 01 năm 2008 đến tháng 04 năm 2018 về tổng doanh thu bán nhà ở Thổ Nhĩ Kỳ. Kết quả là mô hình kết hợp ARIMA-LSTM có phần trăm sai số tuyệt đối trung bình (MAPE) và sai số bình phương trung bình (MSE) là thấp nhất. Hyeong Kyu Choi (2018) cũng sử dụng sự kết hợp này trong việc dự báo hệ số tương quan về giá của hai cổ phiếu riêng biệt. Kết quả là mô hình kết hợp ARIMA-LSTM có khả năng dự báo vượt trội so với các mô hình truyền thống.

Tuy nhiên, ở Việt Nam những nghiên cứu tương tự còn khá mới mẻ. Phạm Nguyễn Hoàng Phúc và Trương Tấn Phát (2020) đã sử dụng LSTM dự báo biến động của thị trường giao dịch tài chính. Dữ liệu bắt đầu từ

ngày đầu tiên mà các công ty được cập nhật tên trên Yahoo Finance và kết thúc vào ngày 24/03/2020. Kết quả dự báo có xu hướng tương đồng với dữ liệu thực tế trong dài hạn có tính chính xác cao trong ngắn hạn. Do đó, nhóm tác giả lựa chọn đề tài "Kết hợp mô hình học máy và mô hình thống kê trong dự báo chuỗi thời gian: Trường hợp lạm phát tại Việt Nam giai đoạn 2000 - 2021". Bài viết nhằm mục đích lựa chọn kết hợp tối ưu trong ba kết hợp: ARIMA-RNN, ARIMA-LSTM và ARIMA-RNN-LSTM. Số liệu được sử dụng để minh họa cho nghiên cứu là chỉ số lạm phát tại Việt Nam giai đoạn từ tháng 01 năm 2000 đến tháng 07 năm 2021, được thu thập bởi Tổng cục Thống kê Việt Nam (GSO). Bài viết sử dụng phần mềm Python trong tính toán và phân tích các kết quả.

2. Cơ sở lý thuyết và phương pháp nghiên cứu

2.1. Cơ sở lý thuyết

2.1.1. Một số mô hình dự báo chuỗi thời gian

a) **Mô hình ARIMA**

Xét chuỗi dừng Y_t, mô hình tự hồi quy tích hợp trung bình trượt ARIMA được đề xuất bởi hai nhà khoa học George Box và Gwilym Jenkins. Mô hình gồm ba thành phần chính: thành phần tự hồi quy (AR), tính dừng của chuỗi thời gian (I) và thành phần trung bình trượt (MA). Mô hình có dạng sau:

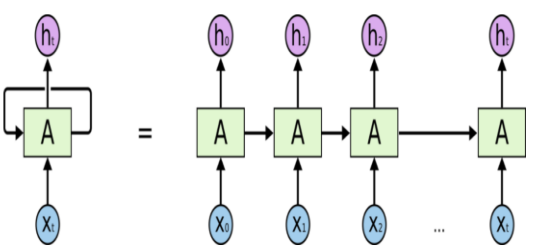
$$Y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i Y_{t-i} + \sum_{i=1}^{q} \beta_i U_{t-i}$$

trong đó, u_t là nhiễu trắng, p là bậc tự hồi quy, d là bậc của sai phân và q là bậc của trung bình trượt.

Để đánh giá sự phù hợp của mô hình ARIMA, người ta thường sử dụng phương pháp Box-Jenkins. Bên cạnh đó, chuỗi thời gian trong mô hình ARIMA phải thỏa mãn là chuỗi dừng, trong khi một chuỗi thời gian thường có ba thành phần chính là xu thế, chu kì và mùa. Để đưa yếu tố mùa vào trong mô hình, người ta đã phát triển thành mô hình ARIMA có yếu tố mùa (SARIMA). Tuy nhiên, nếu chuỗi thời gian vẫn chứa hai thành phần còn lại (xu thế và chu kì) thì khi dự báo trong dài hạn, mô hình SARIMA cho kết quả không còn chính xác. Để cải thiện chất lượng dự báo trong dài hạn, người ta tìm một hàm trơn thông qua mạng nơ-ron. Qua đó, chuỗi thời gian gốc được chuyển thành hàm trơn này bằng việc loại bỏ yếu tố mùa ra khỏi chuỗi thời gian (Fathi, 2019).

