Modeling and Forecasting the Volatility of Vietnam's Stock Index (VN-Index): A Deep Learning and ARMA-GARCH Approach

***Phung Duy Quang*[1]**

Faculty of Technology and Data Science - Foreign Trade University.
Email: quangpd@ftu.edu.vn

Nguyen Khanh Huyen

K61 - Faculty of Economics and International Business - Foreign Trade University.
Email: khhuyen1601@gmail.com

Hoang Nam Quyen

K61 - Faculty of Economics and International Business - Foreign Trade University.
Email: k61.2211530042@ftu.edu.vn

Nguyen Thi Linh Dan

K60 - Faculty of International Economics - Foreign Trade University.
Email: linhdannguyenthidl@gmail.com

Nguyen Hai Minh

K66 - School of Information and Communication Technology - Hanoi University of Science and Technology.
Email: minh.nh210611@sis.hust.edu.vn

Bui Đuc Duong

Luong The Vinh Secondary and High School.
Email: duongbuiduc@outlook.com

Nguyen Dang An

Hanoi University of Science and Technology.
Email: trandangan012003@gmail.com

Dang Bui Nam

K70 - Faculty of Mathematics and Informatics, Hanoi National University of Education.
Email: dang.nam.hnue@gmail.com

HIGHLIGHTS

Abstract

This study focuses on forecasting the volatility of Vietnam's Stock Index (VN-Index) by comparing the effectiveness of deep learning models—namely Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM)—with traditional econometric models such as ARMA-GARCH. The dataset comprises daily closing prices of Vietnam's Stock Index from 2000 to 2024, totaling 5,867 trading days. Two approaches are employed to evaluate the deep learning models: one uses a distance-based loss function (Mean Squared Error, MSE), and the other employs a likelihood-based loss function. The findings reveal that LSTM with the likelihood-based loss function delivers the most accurate forecasts, achieving the highest log-likelihood value on the test set (1663.98), outperforming all other models, including DNN and ARMA-GARCH. Although DNN demonstrates strong performance, it still lags behind LSTM in most cases. Meanwhile, the traditional ARMA-GARCH model exhibits significant weaknesses, particularly in handling nonlinear and highly volatile data. Furthermore, the study illustrates the capacity of deep learning models to forecast long-term volatility and their robustness when applied to various data samples. Visual comparisons further affirm the superior predictive accuracy of LSTM over the other approaches.

Keywords: Modelling, forecasting, DNN models, LSTM models, ARMA models, GARCH model, VN-index.

1. Introduction

1.1. Background

[1] Corresponding author

The stock market plays a pivotal role in the global economy, particularly in emerging markets such as Vietnam. First and foremost, it serves as a key channel for businesses to mobilize capital, enabling them to expand production, invest in technology, and develop infrastructure. This, in turn, not only drives corporate growth but also stimulates the entire economy. Additionally, the stock market offers investment opportunities for both individuals and institutions, facilitating efficient capital allocation and resource optimization across society. Moreover, the stock market acts as an essential barometer of the economy's health, assisting policymakers in adjusting fiscal and monetary policies. In the context of emerging markets like Vietnam, the development of the stock market also helps attract both direct and indirect foreign investment, thereby promoting sustainable economic growth. However, forecasting the fluctuations of Vietnam's Stock Index faces numerous challenges due to the uncertain and complex nature of the market. Factors such as macroeconomic volatility, the impact of international policies, and unforeseen events make accurate predictions increasingly difficult. Furthermore, investor sentiment—often irrational—can trigger strong and unpredictable market swings, adding yet another layer of complexity to the forecasting process.

1.2. Research Objectives

The primary goal of this study is to develop a model for forecasting the volatility of Vietnam's Stock Index (VN-Index) by comparing the performance of deep learning models—namely DNN and LSTM—with that of traditional econometric models such as ARMA-GARCH. The dataset comprises daily closing prices of the VN-Index from 2000 to 2024, totaling 5,867 trading days.

The research steps are illustrated in Figure 1 below.

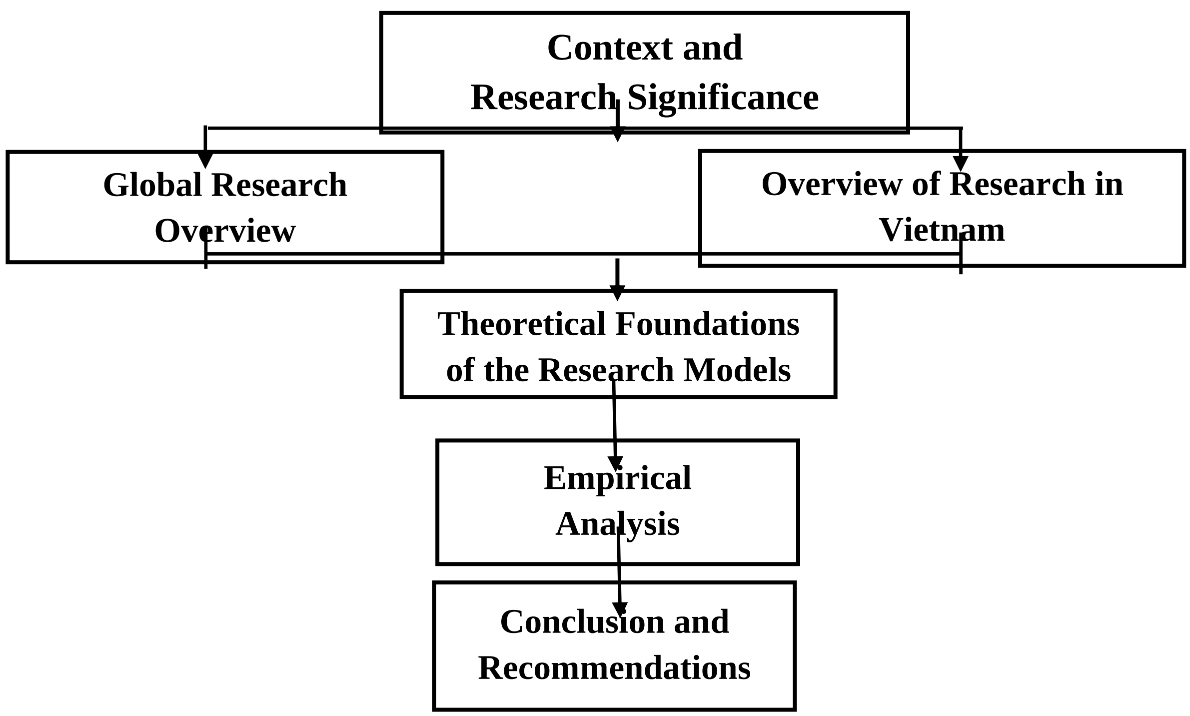


Figure 1. Research approach

Source: Authors' research and compilation

1.3. Structure of the Paper

This paper is organized as follows:

Section 1 introduces the research context, objectives, and approach.

Section 2 provides a review of the related literature. In this section, we summarize research findings on stock price volatility forecasting models in both Vietnam and worldwide.

Section 3 presents the theoretical underpinnings of the models. Here, we introduce the ARMA-GARCH model, the Deep Neural Network (DNN) model, the Long Short-Term Memory (LSTM) model, the experimental approach for deep learning, and the criteria for model comparison.

Section 4 discusses the empirical analysis. We use daily closing prices of the VN-Index from 2000 to 2024, totaling 5,867 trading days. Employing the Python programming language, we conduct statistical analyses and estimate the ARMA-GARCH, DNN, and LSTM models to forecast the volatility of Vietnam's Stock Index (VN-Index).

Section 5 concludes the paper and provides recommendations based on the empirical findings.

2. Literature review

Recognizing the pivotal role of the stock market in national economies, both domestic and international researchers have increasingly focused on forecasting stock indexes.

2.1. Global literature overview

In the study by D. Banerjee [1], the ARIMA model was employed to forecast India's Sensex Index using monthly closing data over a six-year period (2007–2012). Z. Lin [2] conducted research on modeling and predicting the volatility of the SSE Composite Index using GARCH models. The findings indicated that, from a time-series perspective, the SSE Composite Index exhibits pronounced volatility clustering and persistence. Its distribution displays leptokurtosis with evident ARCH and GARCH effects. Moreover, by comparing the fitting and prediction performance of the symmetric GARCH(1,1) model against the asymmetric TARCH(1,1) and EGARCH(1,1) models, the study concluded that EGARCH(1,1) outperforms the other models.

