Predicting stock prices of major companies using machine learning models

1st Pham Hoang Hiep
Faculty of Information Technology
UET-VNU
Hanoi, Vietnam
22028005@vnu.edu.vn

2nd Duong Minh Hoang
Faculty of Information Technology
UET-VNU
Hanoi, Vietnam
22028186@vnu.edu.vn

***Abstract*—Stock price movements have always been a critical topic in financial economics, reflecting both the performance of firms and the broader market sentiment. In the context of growing investor interest and market complexity, there is an increasing need for quantitative approaches to predict stock prices with more accuracy. This study aims to develop a predictive model for stock prices using key financial indicators and macroeconomic variables. The goal is to provide a structured model-based framework that enhances understanding and forecasting of stock price behavior, offering practical implications for investors, analysts, and policymakers.**

***Index Terms*—Stock Market Forecasting, Financial Modeling, Machine Learning**

I. Introduction

In modern economies, the stock market plays a critical role as both a barometer of economic performance and a channel for capital allocation. Stocks represent ownership in companies and reflect investor expectations about future earnings, growth, and risk. As such, stock prices are influenced not only by the internal fundamentals of a company but also by a wide range of macroeconomic and psychological factors. The ability to accurately predict stock prices has long been a central concern in financial economics, as it supports investment strategies, portfolio management, and market efficiency. Forecasting stock prices is especially important in volatile and rapidly evolving markets, where informed decision-making can yield significant financial benefits and help mitigate risks for investors, institutions, and policy makers alike.

Currently, although there are numerous stock evaluations and analyses available, most of them are qualitative in nature, focusing on market sentiment, industry trends, or speculative information. However, the number of quantitative analyses, particularly those based on financial data and specific macroeconomic factors, remains relatively limited. This creates a significant gap in providing more accurate and reliable assessments of a stock's true value, especially when considering long-term factors and unforeseen external variables.

Building such a model presents several challenges, particularly when it comes to accurately identifying the factors that truly impact stock prices. One of the main difficulties is separating the noise created by market psychology and sentiment from more fundamental financial metrics such as revenue, costs, and profitability. Although financial data plays a crucial role, psychological factors, including investor emotions and market rumors, often introduce significant volatility and can distort model predictions. In addition, determining the right weight for each factor and understanding how they interact over time requires sophisticated modeling techniques and a deep understanding of the market. Balancing these complex elements while minimizing errors and overfitting remains a key hurdle in building a reliable predictive model for stock prices.

In this paper, we will evaluate the factors that genuinely influence stock prices, distinguishing between fundamental financial elements and external variables that can introduce noise. By carefully analyzing historical financial reports, market trends, and macroeconomic indicators, our objective is to identify the key drivers of stock price movements. Using these identified factors, we will construct a predictive model that predicts future stock prices, focusing on providing a more accurate and reliable approach. The goal is to leverage the most relevant and impactful data, minimizing the effects of psychological fluctuations and enabling better-informed investment decisions for the near future.

II. Related work

In previous research, numerous studies have explored stock price prediction, identifying several factors that can influence stock prices. These factors include the price-to-earnings (P/E) ratio and the debt-to-equity (D/E) ratio (Nirmala et al., 2011 [1]), earnings per share (EPS) (Islam et al., 2014 [2]), and company size (SIZE) (Shariff et al., 2015 [3]). EPS has been found to have a positive impact on stock prices, while the D/E ratio shows a negative effect. Company size and the P/E ratio also tend to play significant roles, with larger companies and those with higher P/E ratios generally performing better in the stock market. In addition, the inflation rate can affect stock prices, as macroeconomic conditions influence both investor expectations and corporate performance. These findings highlight the complex interplay of various financial and economic factors in determining stock price movements.

III. Data Collection

Data plays a crucial role in training the prediction model. In this study, we used Python scripts to collect consolidated

financial statements and quarterly income statements from CafeF [4], and retrieved daily stock price data from Yahoo Finance [5] using the `yfinance` library. The data was gathered from more than 400 Vietnamese companies from 2005 to 2024. Among them, the data from five key corporations—**FPT**, **MB Bank**, **Hoa Phat Group**, **Hoang Anh Gia Lai**, and **Vinamilk**—were of particular importance to our analysis.

During the data collection process, some inconsistencies and missing values were identified. These were cross-checked and corrected by comparing them with the original financial reports published by the respective companies. In addition, a sample of the collected data was manually verified to ensure its accuracy and reliability before proceeding with further analysis and model training.

IV. First Approach

A. Motivation

As an initial step in building a stock price prediction system, I chose to experiment with the ARIMA (Autoregressive Integrated Moving Average) model. ARIMA is a classical and well-established method in time-series analysis, widely used in financial forecasting due to its effectiveness in modeling data that exhibit trends and stochastic components.

The motivation behind using ARIMA as the first model stems from its simplicity, interpretability, and ability to serve as a strong baseline. It does not rely on external variables or complex features; instead, it captures patterns within the time series itself, such as past values (autoregressive terms) and past forecast errors (moving average terms), to make predictions.