b) **Mô hình RNN**

Mạng nơ-ron truyền thống gồm ba thành phần chính là lớp đầu vào, lớp ẩn và lớp đầu ra, trong đó đầu vào và đầu ra là độc lập. Tuy nhiên, đối với chuỗi thời gian, đầu vào và đầu ra có quan hệ mật thiết. Do đó, RNN xuất hiện với ý tưởng chính là lưu lại thông tin từ những bước trước đó để đưa ra dự báo chính xác nhất ở bước hiện tại.

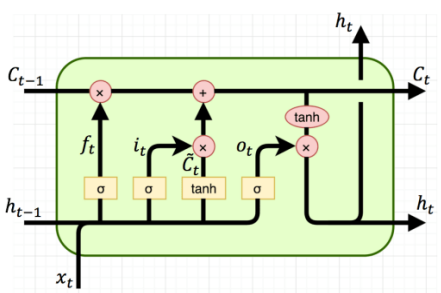


Hình 1: Mô hình RNN [7]

Theo Hình 1, vế trái là một mạng nơ-ron hồi quy A với đầu vào là x_t và đầu ra là h_t. Vế phải là chuỗi các vòng lặp, trong đó thông tin được truyền từ bước này qua bước khác, qua các bản sao của mạng nơ-ron. Tuy nhiên, RNN bị hạn chế bởi nó chỉ nhớ những dữ liệu gần, trong ngắn hạn. Vì giá trị của độ dốc lặp khi thực hiện lan truyền ngược (back propagation) giảm dần qua mỗi vòng lặp, do đó độ dốc (*gradient*) sẽ mất đi sau khoảng thời gian. Vì vậy ma trận trọng số không được cập nhật giá trị và mạng nơ-ron ngừng học trong vòng lặp này. Để khắc phục hạn chế này của RNN, các nhà khoa học đã phát triển mô hình bộ nhớ dài-ngắn hạn LSTM.

c) **Mô hình LSTM**

Mô hình LSTM được Hochreiter và Schmidhuber đề xuất vào năm 1997, là một trong những phát triển mới nhất của mô hình RNN nhằm khắc phục vấn đề độ dốc biến mất và vấn đề phụ thuộc xa của mô hình RNN. Mô hình RNN là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Trong các mô hình RNN tiêu chuẩn, mô-đun lặp có cấu trúc rất đơn giản, thường là một tầng *tanh*. Các mô hình LSTM cũng có cấu trúc. Cấu trúc trình tự tương tự như RNN, nhưng các mô-đun lặp lại có cấu trúc hơi khác. LSTM gồm có 4 thành phần (cell, forget gate, input gate, output gate) tương tác với nhau một cách rất đặc biệt.



Hình 2: Mô tả mô hình LSTM [7]

Cổng *forget* có nhiệm vụ chọn thông tin bị bỏ quên từ trạng thái trước. Thông tin được lấy từ đầu vào x_t và trạng thái ẩn (hidden state) h_{t-1} qua hàm kích hoạt σ đưa về giá trị $(0,1)$

$$f_t = \sigma(U_f * x_t + W_f * h_{t-1} + b_f)$$

Thông tin mới sau đó được lưu trữ trong đầu vào trạng thái hiện tại. Cổng đầu vào truyền thông tin ở trạng thái hiện tại và trạng thái ẩn của cổng trước qua 2 hàm: hàm kích hoạt σ trả về giá trị $(0,1)$ để quyết định giá trị nào cần được cập nhật và hàm kích hoạt *tanh* tạo một véc tơ mới C_t có giá trị $(-1,1)$ để thêm vào trạng thái.