ARIMA has also been adopted by C. D. Reedy [3] to forecast stock index trends. The results showed that the ARIMA(0,1,0) model was the best fit for predictions from January 7, 2018, to June 3, 2018 (22 projected values), using weekly data from January 6, 2014, to December 31, 2017 (187 observed values). Similarly, T. Mashadihasanli [4] employed ARIMA to forecast monthly stock prices in Istanbul from January 2009 to March 2021. Predictions were made using the ARIMA(3,1,5) model, indicating a downward trend in Istanbul's stock price index over the forecast horizon.

Deep learning models, such as DNN and LSTM, have also demonstrated significant potential in forecasting stock indexes, as affirmed by numerous researchers worldwide. Specifically, D. Shah, W. Campbell and F. H. Zulkernine [5] compared LSTM and DNN models for predicting the daily and weekly movements of India's BSE Sensex. Daily stock price forecasts for Tech Mahindra (NSE: TECHM) were used to evaluate each model's generalization capabilities. The findings showed that both networks performed well for daily forecasting and generalized effectively using Tech Mahindra's data. However, LSTM-RNN outperformed DNN in weekly forecasts, underscoring its greater potential for long-term predictions.

A. M. Rather [6] proposed a novel methodology for time-series stock price forecasting and portfolio construction based on those forecasts. To achieve this, a new regression mechanism was deployed on a deep neural network built upon LSTM. Experiments involving the NIFTY-50 stocks from India's National Stock Exchange revealed that the proposed model outperformed standard forecasting and portfolio optimization frameworks. In a similar vein, C. R. Karthik, Raghunandan,B. A. Rao and N. V. S. Reddy [7]compared two neural network models for predicting daily volatility in the NIFTYIT index on both the BSE and NSE markets. This study contrasted DNN with LSTM, a form of recurrent neural network (RNN). Results indicated that both models performed well in daily forecasting and demonstrated impressive generalization to NiftyIT data; however, LSTM-RNN again proved superior in predictive accuracy, highlighting its promise for long-term forecasting.

In the study by K. Alam, M. H. Bhuiyan,I. U. Haque, M. F. Monir and T. Ahmed [8], the authors proposed a hybrid DNN-LSTM model to address the challenges of stock market prediction, particularly in relation to volatility and complex patterns. Their research demonstrated the robustness and scalability of this hybrid model through careful comparisons across 26 stock datasets, achieving an average R-squared (R) of 0.98606, a mean absolute error (MAE) of 0.0210, and a mean squared error (MSE) of 0.00111. Additionally, the study included an ablation analysis on the LSTM-DNN model to assess the role of each component in predicting stock closing prices—offering crucial insights for optimizing future market forecasting models. The findings showed the model's superior performance, setting a new benchmark in market prediction and opening up multiple opportunities for investors, traders, and financial experts.

U. M. Sirisha,M. C. Belavagi and G. Attigeri [9] conducted a study comparing the forecasting accuracy of ARIMA, SARIMA, and LSTM in predicting stock returns. The data were converted into a stationary time series for ARIMA but not for SARIMA or LSTM. After achieving satisfactory accuracy levels of approximately 93.84% (ARIMA), 94.378% (SARIMA), and 97.01% (LSTM) on the test data, the authors proceeded to forecast returns for the subsequent five years. Results showed that LSTM outperformed both statistical models in achieving the best-fitting forecast. V. Goyal and R. R. Raj [10] further examined artificial neural networks from an analytical perspective to compare their performance in predictive modeling. Their findings indicated that traditional regression models were less effective compared to the strong performance demonstrated by neural networks.

2.2. Overview of research in Vietnam

In Vietnam, the ARIMA and GARCH models have also been employed to forecast the Vietnam's Stock Index (VN-Index). Nguyen Thanh Huong and Bui Quang Trung [11] applied these two models to predict the VN-Index in the context of Vietnam's stock market. A comparison of three models indicated that the combined ARIMA(1,1,1)-GARCH(1,1) model delivered better forecasts than the individual ARIMA(1,1,1) and GARCH(1,1) models. Le Van Tuan and Phung Duy Quang [12] used the GARCH model to capture and forecast the volatility of the VN-Index, the benchmark for Vietnam's stock market. Their findings suggested that GARCH(1,1) was the most suitable model for capturing VN-Index volatility. Simulation results also revealed that the impact of the first wave of the COVID-19 pandemic on Vietnam's stock market persisted for a considerable period, suggesting that the market might require approximately three years and three months to return to pre-pandemic levels.

In 2024, Phung, Duy Quang and et al. [13] employed the ARMA-GARCH model to estimate and forecast Bitcoin price volatility using daily closing prices from January 17, 2021, to December 17, 2023, representing a total of 1,065 observations. Model estimation was performed using data from this three-year period (2021–2023), while the remaining data (January 1, 2024, to January 17, 2024) were used for out-of-sample forecasting. The ARIMA-GARCH framework proved robust for time series with non-seasonal components, and the chosen model was based on the lowest corrected Akaike Information Criterion (AICc) and the highest log-likelihood value. Through the Box–Jenkins methodology, multiple AR and MA lags were tested to identify the optimal specification. ARIMA(12,1,12) was ultimately deemed the best-fitting model based on its AIC. Given the volatility inherent in financial time series, such as Bitcoin returns, GARCH(1,1) was then employed to account for this heteroscedasticity.

Nevertheless, ARIMA, GARCH, and linear regression models have limitations when forecasting stock indexes, particularly in highly volatile markets like Vietnam. ARIMA encounters difficulties with nonlinear market data and relies solely on historical information, rendering it less effective when confronted with major fluctuations or unforeseen events. GARCH is suitable for modeling volatility but assumes normal distribution, does not incorporate external factors, and is less accurate for long-term forecasts. Linear regression only captures linear relationships, failing to reflect the dynamic changes and nonlinear elements in the stock market, and it is heavily dependent on the correct selection of independent variables.

By contrast, DNN and LSTM models offer significant advantages. DNNs can learn complex nonlinear relationships and handle large datasets, facilitating more accurate predictions. LSTM excels in time-series forecasting, thanks to its capacity for storing long-term information, thereby better capturing trends and longer-term volatility. Such strengths enable DNN and LSTM to greatly enhance predictive performance in the highly volatile, nonlinear environment of the Vietnamese stock market.

When comparing ARIMA and LSTM-RNN for VN-Index forecasting, Nguyen Trong Co and et al. [14] found that the LSTM-RNN model outperforms ARIMA in predicting the VN-Index and when applying LSTM-RNN, the window size converges if it exceeds 15 days for LSTM-256 or 10 days for LSTM-512. Based on these results, the authors concluded that using one month of data in the LSTM-RNN model is optimal, aligning with Wave Theory and market psychology.

In H. S. Nguyen [15], the author conducted an empirical study comparing LSTM and ARIMA performance for forecasting the next 20 days of the S&P 500 Index, based on historical daily data from January 2000 to January 2024. The results showed that LSTM clearly outperformed ARIMA in terms of mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). Specifically, LSTM reduced MAE and RMSE by about 82–84% on average, while MAPE decreased by 3.96%. Furthermore, the LSTM forecasts exhibited high stability with minimal fluctuation across different data samples, whereas ARIMA displayed greater variance, emphasizing the potential of deep learning methods to enhance accuracy and set new benchmarks in forecasting.

In Vietnam, the work of Dao Le Kieu Oanh and Nguyen Thi Minh Chau [16] on using machine learning to forecast Vietnam's Stock Index found that an LSTM-based approach can deliver highly accurate volatility predictions of the VN-Index, with LSTM emerging as the most effective method overall. Similarly, the conclusion that LSTM architectures are well-suited for both short-term and long-term stock forecasting was also reached by Truong Cong Doan and et al. [17], who applied this approach to large-scale data, including the VN-Index.

2.3. The HoSE exchange and the VN-Index

HoSE stands for the Ho Chi Minh City Stock Exchange in Vietnam, established in July 2000. It operates under the supervision of the State Securities Commission and manages a listing-based trading system in the country. With an initial charter capital of over VND 1,000 billion, HoSE is currently Vietnam's largest stock exchange, providing a platform where most Vietnamese enterprises list and offer their securities to the market. In other words, it serves as an intermediary enabling businesses to access and mobilize capital in the secondary market.

The stock price index traded on HoSE is known as the VN-Index, which encompasses nearly all Vietnamese companies listed on the exchange. As of December 31, 2024, HoSE officially lists and trades 393 stocks, 16 ETF certificates, 4 closed-end fund certificates, and 114 covered warrants (CW). The total volume of listed securities reached more than 168.54 billion units, with a total market value of VND 1.67 quadrillion—an increase of 8.82% in volume and 9.34% in value compared to the end of 2023.