In this section, I train the ARIMA model on the log returns of stock prices, which is a standard approach to ensure the stationarity assumption required by the model. The parameters (p, d, q) are selected based on the Akaike Information Criterion (AIC), aiming to find the most parsimonious yet accurate model. The trained ARIMA model is then used to forecast future values, and its predictions are evaluated through visualization and basic performance comparison against actual data.

Although ARIMA lacks the capability to incorporate exogenous variables or market-related factors, it provides a crucial reference point. Its performance indicates whether the historical price series alone contains sufficient predictive power. This insight lays the groundwork for later stages, where more advanced models—potentially incorporating financial indicators, news sentiment, or macroeconomic variables—can be introduced and compared.

B. ARIMA Model

The **ARIMA (Autoregressive Integrated Moving Average)** model is a classical approach for modeling and forecasting time series data that exhibit temporal dependencies. It combines three key components: autoregression (AR), differencing (I), and moving average (MA). The general form of the ARIMA model is denoted as ARIMA(p, d, q), where:

- p is the order of the autoregressive (AR) component,
- d is the degree of differencing required to make the time series stationary,
- q is the order of the moving average (MA) component.

1) Autoregressive (AR) Process: : The autoregressive component of the ARIMA model captures the relationship between the current observation and a specified number of lagged observations. The general form of the AR process is:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t \tag{1}$$

where: - Y_t is the value at time t, - $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters of the autoregressive model, - ϵ_t is the error term at time t, which is assumed to be white noise.

2) Differencing (I) Process: To make the time series stationary, differencing is applied. The degree of differencing, d, determines how many times the series must be differenced. The differenced series is given by:

$$Y_t^{(d)} = Y_t - Y_{t-1} \tag{2}$$

If the series is still non-stationary after first differencing, second differencing may be applied, and so on. The goal is to remove trends and make the series stationary.

3) Moving Average (MA) Process: The moving average component models the relationship between the current value and past error terms. The general form of the MA process is:

$$Y_t = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{3}$$

where: - μ is the mean of the series, - ϵ_t is the white noise error term at time t, - $\theta_1, \theta_2, \ldots, \theta_q$ are the parameters of the moving average model.

4) ARIMA Model Equation: Combining the AR, I, and MA components, the general ARIMA(p, d, q) model equation is given by:

$$\Delta^d Y_t = \phi_1 \Delta^d Y_{t-1} + \phi_2 \Delta^d Y_{t-2} + \cdots + \phi_p \Delta^d Y_{t-p} + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{4}$$

where $\Delta^d Y_t$ represents the d-th differenced series, and the terms ϕ_i and θ_j are the parameters of the model that need to be estimated.

In practice, ARIMA models are identified and estimated by analyzing the autocorrelation (ACF) and partial autocorrelation (PACF) plots, which help determine the values of p, d, and q that best fit the data.

C. Data Preprocessing and Visualization

For each stock, the first step in data preprocessing is to calculate the log return, defined as the logarithmic difference between the stock price on day t and day $t-1$:

$$\text{Log Return} = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

This transformation generates a time series of returns, which are used for prediction instead of the actual stock prices. This approach is necessary because ARIMA requires the time series to be stationary, and the raw stock prices generally exhibit trends or volatility that violate this assumption.

After calculating the log returns, various visualizations are produced to gain a deeper understanding of the data:

- **Return Rate vs. Date Order**: This plot shows the return rate over time, providing a clear view of the fluctuations in stock price changes across the selected period (see Figure 1).
- **Return Rate with Lag Order 1**: This plot displays the return rate with a one-period lag to identify any immediate autocorrelation patterns in the data.
- **Distribution Plot**: A histogram or density plot is drawn to observe the distribution of the return rates, helping to assess whether the returns follow a normal distribution or exhibit skewness, which can be important for model assumptions.

Next, the stationarity of the log return series is tested using the **Augmented Dickey-Fuller (ADF) test**. This test is crucial because ARIMA models assume that the time series is stationary, meaning its statistical properties do not change over time. If the series fails the stationarity test, transformations such as differencing may be applied.

Finally, **Autocorrelation Function (ACF)** and **Partial Autocorrelation Function (PACF)** plots are generated to determine the optimal values for the ARIMA parameters p and q. The ACF plot helps identify the moving average (MA) order q, while the PACF plot is used to determine the autoregressive (AR) order p. These plots are essential in selecting the appropriate model configuration for accurate forecasting.

D. Training and Predicting with ARIMA Model

After preprocessing the data, the ARIMA model is trained using the historical time series data. The processed data, which consists of the log returns of stock prices, is used to fit the model. Once trained, the ARIMA model is used to predict future values of the stock price by leveraging the patterns identified during the training phase.