$$i_t = \sigma(U_i * x_t + W_i * h_{t-1} + b_i)$$

$$\widetilde{C_t} = tanh(U_c * x_t + W_c * h_t + b_c)$$

Sau đó, ô mới (*cell state*) C_t cập nhật trạng thái từ các thông tin bị bỏ qua ở ô trước đó $f_t * C_{t-1}$ và khi đó thông tin mới sẽ được cập nhật ở đầu vào $i_t * \widetilde{C_t}$

$$C_t = \sigma(f_t * C_{t-1} + i_t * \widetilde{C_t})$$

Cổng *output* xác định thông tin đầu ra của trạng thái hiện tại bằng cách sử dụng giá trị trả về của hàm kích hoạt σ để quyết định xuất bao nhiêu thông tin trạng thái. Trạng thái mới C_t được chuyển cho hàm *tanh* để trả về giá trị $(-1,1)$. Kết hợp 2 giá trị được xuất thành đầu ra cho trạng thái ẩn (*hidden state*).

$$o_t = \sigma(U_0 * x_t + W_0 * h_{t-1} + b_0)$$

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

Từ cách thức hoạt động trên, LSTM được đánh giá là vượt trội hơn so với RNN. LSTM có thể truy vấn được thông tin từ một tập thông tin lớn hơn. Vì vậy, LSTM rất thích hợp để dự báo chuỗi thời gian trong dài hạn.

d) **Dự báo kết hợp**

Mỗi phương pháp dự báo đều có những ưu điểm và hạn chế riêng, một phương pháp có thể dự báo tốt trong giai đoạn này nhưng lại chưa phải là là tốt trong giai đoạn khác. Thực tế, khó có thể có một phương pháp tốt hơn hẳn các phương pháp khác cho mọi tình huống (Armstrong, 2001). Do đó ý tưởng về phương pháp dự báo kết hợp đã được giới thiệu bởi Newbold and Granger (1974) và Yang (2004). Ý tưởng của phương pháp này là lấy trung bình không có trọng số các kết quả dự báo từ các phương pháp khác nhau. Armstrong (2001) đã so sánh 30 mẫu thử

nghiệm, mức giảm sai số trước đó đối với các dự báo kết hợp có trọng số bằng nhau trung bình khoảng 12,5% và dao động từ 3% - 24%. Đôi khi, dự báo kết hợp chính xác hơn các phương pháp dự báo đơn. Hugo và cộng sự (2008) sử dụng ba lớp mô hình: BVAR, FAVAR, DSGE và dự báo kết hợp để dự báo lạm phát cho Australia. Kết quả cho thấy rằng phương pháp dự báo kết hợp phù hợp nhất là lấy trung bình của các giá trị dự báo từ các mô hình này.

2.1.2. Một số chỉ số đánh giá chất lượng dự báo

Có rất nhiều chỉ số dùng để đo lường hoặc kiểm định để đánh giá chất lượng của một mô hình bất kì. Một số chỉ số thường được dùng trong đánh giá chất lượng mô hình dự báo là:

***a)* Sai số phần trăm tuyệt đối trung bình**

MAPE (Mean absolute percentage error) là phần trăm sai số trung bình tuyệt đối của các dự báo. Kết quả dự báo càng tốt khi MAPE càng nhỏ. MAPE có thể được tính như sau:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|Y_t - \bar{Y}_t|}{|Y_t|}$$

Y_t là giá trị thực tế tại thời điểm t, $\bar{Y}_t$ là dự báo giá trị tại thời điểm t và n là số lần quan sát.

***b)* Sai số trung bình tuyệt đối**

MAE (Mean absolute error) là sai số trung bình tuyệt đối. MAE đo lường mức độ trung bình của các sai số trong một tập hợp các dự báo, mà không xem xét hướng của chúng.

$$MAE = \frac{\sum_{t=1}^{n}|\bar{Y}_t - Y_t|}{n} = \frac{\sum_{t=1}^{n}|e_t|}{n}$$

với e_t là sai số của dự báo ở quan sát t.