In 2024, HoSE licensed and launched 155 new securities for trading, comprising 10 stocks (with a total of over 2.54 billion shares), 2 new ETF certificates (exceeding 10.3 million ETF units), 1 closed-end fund certificate (17.28 million units), and 142 covered warrants (CW) (over 1.65 billion CW). Meanwhile, HoSE continued to evaluate the quality of listed securities, delisting 11 stocks that no longer met listing and trading standards, amounting to 2.18 billion shares in total. In addition, 257 CW issues—totaling nearly 2.44 billion warrants—were delisted upon reaching their maturity date.

3. Data and methodology

3.1. Data

The data utilized in this study consists of the daily closing prices of Vietnam's Stock Index (VN-Index) from July 31, 2000, to December 16, 2024 (encompassing 5,931 trading days), gathered from Investing.com. According to L. Zhang et al. [18], the most commonly analyzed univariate financial time series in the finance sector is the daily closing price. Accordingly, the returns of Vietnam's Stock Index are calculated using the following formula:

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right) \tag{1}$$

where P_t, P_{t-1} denote the closing prices of Vietnam's Stock Index at time t, t-1.

Table 1 presents the summary statistics for these returns r_t . Overall, Vietnam's Stock Index exhibited a positive average return during the sample period. The standard deviation of the return series is 0.014. In addition, the index shows negative skewness and high kurtosis. The Jarque–Bera test confirms that this return series does not follow a normal distribution.

Table 1. Descriptive statistics of the return series of VN-Index

	Mean	**Maximum**	**Minimun**	**Std.dev**
r_t	0.000425	0.066561	-0.076557	0.014439
	Skewness	**Kurtosis**	**Jarrque-Bera test**	**Observations**
r_t	-0.395399	3.409743	0.0	5931

Source: Author's calculations**.**

3.2. Deep neural networks (DNN)

Artificial Neural Networks (ANN) are among the most renowned machine learning algorithms. They are designed to mimic the structure of neurons in the human brain. In this model, artificial neurons are interconnected, and the networks can develop problem-solving capabilities by adjusting their connection weights through the learning process. Deep Neural Networks (DNN) are ANNs with a certain level of complexity. By definition, a DNN is a neural network that has two or more hidden layers. Figure 2 illustrates the typical architecture of a DNN. Inputs are fed into the model through the input layer. The hidden layers and the output layer are computed sequentially by multiplying the previous layer with the connection weights, and an activation function is applied at each step.

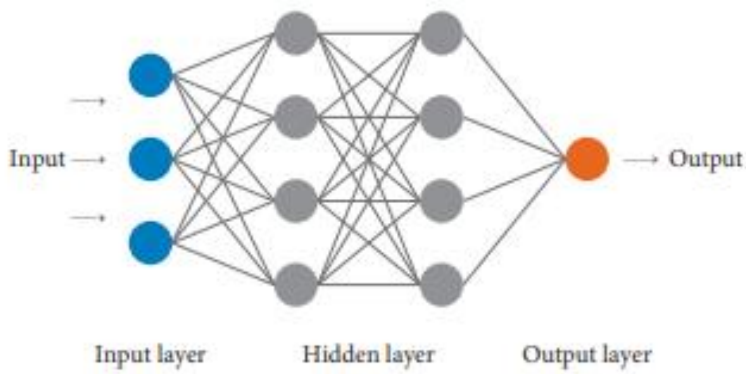


Figure 2. Deep neural network (DNN) architecture
Source: Authors' research and compilation

To enable the DNN model to function, it must be trained. By utilizing the predicted values obtained from the output layer and the target values, we can formulate a loss function. The connection weights in the DNN are learned by minimizing the loss function. We primarily employ forward and backward propagation to handle the computational process during network optimization.

3.3. Long short-term memory (LSTM)

Long Short-Term Memory (LSTM) is a specialized type of deep neural network developed by S. Hochreiter and J. Schmidhuber [19]. By inheriting the advantages of Recurrent Neural Networks (RNNs), LSTM has demonstrated significant strength in modeling sequential data, such as time series. Compared to vanilla RNNs, LSTM utilizes gates to control the flow of information through the sequence, enabling it to learn long-term dependencies and address the vanishing gradient problem. Figure 3 illustrates the typical architecture of an LSTM. Similar to vanilla RNNs, inputs are fed into the model at each time step, and outputs can be produced at each time step.

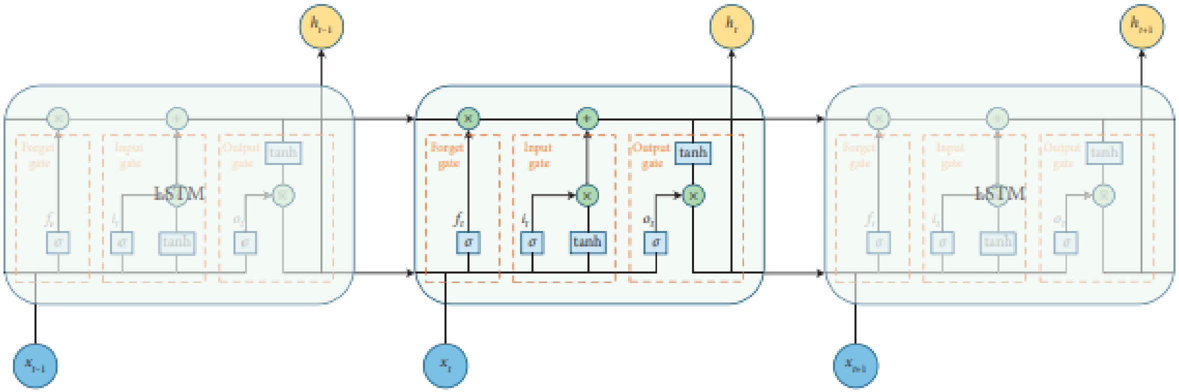


Figure 3. LSTM network architecture
Source: Authors' research and compilation

Specifically, LSTM employs memory blocks to replace neurons in RNNs. A memory block consists of a memory cell, an input gate, a forget gate, and an output gate. We can describe LSTM using vector equations as follows:

$$X=\begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}, f_t=\sigma(W_t X+B_f), i_t=\sigma(W_i X+B_i), o_t=\sigma(W_o X+B_o),$$
$$c_t=f_t\nabla c_{t-1}+i_t\nabla \tanh(W_c+B_c), h_t=o_t\nabla \tanh(c_t), \tag{2}$$

Where h_t denotes the hidden state at time t, x_t represents the inputs at time t. σ is the activation function, with the logistic sigmoid function being commonly utilized, f_t represents the forget gate, i_t represents the input gate, and o_t represents the output gate, and ∇ denotes element-wise multiplication.

3.4. Deep learning model with an empirical approach

As discussed in the Introduction, we utilize a likelihood-based loss function to train our DNN and LSTM models. If r_t represents the observed return series from time 1 to time T, and σ_t represents the volatility forecasted by our models at the same time step, the sample likelihood can be calculated using the following function:

$$L=\prod_{i=1}^{T}\frac{1}{\sqrt{2\pi}\sigma_t}e^{\frac{-r_t^2}{2\sigma_t^2}} \tag{3}$$

Then, we can then derive the log-likelihood function as follows:

$$\log L=\sum_{i=1}^{T}\left(\frac{-1}{2}\log(2\pi)-\log\sigma_t-\frac{r_t^2}{2\sigma_t^2}\right) \tag{4}$$

The log-likelihood function should be maximized through optimization, whereas deep learning models are always trained by minimizing their loss functions. Therefore, we need to use the negative log-likelihood as the loss function in our deep learning models. To further reduce the computational cost of deep learning models, we employ a simplified function as the loss function for our DNN and LSTM models:

$$\text{loss}=\sum_{i=1}^{T}\left(2\log\sigma_t+\frac{r_t^2}{\sigma_t^2}\right) \tag{5}$$

When evaluating all the models discussed in this paper, we use Equation (5) to compute the log-likelihood function for the test sample and compare the resulting values.

Regardless of whether during training or testing, the forecasting process is implemented on a daily basis. This means that volatility is forecasted within a rolling window, which advances by one trading day at each step. After forecasting volatility on a daily basis and obtaining the volatility series σ_t from time 1 to time T, we can use it to compute Equations (4) and (5) in various procedures.