***c)* Sai số bình phương trung bình**

Trong thống kê, MSE (Mean squared error) là trung bình bình phương của các sai số. MSE đo mức chênh lệch bình phương trung bình giữa các giá trị thực và các giá trị dự báo.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(Y_t - \bar{Y}_t)^2$$

***d)* Tiêu chuẩn thông tin Akaike**

Một trong những tiêu chí thường được dùng để lựa chọn mô hình đó là chỉ số AIC (*Akaike Information Criteria*). AIC được hình thành dựa trên lý thuyết thông tin (information theory)

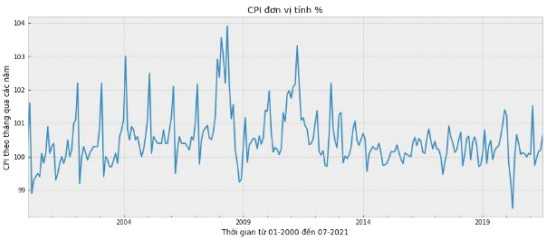
AIC ước tính lượng thông tin tương đối bị mất bởi một mô hình nhất định: một mô hình càng mất ít thông tin thì chất lượng của mô hình đó càng cao. Khi ước tính lượng thông tin bị mất bởi một mô hình, AIC đề cập đến sự cân bằng giữa tính phù hợp của mô hình và tính đơn giản của mô hình. Nói cách khác, AIC đối phó với cả rủi ro trang bị quá mức và rủi ro trang bị thiếu. Với k là số lượng tham số ước tính, $\hat{L}$ là giá trị tối đa của hàm hợp lý (maximum likelihood function) của mô hình. Khi đó, giá trị AIC được tính như sau:

$$AIC = 2k - 2\ln\hat{L}$$

2.2. Phương pháp nghiên cứu

2.2.1. Số liệu

Để đo lường tỉ lệ lạm phát, bài viết đã sử dụng bộ dữ liệu "Chỉ số giá tiêu dùng của Việt Nam" là chuỗi thời gian loạt gồm 259 mẫu được lấy hàng tháng từ tháng 1 năm 2000 đến tháng 7 năm 2021. Tập dữ liệu có thể được tìm thấy trong tài liệu chính thức trang web của chính phủ Tổng cục Thống kê. Số liệu CPI theo tháng được xây dựng dưới dạng file CSV, số liệu xây dựng trên đơn vị tính là %. Số liệu được mô tả theo biểu đồ dưới đây:



Hình 3: Biểu đồ chỉ số lạm phát của Việt Nam giai đoạn tháng 1/ 2000 - 7/ 2021

Nguồn: Tác giả

CPI Việt Nam giai đoạn tháng 1 năm 2000 đến tháng 7 năm 2021 có sự biến động. Tháng 5 năm 2008 có chỉ số giá tiêu dùng ở mức đỉnh cao nhất là 103,91% trong khi tháng có chỉ số thấp nhất là 98,457% (tháng 3 năm 2020)

2.2.2. Phương pháp LSTM

Để điều chỉnh mô hình LSTM một cách sinh động, việc điều chỉnh siêu tham số mở rộng đã được thực hiện sau khi thực hiện một mô hình chung. Tìm kiếm siêu tham số bao gồm kích thước lô, số kỷ nguyên tối đa, tỷ lệ học tập, kích thước lớp ẩn, số lượng lớp LSTM xếp chồng lên nhau và tình trạng bỏ học. Các thông số này đã được kiểm tra bằng cách chia tỷ lệ có chọn lọc một thông số duy nhất trong khi sửa chữa những thông số khác. Các thông số liên quan (chẳng hạn như tốc độ học và số kỷ nguyên tối đa) đã được xem xét cùng nhau. So với phương pháp đơn giản, LSTM hoạt động rất tốt. Tỷ lệ bỏ cuộc dường như có tác động đáng kể đến hiệu suất của mô hình với tỷ lệ bỏ cuộc thấp hơn dẫn đến hiệu suất tốt hơn. Mặc dù với thông số cụ thể thay đổi một số mô hình có nhiều lớp hơn đạt được hiệu suất tương đương, nhưng tổng thể các mô hình hoạt động tốt hơn với ít hơn 4 lớp và một số kết quả tốt hơn ở các mô hình có 1 lớp. Thay vì tìm kiếm dạng lưới cho các tham số, nhóm tác giả đã thử các kiến trúc khác nhau làm thay đổi số lớp, cũng như số lượng đơn vị trong các lớp cho đến khi có được MAPE tốt nhất có thể. Mô hình được chọn là mô hình tuần tự của các lớp, 3 lớp LSTM, 3 lớp bỏ học và một lớp dày đặc.

Dữ liệu chuỗi thời gian khi chia dữ liệu đào tạo (*train*) và dữ liệu kiểm tra (*test)* điều chú ý là phải giữ lại thứ tự các quan sát, vì thế chọn 80% các quan sát phần đầu cho dữ liệu *train* và 20% còn lại cho dữ liệu *test* thay vì chọn ngẫu nhiên. Bộ dữ liệu gồm 259 quan sát được thành 2 phần. Phần 1 gồm 207 quan sát đầu tiên được dùng để xây dựng mô hình LSTM. Phần 2 gồm 52 quan sát còn lại để kiểm chứng độ chính xác của mô hình LSTM.

Để thuận lợi trong việc tính toán, nhóm tác giả đã chuẩn hóa dữ liệu về khoảng [0;1] bằng phương pháp MinMaxScaler:

$$x_{norm}^{(i)} = \frac{x^{(i)} - X_{min}}{X_{max} - X_{min}}$$

Phương pháp này được thực hiện trong thư viện Sklearn của Python.

Coi mỗi giá trị của bộ dữ liệu là một bước thời gian và sử dụng 60 bước thời gian để xây dựng các lớp của mô hình LSTM. Bước thời gian được sử dụng trong bài viết là 60, tức là sử dụng dữ liệu trong quá khứ của 60 tháng để dự đoán kết quả của tháng thứ 61. Sau đó ma trận được chuyển thành dạng: giá trị, bước thời gian, 1 chiều đầu ra. Từ đó, xây dựng mô hình LSTM với 50 nơ-ron và 4 lớp ẩn bằng cách sử dụng hàm MSE và thuật toán lặp tối ưu Adam lặp lại 500 lần. Sau đó thêm 1 *Dense Layer* vào cuối mô hình với số nơ-ron là 1. Khởi tạo lớp *Sequential*, đây sẽ là lớp mô hình và sẽ thêm các lớp LSTM, *Dropout* và *Dense* vào mô hình này. Thêm 3 lớp LSTM liên tiếp và cứ tiếp tục như vậy qua 1 lớp là có 1 *dropout* 0.3. Cuối cùng là một lớp *Dense* với 1 chiều đầu ra. Sử dụng hàm mất mát (*loss*) để đánh giá chất lượng mô hình và sử dụng trình Adam để tối ưu hóa thuật toán.

2.2.3. Phương pháp dự báo theo mô hình RNN

Sử dụng thư viện *tensorflow* để hỗ trợ quá trình *training*. Đồng thời kết hợp với *pandas* và *numpy* để phân tích, và xử lý cấu trúc dữ liệu, và *matplotlib* dùng để vẽ đồ thị.

Đầu tiên khai báo các thư viện, sau đó tải dữ liệu đào tạo. x_train được sử dụng ở đây từ 01/2000 đến 07/2021, khi đó, có 259 giá trị test để so sánh.

Ở đây khi sử dụng mạng nơ-ron mô hình RNN truyền thống, khai báo số liệu với hàm *activation* là *relu*. *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 đầu vào cho tầng tiếp theo. Hàm *relu* đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng nơ-ron. *Relu* đơn giản lọc các giá trị < 0. So với *sigmoid* và *tanh*, *relu* có tốc độ tính toán nhanh và tốc độ hội tụ nhanh hơn hẳn.

2.2.4. Phương pháp dự báo theo mô hình ARIMA

Trước tiên, sử dụng kiểm định Dickey Fuller để kiểm định tính dừng của chuỗi thời gian. Vì $P_{\text{-value}}$=0.011734 nên có thể nói chuỗi không dừng tại mức ý nghĩa 5%.

Sử dụng "tìm kiếm lưới" kết hợp thông số khác nhau. Mỗi tổ hợp các thông số phù hợp với một mô hình SARIMA và đánh giá chất lượng tổng thể của nó. Mô hình ARIMA tốt nhất là ARIMA (1,1,1)x(1,0, 1, 12) với AICmin =357,91.

Khi áp dụng các mô hình SARIMA điều quan trọng là đảm bảo rằng không vi phạm bất kì giả định nào. *Plot_diagnostics* cho phép tạo ra mô hình dự báo nhanh và kiểm định giả thuyết. Hàm *get_prediction()* và *conf_int()* có chức năng lấy các giá trị và khoảng tin cậy cho dự báo.