3.5. ARMA-GARCH model

The ARMA-GARCH model is a widely used time series model for studying volatility. We will utilize this model to forecast the volatility of the VN-Index stock return series and compare its performance with deep learning models for forecasting the same series. T. Bollerslev [20] generalized the GARCH model from R. F. Engle's [21] ARCH model. The general ARMA(m, n) – GARCH(p, q) model is structured as follows:

First, the return series r_t is identified using the ARMA(m, n) model, expressed as:

$$r_t = \mu + \sum_{i=1}^{m} \theta_i \left(r_{t-1} - \mu \right) + \sum_{j=1}^{n} \gamma_j \varepsilon_{t-j}, \tag{6}$$

where $\mu_t = E\left(r_t \middle| F_{t-1}\right)$, F_t represents the set of information available at time t, and ε_t denotes the error term.

Subsequently, ε_t satisfies the following equation:

$$\varepsilon_t^2 = z_t^2.\sigma_t^2, \tag{7}$$

Where $z_t \sim N(0;1)$ is a standard normal random variable, and $\sigma_t^2 = \mathrm{Var}\left(r_t \middle| F_{t-1}\right)$ is the conditional variance modeled using the GARCH(p,q) process.

ARCH (p) model

The ARCH (Autoregressive Conditional Heteroskedasticity) model is the foundational model for capturing stochastic variance, first introduced by R. F. Engle [21]. The ARCH model assumes that the variance of the error terms at time t depends on the squared errors from previous periods. The ARCH model is expressed as follows:

$$\sigma_t^2 = \alpha_o + \alpha_1 \varepsilon_{t-1}^2 + \ldots + \alpha_q \varepsilon_{t-q}^2 \tag{8}$$

where σ_t^2 represents the volatility of r_t, p is the autoregressive order, $\alpha_o > 0$ and $\alpha_i \geq 0\,(\mathrm{i} = 1, 2, \ldots, \mathrm{q})$. To ensure that the ARCH model described by Equation (8) is weakly stationary, the following condition must be satisfied:

$$\sum_{i=1}^{q} \alpha_i < 1 \tag{9}$$

ARCH (q) model used in forecasting financial time series data has the following advantages:

(1) Firstly, it considers data types where its variance depends on past variance values to estimate risk levels.

(2) Secondly, it forecasts the volatility of highly volatile financial time series.

In addition to the advantages, the model has the following disadvantages: there is no standard criterion for determining the order q of the model. One method used is the likelihood ratio test; however, this method is not the best, as the q value of the residuals can be a very large number to adequately control all dependencies of the conditional variance. This leads to an unbounded conditional variance model, where the non-negativity constraint of the variance can be violated. If everything remains the same, the more parameters in the conditional variance equation, the higher the likelihood of encountering negative variances. (L. Kalyanaraman [22]).

Generalized ARCH model (GARCH)

Due to some limitations of the ARCH model, it has been replaced by the Generalized ARCH model (GARCH) independently proposed by T. Bollerslev [20]. Based on the ARCH model, the GARCH model adds lagged conditional variance terms σ_{t-j}^2 as new terms. Mô hình GARCH giúp giảm số lượng tham số cần ước lượng. The GARCH model helps reduce the number of parameters that need to be estimated. In this model, the conditional variance remains a linear function of its own lags and errors, and is described by the following formula:

$$\sigma_t^2 = \alpha_o + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{10}$$

where q denotes the order of ε_{t-i}^2 , p denotes the order of σ_{t-j}^2. To ensure a positive value, a sufficient condition for the conditional variance is $\alpha_o > 0; \alpha_i \geq 0 (\mathrm{i} = 1, 2, ..., \mathrm{q})$ and $\beta_j \geq 0 (\mathrm{j} = 1, 2, ..., \mathrm{p})$.

The GARCH(p, q) model will be stationary if and only if:

$$\sum_{i=1}^{q} \alpha_i + \sum_{j=1}^{p} \beta_j < 1 \tag{11}$$

Moreover, the GARCH model not only retains all the characteristics of the ARCH model but also incorporates a linear function of lagged conditional variances. Therefore, the GARCH model is an extension of the ARCH model. The GARCH model posits that the current variance ε_t^2 depends on both the past values of shocks, represented by the lagged squared noise terms, and the past values of the variance itself ε_t^2, represented by the lagged variance terms ε_{t-i}^2 . If p=0, then the GARCH(0,q model simplifies to the ARCH(q) model. The simplest form of the GARCH model is the GARCH(1,1) model..

The GARCH model has the advantage of explaining how investors forecast the variance of assets in the current period by generating a long-term average weight and forecasting variance based on previous periods, incorporating information on volatility from past periods. It considers data types where its variance depends on past variance values to estimate risk levels and forecast the volatility of highly volatile financial time series. The drawback of the model is that it can explain variance anomalies by solely using its own past noise information, unable to distinguish the impact of positive and negative shocks in later periods on current period returns (R. F. Engle [21]).

3.6. Model comparison criteria

As discussed, most of the referenced papers employing machine learning for volatility forecasting use the discrepancy between the estimated volatility and the forecasted volatility as the loss function to train their models. To compare this popular approach with our deep learning models that utilize a likelihood-based loss function, we proceeded to test DNN and LSTM models with a similar discrepancy-based loss function. This allows us to investigate whether deep learning models can still achieve good forecasting performance when trained by minimizing a discrepancy loss function and evaluated by calculating the likelihood value on the test samples. Since our training method is inconsistent with our testing method, deep learning models with discrepancy-based loss functions are at a disadvantage when compared to the other models presented in this paper.

The discrepancy loss functions chosen for DNN and LSTM are Mean Squared Error (MSE). This loss function is the most commonly selected among related studies. Similarly to these studies, the actual volatility is calculated using the standard deviation of the return series within a 21-day trading window, which is close to the average number of trading days in a month. The activation function of the output layer is set to a linear function, which is also a popular choice for deep learning models with discrepancy-based loss functions when forecasting volatility. All other settings are consistent with our deep learning models that use a likelihood-based loss function.

When training the ARMA(m,n)-GARCH(p,q) model, we set m, n, p and q from 1 to 3 and estimated all possible combinations. The model with the minimum Bayesian Information Criterion (BIC) was selected and used to forecast the test set. This straightforward method simply involves selecting the optimal window length based on the training set, as discussed in the previous section.

4. Model estimation

Below is a chart of returns over a 42-day period. In this chart, we can observe the volatility cycle within each week. In the first two weeks, volatility is relatively high. However, in the subsequent weeks, volatility decreases. Therefore, if we use data from the first two weeks to predict the following weeks, the forecasts will be relatively unstable for each week. Conversely, if we use data from the last two weeks to predict, the forecast results will be completely different. This raises the issue of the importance of data partitioning in the forecasting process.

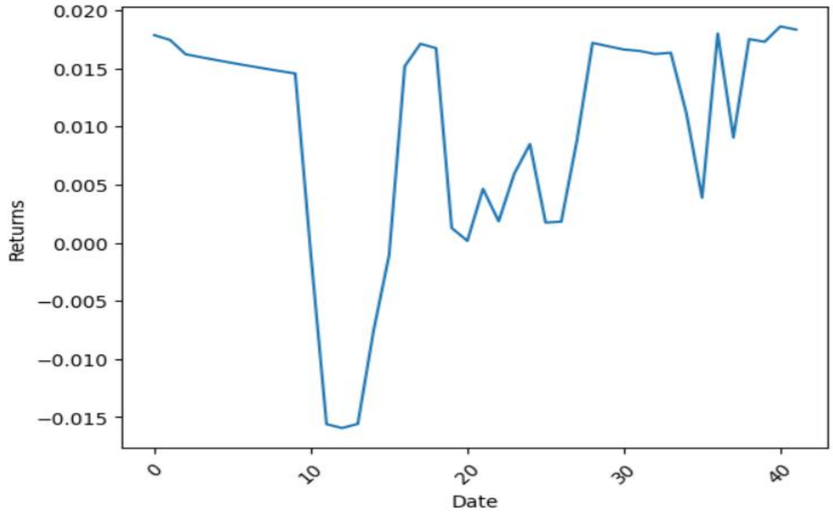


Figure 4. Returns of the VN-Index series over a 42-day period

Source: Authors' modification

This issue will be addressed by using the parameters lookback, steps, and delay to shape the input data for the prediction models. Specifically, Lookback is the number of previous time steps the model will consider to predict future prices. Steps is the interval between consecutive samples within the lookback window. Delay is the number of future time steps the model needs to predict after considering the lookback steps.

The parameter values used are as follows::

Lookback = 10 – The model will observe the previous 10 days to predict the next day's price.