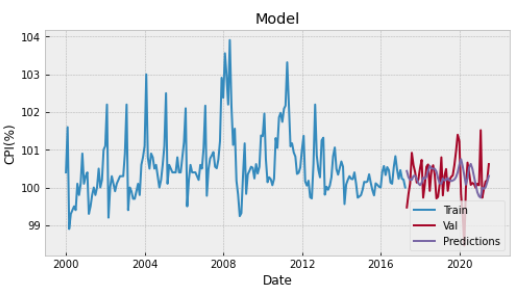
2.2.5. Phương pháp dự báo kết hợp

Để cải thiện độ chính xác của dự báo, tác giả dự báo kết hợp bằng việc lấy trung bình các giá trị dự báo thu được từ các mô hình dự báo LSTM, RNN và ARIMA.

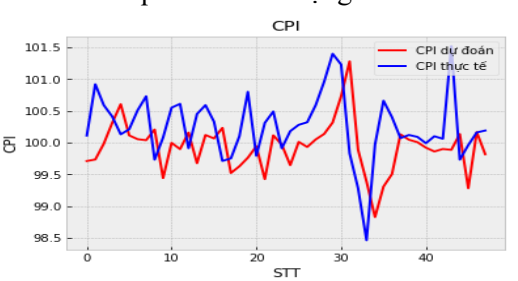
3. Kết quả và đánh giá

3.1. Kết quả

***a)* Kết quả dự báo theo mô hình LSTM**



Hình 4: Kết quả mô hình mạng LSTMa



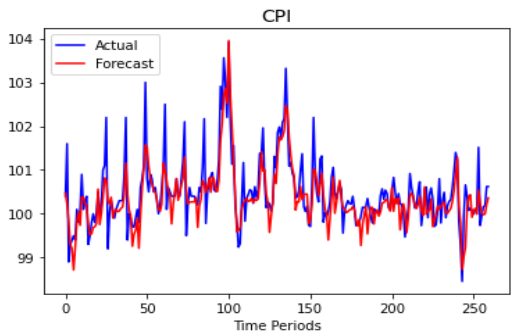
Hình 5: Kết quả mô hình mạng LSTMb
Nguồn: Tác giả

Hai đồ thị trên minh họa cho tỉ lệ thực tế và tỉ lệ dự báo trong hai mô hình dự báo lạm phát LSTM. Trong hình 4, đường màu đỏ là chỉ số thực tế và đường màu tím là dự báo. Có thể xem, kết quả dự báo gần đúng xu hướng tăng giảm của lạm phát thực tế.

Tuy nhiên kết quả dự báo chưa sát với kết quả thực tế bởi vì giá trị MAPE=0.4074 - khác biệt trung bình giữa giá trị dự báo và giá trị thực tế khoảng 41% - khá lớn. Ở hình 5, đường màu xanh hiển thị dữ liệu thực tế, trong khi đường màu cam thể hiện giá trị dự báo. Kích thước của dữ liệu thử nghiệm nằm trong khoảng từ 1 đến 50, phạm vi giá trị từ 98 đến 102. Theo biểu đồ, có một số điểm

cho thấy giá trị dự báo tiệm cận giá trị thực tế. Thậm chí mô hình còn dự báo được giai đoạn khủng hoảng vào tháng 4/2020 do Covid-19. MAPE của mô hình dự báo dài hạn là 0.517.