Steps = 1 – The model will sample each day.

Delay = 1 – The model will predict the price value immediately after the lookback days.

Other settings for our DNN model are as follows: The number of input layers is set to 10, meaning we use a sequence of 10 return values as input each time. There are 2 hidden layers, with the first hidden layer having 40 units and the second hidden layer having 20 units. The activation function for the hidden layers is chosen to be ReLU, and the activation function for the output layer is set to sigmoid. RMSprop is used as the optimizer for training the model. The batch size is set to 32. The dropout rate is set to 0.3.

For the LSTM model, the input sequence length is set to 10, and each LSTM layer has 40 units. The fully connected layer has 20 units with ReLU activation. The activation function of the output layer is set to sigmoid. RMSprop is also used as the optimizer for training the LSTM model. The batch size is set to 32. We also use a dropout rate of 0.3 during the training process of the LSTM model.

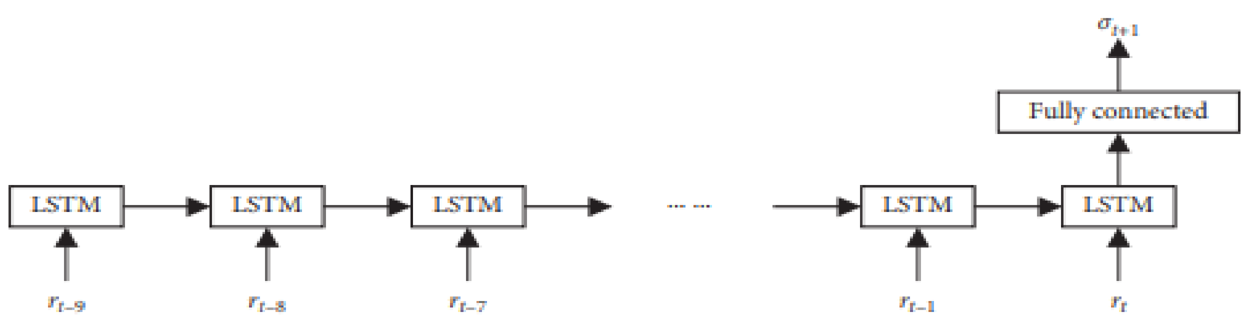


Figure 5. LSTM model

Source: Authors' modification

A notable aspect of our study is that we do not use a linear activation function in the output layer. If using a discrepancy-based loss function as in other studies, a linear activation function might suffice. However, when using a probability-based loss function as in our study, some activation functions like linear cannot help the model converge during training; thus, the sigmoid activation function is appropriate for successful training of the models.

All models in this study are implemented using the Keras library.

For deep learning models, we split the dataset into three parts: 80% training set, 10% validation set, and the remaining 10% test set. The ARMA-GARCH model does not require a validation set during training, so we combine the training and validation sets into a single training set with 90% of the data and keep the test set unchanged. The training and test sets of the models are split in chronological order, so the test sets are the same across models. Therefore, models can be compared by comparing the log-likelihood values of the test set. Additionally, we also split the data into ratios of 70% training, 15% validation, 15% test, and 60% training, 20% validation, 20% test to test the stability of the models.

4.1. Experimental results

When we trained the LSTM model using the likelihood-based loss function, we obtained the loss values for both the training and validation sets at each epoch, then plotted the learning curves as shown in the figure below. For the VN-Index, the model reached its lowest validation loss at epoch 387, with a validation loss value of −7.78478. The figure also illustrates how the loss changes over the training and validation sets during the training process, indicating positive signs of the model learning features from the data. Although the training loss fluctuated, its downward trend over time shows that the model progressively learns and improves. Notably, the validation loss remained stable and was consistently lower than the training loss, suggesting that the model generalizes well and effectively avoids overfitting.

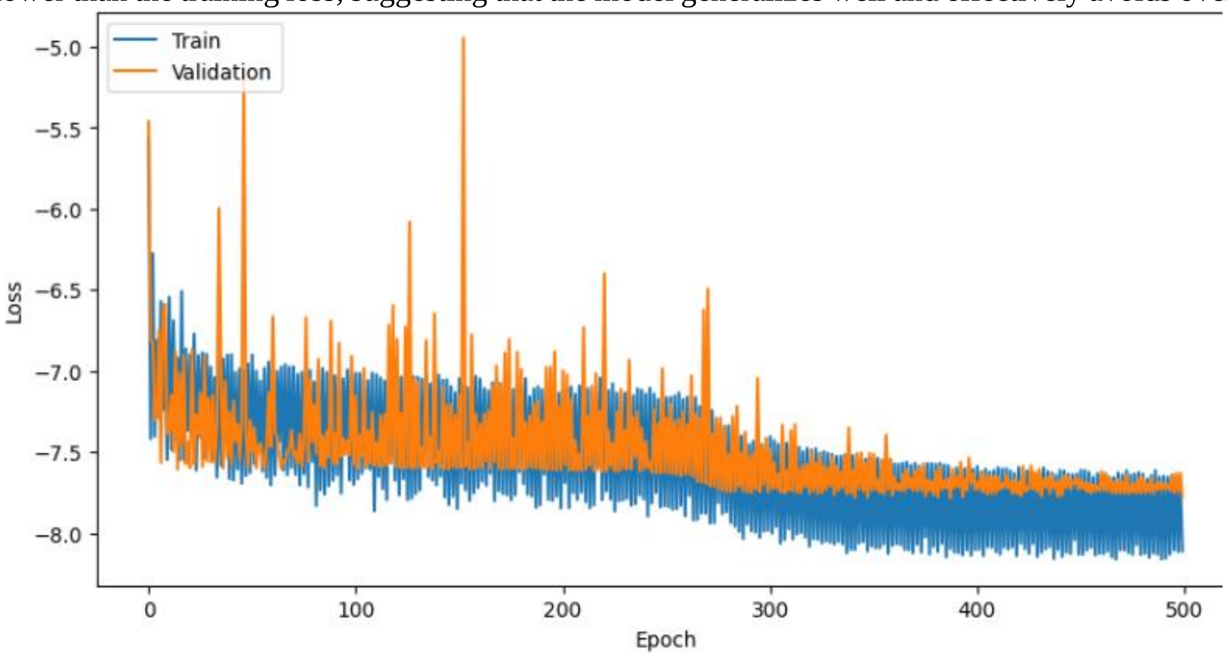


Figure 6. Learning curve during LSTM (Likelihood-Based) model training

Source: Authors' modification

When we trained the DNN model using the likelihood-based loss function, we likewise recorded the training and validation loss values at each epoch and plotted the learning curves, as shown in the figure below. For the VN-Index, the model achieved its lowest validation loss at epoch 441, with a validation loss of −7.79503. The chart below illustrates how the training and validation losses evolved over 500 epochs, indicating positive signs regarding the model's performance.

The training loss clearly trends downward, reflecting that the model is progressively learning and improving its predictive capabilities. Although some fluctuations occur during training, they gradually diminish over time, suggesting stable convergence in the later stages. Notably, the validation loss remains low and stable while also staying below the training loss—a strong indication that the model generalizes effectively and avoids overfitting.

Furthermore, the gap between the two loss curves narrows as the epochs progress, indicating increasing similarity between performance on the training and validation sets. Overall, this chart demonstrates that the model is functioning efficiently.

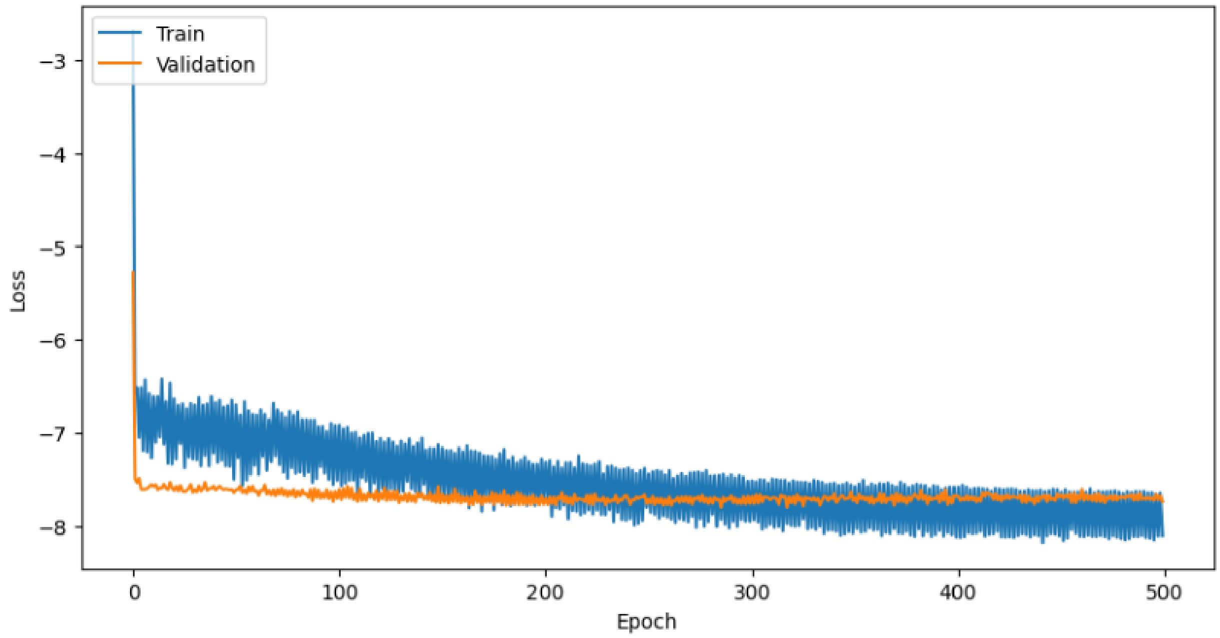


Figure 7. Learning curve during DNN (Likelihood-Based) model training

Source: Authors' modification

When we trained the LSTM model using the Mean Squared Error (MSE) loss function, we observed that the loss values for both the training and validation sets decreased more rapidly compared to training the LSTM model with a likelihood-based loss function. Similar results were also noted when training the DNN model. This suggests that deep learning models utilizing the MSE loss function generally require fewer computational resources compared to those using a likelihood-based loss function.

Subsequently, we proceeded to identify the most suitable ARMA(m,n)-GARCH(p,q) model for the VN-Index. After estimating ARMA(m,n)-GARCH(p,q) models with m,n,p,q ranging from 1 to 3, we selected the ARMA(1,2)-GARCH(1,3) model, which yielded a Bayesian Information Criterion (BIC) value of -32,521.05. The table below presents the estimation results of the selected model. Based on the p-values, we can confirm that most parameters are statistically significant at the 5% level. The estimation results from the ARMA-GARCH model provide positive indicators regarding its ability to explain the volatility dynamics within the data series.

Table 2. Estimation results of the ARMA-GARCH model

Variable	**Coefficient (coef)**	**P-value (P>|t|)**	**Variable**	**Coefficient (coef)**	**P-value (P>|t|)**
Const	0.0447	0.274	ω	0.0417	3.212e-04
ar.L1	0.9436	0.000	α_1	0.2350	3.914e-24
ma.L1	-0.7000	0.000	β_1	0.5791	4.329e-04
ma.L2	-0.1880	0.000	β_2	4.0666e-10	1.000
σ_2	2.0175	0.000	β_3	0.1809	0.109

Source: Authors' calculations

After training all the models, we proceeded to forecast volatility using the test set. We calculated the log-likelihood values on the test set for each model and compared these log-likelihood values. The comparison results are presented in the table below:

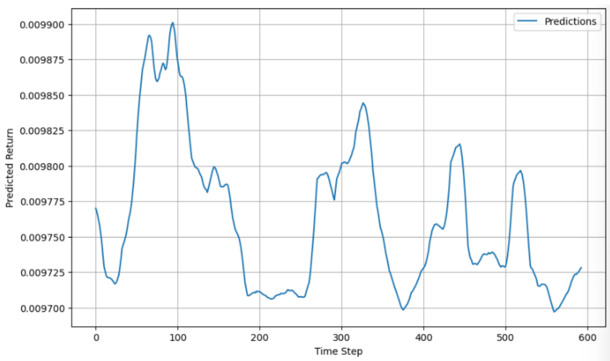
Table 3. Log-Likelihood values of 5 models

Model	**Log likelihood**	**Model**	**Log likelihood**
ARMA-GARCH	1751.72	DNN-Likelihood-based loss	1044151.1
LSTM-MSE	1064726.8	LSTM-Likelihood-based loss	1049298.0
DNN-MSE	1044150.7		

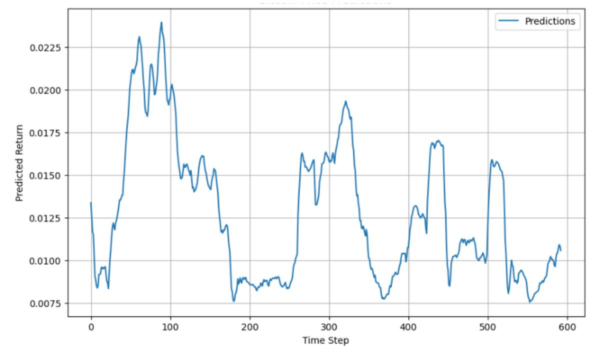
Source: Authors' calculations

The table above provides the log-likelihood values for five models, including ARMA-GARCH, LSTM-MSE, DNN-MSE, DNN-Likelihood-Based Loss, and LSTM-Likelihood-Based Loss. The results indicate that the ARMA-GARCH model has the lowest log-likelihood value (1,751.72), demonstrating the poorest forecasting performance among the models. In contrast, the LSTM-MSE model achieves the highest log-likelihood value (1,064,726.8), indicating it has the best predictive capability in this study. Compared to the likelihood-based optimization method, the LSTM-MSE model outperforms the LSTM-Likelihood-Based Loss model (log-likelihood = 1,049,298.0). For the DNN models, the log-likelihood values of DNN-MSE (1,044,150.7) and DNN-Likelihood-Based Loss (1,044,151.1) are very close, reflecting an insignificant difference between the two optimization methods. Overall, the LSTM-MSE model exhibits the best performance among the models, while the ARMA-GARCH model is the least effective choice.

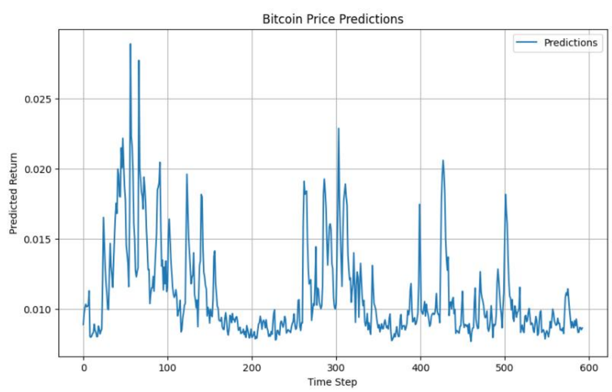
The figure below illustrates the volatility of the VN-Index as predicted by the 5 models on a daily basis. In the long term, the models predict relatively similar trends. However, the model based on the likelihood-based loss function exhibits more detailed and pronounced volatility in the short term.



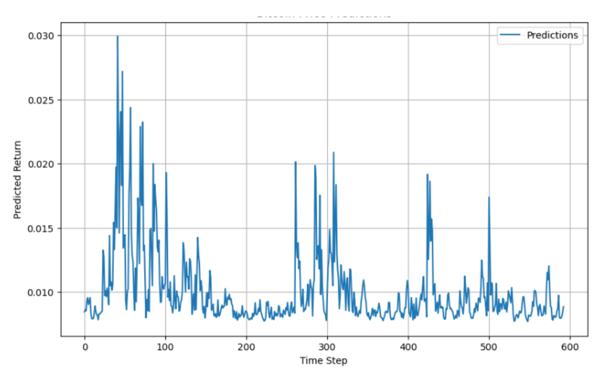
LSTM - MSE



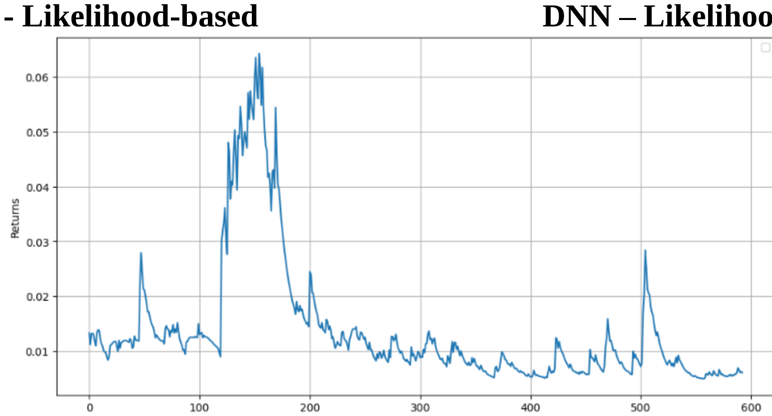
DNN - MSE



LSTM - Likelihood-based



DNN – Likelihood-based



ARMA - GARCH

Figure 8. Prediction results of VNIndex fluctuations using 5 models

Source: Authors' modification

The figure above illustrates that each method exhibits unique characteristics in replicating fluctuations, thereby providing useful information about the effectiveness of each model for the price volatility forecasting problem.