b) **Kết quả dự báo theo mô hình RNN**

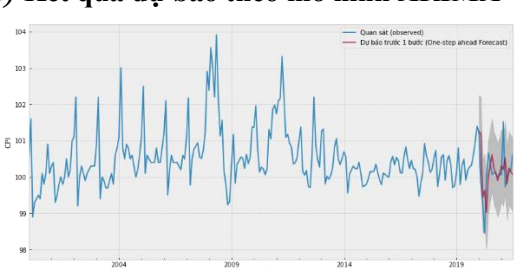


Hình 6: Kết quả dự báo RNN

Nguồn: Tác giả

Hình 6, đường màu đỏ là giá trị dự báo và đường màu xanh là giá trị thực tế. Về xu hướng, không có sự khác biệt đáng kể so với giá trị thực tế và giá trị dự báo. Tuy nhiên, giữa các điểm đạt đỉnh lại có nhiều khác biệt (giá trị dự báo thấp hơn nhiều so với giá trị thực tế). Để có giá trị dự báo chính xác hơn tại những điểm này, phải sử dụng thông tin trong quá khứ xa hơn, để xem bối cảnh của Việt Nam ảnh hưởng như thế nào đến xu hướng này. Khi đó, RNN có thể phải tìm thông tin dài hạn và tập dữ liệu trở nên rất lớn. Tuy nhiên RNN lại không thể học cách liên kết thông tin. Điều này làm kết quả dự báo thiếu chính xác ở những điểm đạt đỉnh.

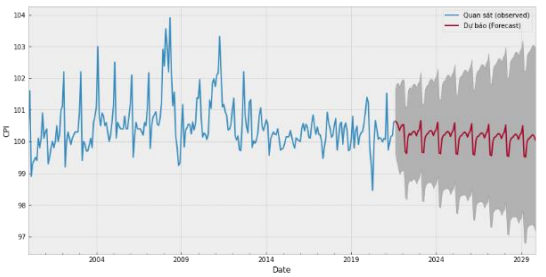
c) **Kết quả dự báo theo mô hình ARIMA**



Hình 7: Kết quả dự báo trước 1 bước

Nguồn: Tác giả

Hình 7 cho thấy các giá trị dự báo theo xu hướng tương đối phù hợp với các giá trị thực tế. MSE bằng 0.4 cho thấy dự báo có độ chính xác cao.



Hình 8: Kết quả dự báo 10 năm tới

Nguồn: Tác giả

Từ hình 8 có thể thấy chuỗi thời gian dự báo sẽ tiếp tục tăng giảm với tốc độ ổn định với các khoảng tin cậy càng lớn khi dự báo càng xa trong tương lai.

d) **Kết quả dự báo kết hợp**

Bảng 1*: Bảng so sánh kết quả dự báo*

Phương pháp	MAPE	MAE
LSTM	0.00407	0.00485
RNN	0.00414	0.00486
SARIMA	0.00422	0.00423
SARIMA kết hợp RNN	0.003911	0.391766
SARIMA kết hợp LSTM	0.004375	0.437422
Kết hợp 3 phương pháp	0.003906	0.390805

Nguồn: Tác giả

3.2. Đánh giá

Xét kết quả dự báo của 3 phương pháp: ARIMA, RNN, LSTM. Khi kết hợp hai phương pháp dự báo ARIMA và RNN, MAPE lần lượt giảm 4.06% và 11,16% (so với RNN và ARIMA). Kết hợp hai phương pháp dự báo ARIMA và LSTM, MAPE giảm 14,65% và 0,62%. Kết hợp cả 3 phương pháp dự báo với nhau bằng cách lấy trung bình các kết quả dự báo, MAPE và MAE lần lượt giảm 10,73% và 10,66% so với cách kết hợp 2 phương pháp. Như vậy, kết hợp dự báo bằng cả 3 phương pháp đem lại hiệu quả tốt nhất.

Từ Bảng 1 có thể thấy LSTM cho kết quả dự báo trong ngắn hạn tốt hơn mô hình SARIMA (MAPE bé hơn). Tuy nhiên, SARIMA lại dự báo tốt hơn trong dài hạn vì MAE nhỏ hơn. Mô hình SARIMA chủ yếu dựa trên dữ liệu theo mùa, có thể tận dụng tính thời vụ cao, đồng thời xem xét tất cả dữ liệu đầu vào cùng một lúc trong một tập dữ liệu để thiết lập xu hướng dài hạn. Mô hình LSTM sử dụng một tập dữ liệu giới hạn ở mỗi lần lặp, kích thước bị giới hạn. Điều này có nghĩa là mô hình bị giới hạn trong việc hình thành các phụ thuộc, hiệu suất vì thế kém hơn. Mặt khác, dù có dự báo bao nhiêu điểm thì SARIMA vẫn đào tạo mô hình theo cùng một cách. Mô hình LSTM vượt trội hơn hẳn so với mô hình SARIMA về mặt thời gian tính toán.