The forecasting results of the models reveal significant differences in forecasting performance. The LSTM-MSE model has the highest fit with the actual data, demonstrating superior capability in capturing both large and small fluctuations as well as long-term trends. In contrast, the DNN-MSE model forecasts are relatively close to the actual data but are less accurate, particularly for small fluctuations. The LSTM-Likelihood-Based Loss model outperforms the DNN and ARMA-GARCH models in forecasting effectiveness but still does not achieve the accuracy of the LSTM-MSE model. The DNN-MSE and DNN-Likelihood-Based Loss models exhibit similar forecasting performance, with limited ability to simulate fluctuations and a tendency to smooth data points, leading to deviations during large fluctuations.

Finally, the ARMA-GARCH model has the lowest forecasting performance, characterized by large deviations and an inability to accurately capture trends and fluctuations in the data. In summary, the LSTM-MSE model is the best-performing model in terms of both log-likelihood values and forecasting accuracy, while the ARMA-GARCH model proves to be the least effective choice.

Overall, the LSTM-MSE model leads in forecasting capability, followed by models utilizing likelihood-based loss functions, while ARMA-GARCH shows limitations in forecasting with modern data. This result emphasizes the importance of selecting a model that is appropriate for the data characteristics and forecasting objectives.

To provide a more detailed illustration of the predicted price volatility, we plotted the line charts of the five models on the same graph. The comparison results are shown in the figure below:

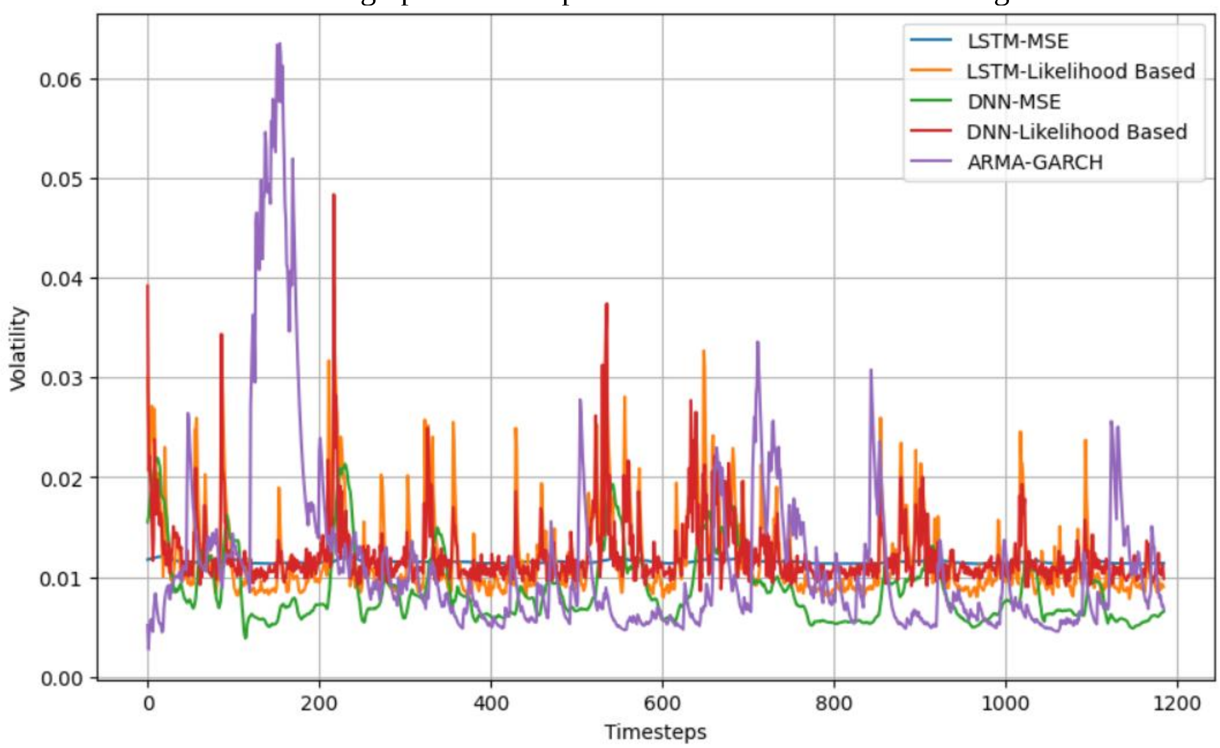


Figure 9. Comparison of VN-Index volatility predictions by 5 models

Source: Authors' modification

The results show that the LSTM-MSE (blue) model provides relatively stable predictions, suitable for data series with smooth and low volatility trends, but does not accurately reflect strong fluctuations. In contrast, the LSTM-Likelihood-Based (orange) model exhibits high sensitivity to large fluctuations in the data, indicating that the likelihood loss function helps the model capture abnormal changes. Similarly, the DNN-MSE (green) model also predicts smoothly and stably, fitting non-volatile trends, while the DNN-Likelihood-Based (red) model improves sensitivity to fluctuations, especially during periods of strong volatility compared to DNN-MSE. The ARMA-GARCH (purple) model shows high sensitivity to strong fluctuations, particularly during sudden changes, but struggles to predict overall trends, resulting in higher deviations during stable periods.

In conclusion, the MSE models (LSTM and DNN) are suitable for smooth and stable data but are less sensitive to sudden changes, whereas the likelihood-based models (LSTM and DNN) significantly improve forecasting capability during periods of high volatility. Although ARMA-GARCH is specialized for financial data, it fails to effectively capture overall trends. Each model has its own strengths, making them suitable for different forecasting objectives and data characteristics.

4.2. Analysis of results

The results in this section indicate that the forecasting capability of deep learning models with a likelihood-based loss function remains stable when employing different data partitioning methods. Across various stages, it can be affirmed that their forecasting performance is not sensitive to different sample period selections. Deep learning models with a likelihood-based loss function consistently forecast volatility better than other models, with the LSTM (likelihood-based loss) consistently being the best method among all the models tested.

In the case where the test set comprises 10% of the data, likelihood-based loss function models, although they do not yield as high log likelihood values as the LSTM-MSE model, exhibit only a slight difference. Additionally, models with likelihood-based loss functions provide short-term predictions that MSE-based models fail to capture. Overall, the results of models utilizing likelihood-based loss functions remain more effective and stable compared to those using the MSE loss function.

The authors also examined scenarios where the test set accounts for 15% and 20% of the data, obtaining the following results:

For the 15% test set, while the ARMA-GARCH model remains the least performant, models with likelihood-based loss functions continue to demonstrate high stability and accuracy. The LSTM-MSE model achieved the highest performance when the test set was 10%, but in this scenario, it exhibited the lowest performance among the deep learning models. This indicates the instability of MSE-based loss function models.

In the case of a 20% test set, among the deep learning models employed, the DNN-MSE model showed the poorest performance, whereas the two models with likelihood-based loss functions maintained their stability and effectiveness.

Based on the results from test set proportions of 10%, 15%, and 20%, it is evident that deep learning models with MSE loss functions are less effective and unstable compared to those with likelihood-based loss functions. The primary reason for this is the inconsistency discussed earlier. However, LSTM and DNN models using MSE loss functions still perform better than the traditional ARMA-GARCH model across all stages and data partitioning methods**.**

5. Conclusion and recommendations

5.1. Conclusion

From the above analysis, we can draw the following conclusions:

(1) Applicability of ARMA-GARCH, DNN, and LSTM Models: The ARMA-GARCH, DNN, and LSTM models are all applicable for analyzing and forecasting the VN-Index of the Vietnamese stock market. Although these models have been widely developed and researched in financial markets of developed countries worldwide, this does not hinder their application to the Vietnamese market. All three models—ARMA-GARCH, DNN, and LSTM—indicate the presence of heteroscedasticity in the volatility of the VN-Index in the Vietnamese stock market.

(2) Superior Potential of Deep Learning Models: The study has affirmed the outstanding potential of deep learning models in the field of financial forecasting, especially for the VN-Index in the Vietnamese stock market. The results show that the LSTM model using a likelihood-based loss function achieved the highest effectiveness, as evidenced by superior log-likelihood values and stability when applied to different datasets. This demonstrates that LSTM can effectively capture the nonlinear characteristics and complex volatility of financial data, far surpassing the limitations of traditional models.