Tuy nhiên, khi dự báo kết hợp 3 phương pháp cho kết quả tốt nhất khi cả MAPE và MAE đều có giá trị thấp nhất. MAPE của dự báo kết hợp giảm 3,93% so với LSTM, và MAE giảm 7,56% so với ARIMA. Chính vì vậy, có thể nói dự báo kết hợp 3 phương pháp là tốt nhất trong cả ngắn hạn và dài hạn.

Để cải thiện chất lượng dự báo, nghiên cứu sau này có thể kết hợp các phương pháp khác như EEMD và SVR. Hướng nghiên cứu tiếp theo là có thể mở rộng lưới để xác định siêu tham số. Các biến số được phát hành hàng tháng, trong đó có lạm phát, là một nguồn phong phú để nghiên cứu thêm.

4. Kết luận

Bài viết đã sử dụng các mô hình SARIMA, RNN, LSTM và kết hợp dự báo để dự báo lạm phát hàng tháng. Kết quả cho thấy LSTM cho kết quả tốt hơn so với SARIMA trong ngắn hạn, nhưng kém hơn trong dài hạn và dự báo kết hợp cho kết quả dự báo tốt nhất ở cả ngắn hạn và dài hạn. Tuy nhiên, kết quả dự báo chưa thực sự tốt như mong đợi vì một phần của dữ liệu thuộc giai đoạn dịch bệnh, do đó có sai số lớn trong dự báo lạm phát. Hơn nữa, bộ dữ liệu chỉ có 259 quan sát, số quan sát chưa thực sự lớn để mô hình học máy thể hiện được hết sự ưu việt của mình so với các mô hình dự báo truyền thống.

Chúng ta có thể xem xét phát triển các nghiên cứu sau này dựa trên các bộ dữ liệu có số quan sát lớn hơn (chẳng hạn như thị trường chứng khoán, giá dầu, giá vàng, tỉ giá hối đoái…). Hoặc có thể tiếp tục nghiên cứu lại mô hình SARIMA, khám phá tính mùa vụ của dữ liệu, nghiên cứu thêm các mô hình học máy truyền thống khác như mô hình hàm mũ. Bên cạnh đó việc cố gắng tìm ra một cách để chạy mô hình ARIMA mà không cần lấy mẫu dữ liệu là cần thiết vì các hạn chế về tính toán. Điều này cũng sẽ cho phép so sánh chính xác hơn giữa ARIMA và LSTM. Chúng ta có thể làm nhiều việc hơn nữa trong việc tìm hiểu các xu hướng được thiết lập bởi các mô hình LSTM.

TÀI LIỆU THAM KHẢO

Armstrong J. S (2001), “Principles of forecasting: a handbook for researchers and practitioners, Kluwer Academic Publishing,” 2001, pages 417-439.Nguyễn Quang Dong, Nguyễn Thị Minh (2013). Giáo trình Kinh tế lượng, NXB ĐHKDTQ.

Hyeong Kyu Choi, 2018. "Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model," Papers 1808.01560, arXiv.org, revised Oct 2018.

Newbold, Paul, and Clive W.J. Granger. 1974. “Experience with Forecasting Univariate Time Series and the Combination of Forecasts.” Journal of the Royal Statistical Society 137(2): 131-165.

Phạm Nguyễn Hoàng Phúc, Trương Tấn Phát (2020), "Ứng dụng Long Short-term Memory trong dự đoán tài chính". Khoa Công nghệ Thông tin, Trường Đại học Công nghệ TP. Hồ Chí Minh.

Sofiyanti, N.; Fitmawati, D.I.; Roza, A.A. Understand LSTM Networks. GITHUB Colah Blog 2015, 22, 137-141.

S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

T. Gao, Y. Chai, and Y. Liu (2018), "Applying Long Short Term Memory Neural Networks for Predicting Stock Closing Price." In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). pp. 3-6. Beijing, China 24-26 Nov. 2017.

Temür, A.S.; Akgün, M.; Temür, G. Predicting housing sales in Turkey using ARIMA, LSTM and hybrid models. J. Bus. Econ. Manag. 2019, 20, 920-938.

Yang, Yuhong. 2004. "Combining Forecasting Procedures: Some Theoretical Results." Econometric Theory 20(1): 176-222.