(3) Limitations of Traditional ARMA-GARCH Models: While methods such as ARMA-GARCH remain popular in financial forecasting, the study indicates that they cannot compete with deep learning models in terms of accuracy and flexibility. This is clear evidence of the necessity to apply more modern forecasting tools to meet the increasingly high demands of the market. Furthermore, deep learning models not only provide accurate forecasts but also demonstrate superior stability when applied to different data periods, a particularly important factor in the context of stock markets that frequently fluctuate and are influenced by many unexpected factors.

5.2. Analysis of causes

Overall, the forecasting predictions of the models are unstable, and the ARMA-GARCH model is less effective compared to deep learning models when analyzing the VN-Index in Vietnam due to several limitations.

(1) Linearity of ARMA-GARCH Models: Firstly, ARMA-GARCH is a linear model, suitable for time series with linear relationships and homoscedastic volatility. In contrast, financial data in Vietnam often exhibit high nonlinearity and are influenced by numerous exogenous factors such as economic policies, investor psychology, and market fluctuations.

(2) Parameter Determination Challenges: Secondly, when using the ARMA-GARCH model, users must determine parameters such as the orders of the AR, MA, and GARCH components. This process heavily relies on the user's judgment, making it prone to errors if the analysis is incomplete or if the data are more complex than the model assumes. Traditional models lack the ability to optimize parameters during the training process across the entire dataset, unlike deep learning models, which limits their flexibility and effectiveness in forecasting nonlinear financial time series like the VN-Index.

(3) Limited Memory of ARMA-GARCH Models: Thirdly, ARMA-GARCH primarily focuses on recent historical data and cannot retain or analyze long-term trends or complex temporal relationships. In contrast, deep learning models such as LSTM and DNN excel due to their ability to handle long time sequences. Therefore, in a complex environment like the Vietnamese stock market, deep learning models generally provide much better forecasting results compared to ARMA-GARCH.

(4) Impact of Unusual Volatility and Investor Behavior: Lastly, the VN-Index return series experiences abnormal volatility when shocked by significant "good news" or particularly "bad news," making it difficult to eliminate such impacts in the short term. Additionally, investors in Vietnam have not yet developed a solid investment philosophy, making their investment behavior easily influenced by various types of market news. This explains the reasons behind the fluctuations in the predictions.

5.3. Recommendations

Based on the above analysis, we propose the following policy recommendations:

(1) Advancement of Deep Learning Models in Financial Forecasting: This study not only provides empirical evidence of the effectiveness of deep learning models but also paves the way for developing advanced financial forecasting systems. The application of tools such as LSTM not only enhances forecasting accuracy but also robustly supports investors, businesses, and policymakers in making strategic decisions.

(2) Establishing a Healthy Vietnamese Stock Market: To build a healthy Vietnamese stock market, the government should focus on improving the information disclosure system, enhancing investment philosophy education, and strengthening risk management. Additionally, the Vietnamese government needs to ensure that the market operates according to regulatory principles, avoiding excessive intervention in the Vietnamese stock market.

Funding

This article has been financially supported by Foreign Trade University under research grant number FTURP02-2023-13.

References

[1] D. Banerjee, Forecasting of Indian stock market using time-series ARIMA model, 2nd International Conference on Business and Information Management (ICBIM), Durgapur, India (2014), 131-135, doi: 10.1109/ICBIM.2014.6970973.
[2] Z. Lin, Modelling and forecasting the stock market volatility of SSE Composite Index using GARCH models. Future Generation Computer Systems, 79(Part 3)(2018), 960–972. https://doi.org/10.1016/j.future.2017.08.033.
[3] C. V. Reedy,Predicting the stock market index using stochastic time series ARIMA modelling: The sample of BSE and NSE. SSRN (2019). https://doi.org/10.2139/ssrn.3451677.
[4] T. Mashadihasanli, Stock Market Price Forecasting Using the Arima Model: an Application to Istanbul, Turkiye. İktisat Politikası Araştırmaları Dergisi, 9(2)(2022), 439-454. https://doi.org/10.26650/JEPR1056771.
[5] D. Shah, W. Campbell and F. H. Zulkernine, A comparative study of LSTM and DNN for stock market forecasting. In 2018 IEEE International Conference on Big Data (Big Data), 4148–4155, IEEE (2018). https://doi.org/10.1109/BigData.2018.8622462.
[6] A. M. Rather, LSTM-based deep learning model for stock prediction and predictive optimization model. EURO Journal on Decision Processes, 9 (2021), 100001. https://doi.org/10.1016/j.ejdp.2021.100001.
[7] C. R. Karthik, Raghunandan,B. A. Rao and N. V. S. Reddy, Forecasting variance of NiftyIT index with RNN and DNN. Journal of Physics: Conference Series, 2161 (2022), 012005. https://doi.org/10.1088/1742-6596/2161/1/012005.
[8] K. Alam, M. H. Bhuiyan,I. U. Haque, M. F. Monir and T. Ahmed, Enhancing stock market prediction: A robust LSTM-DNN model analysis on 26 real-life datasets. IEEE Access, 12 (2024), 122757–122768. https://doi.org/10.1109/ACCESS.2024.3434524.
[9] U. M. Sirisha,M. C. Belavagi and G. Attigeri, Profit prediction using ARIMA, SARIMA, and LSTM models in time series forecasting: A comparison. IEEE Access, 10 (2022), 124715–124727. https://doi.org/10.1109/ACCESS.2022.3224938.
[10] V. Goyal and R. R. Raj, Using LSTM, DNN, and ARIMA approaches to predict the trends in the stock market. In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS, 918–922 (2022). IEEE. https://doi.org/10.1109/ICACCS54159.2022.9785150
[11] Nguyen Thanh Huong and Bui Quang Trung, Applying the combined ARIMA-GARCH model to forecast the VN-Index. Journal of Science and Technology - University of Danang (Vietnam), 4(89) (2015), 118–122. https://jst-ud.vn/jst-ud/article/view/2221.
[12] Le Van Tuan and Phung Duy Quang, Applying the GARCH model to forecast the impact of the Covid-19 pandemic on the Vietnamese stock market, Industry and Trade Magazine (Vietnam), 8 (2020), 93-98.
[13] Phung Duy Quang and et al., Estimating and forecasting bitcoin daily prices using ARIMA-GARCH models, Business Analyst Journal, 45 (1) (2024), 1-13. DOI: 10.1108/BAJ-05-2024-0027.
[14] Nguyen Trong Co and et al., Comparison between ARIMA and LSTM-RNN for VN-Index prediction. In T. Ahram, W. Karwowski, A. Vergnano, F. Leali, & R. Taiar (Eds.), Intelligent Human Systems Integration 2020. IHSI 2020. Advances in Intelligent Systems and Computing (Vol. 1131). Springer, Cham. https://doi.org/10.1007/978-3-030-39512-4_168.
[15] H. S. Nguyen, Financial time series forecasting with long short-term memory (LSTM): A comparative experiment between deep learning and econometrics (Master's thesis). Lappeenranta-Lahti University of Technology, School of Business and Management (2024).

[16] Dao Le Kieu Oanh and Nguyen Thi Minh Chau, Forecasting stock indexes using machine learning: Empirical evidence from the Vietnamese stock market. Journal of economics and forecasting (Vietnam), 6 (2024), 1-6.
[17] Truong Cong Doan and et al., Predicting Vietnamese stock market using the variants of LSTM architecture. In P. C. Vinh & N. H. Nhan (Eds.), Nature of computation and communication. ICTCC 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 408 (2021). Springer, Cham. https://doi.org/10.1007/978-3-030-92942-8_11.
[18] L. Zhang and et al., Benefit analysis of china's stock market based on GARCH type models, Nat. Sci. Daily, 17 (2012), 2.
[19] S. Hochreiter and J. Schmidhuber, Long short – term memory, Neural Comptation, 9(8) (1997), 1735-1780.
[20] R. F. Engle. Autoregresive conditinnal heteroscedasticity with estimates of the variance of United Kingdom inflation", Econometrica, 50(4) (1982), 987-1007.
[21]T . Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of Econometríc, 31 (3) (1986), 307-327.
[22] L. Kalyanaraman, Stock market volatility in Saudi Arabia: An application of univariate GARCH model. Asian Social Science, 10(10) (2014), 142-156. https://doi.org/10.5539/ass.v10n10p142