For forecasting, the model predicts future values based on past observations. The process of making long-term predictions is done iteratively by using the predicted value at each step as input for the next prediction. Specifically, the predicted value at time $t+h$ is used as an input for predicting $t+h+1$, and so on.

E. Evaluation and Limitations

While the ARIMA model provides a useful baseline for forecasting stock returns using only historical price data, its predictive performance over extended horizons reveals certain limitations. As the model continues to generate future values based solely on previous predictions, without incorporating real-world feedback or new market information, the predicted return sequence tends to converge toward zero, as shown in Figure 2.

This convergence is a result of the ARIMA model's reliance entirely on the internal dynamics of the time series. It does not account for external factors such as market sentiment, macroeconomic indicators, or company-specific events that significantly influence stock prices in reality. As a result, the model's forecasts may become increasingly disconnected from actual market movements over time.

This observation highlights a key insight: while the internal structure of a stock's historical data may carry some predictive power, it is not sufficient on its own to model the complexity of financial markets. To improve prediction accuracy and realism, it becomes necessary to adopt more advanced models that integrate additional explanatory variables — such as financial ratios, industry trends, and news sentiment — thereby capturing the external forces that drive stock price behavior.

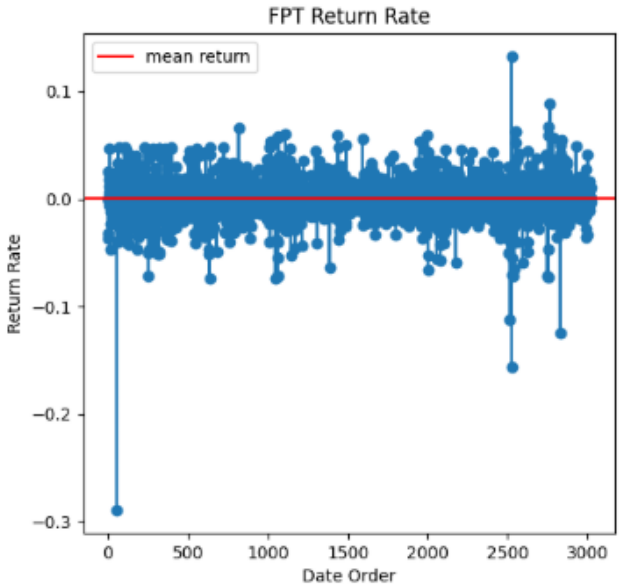

Fig. 1. Log return rate of FPT stock over time.

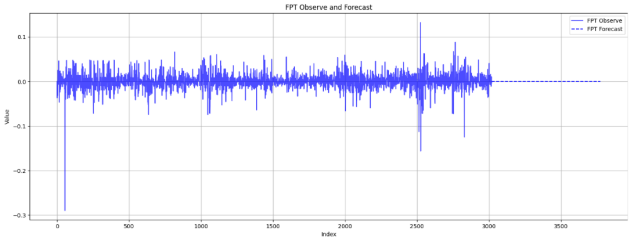

Fig. 2. Observed vs. forecasted log returns of FPT stock using the ARIMA model.

V. Factor Evaluation

In this section, we will evaluate the factors that may influence stock prices, including P/E ratio, D/E ratio, EPS, company size (SIZE), and inflation (INF).

A. Evaluation Model

Based on previous studies, we observe that the stock price at time $t+1$ can be influenced by several financial factors at time t, including the price-to-earnings ratio (P/E), earnings per share (EPS), debt-to-equity ratio (D/E), and company size ($SIZE$).

Therefore, we construct a linear regression model to evaluate the relationship between these factors and future stock prices. The model is specified as follows:

$$P_{i,t+1} = \beta_0 + \beta_1 \cdot (P/E)_{i,t} + \beta_2 \cdot EPS_{i,t} + \beta_3 \cdot (D/E)_{i,t} + \beta_4 \cdot SIZE_{i,t} + \varepsilon_{i,t}$$

Where:

- $P_{i,t+1}$: stock price of company i at quarter $t+1$,
- $(P/E)_{i,t}$, $EPS_{i,t}$, $(D/E)_{i,t}$, $SIZE_{i,t}$, $P_{i,t}$: financial indicators at time t,
- $\beta_0, \ldots, \beta_5$: regression coefficients,
- $\varepsilon_{i,t}$: error term.

B. Dataset Description

The dataset used for evaluating financial factors consists of financial and stock price data collected from more than 400 Vietnamese companies. From the raw data, relevant financial indicators were extracted and computed, resulting in a structured dataset with over 5,300 rows. Each row corresponds to a company-quarter observation and includes variables such as the stock price at time t (P_t), the stock price at time $t+1$ (P_{t+1}), earnings per share (EPS), debt-to-equity ratio (D/E), price-to-earnings ratio (P/E), and firm size (SIZE). These variables serve as the foundation for the subsequent factor evaluation and modeling process.

C. Analytical Methodology

To quantify the impact of financial factors on stock prices, we employ a multivariate regression analysis framework. Specifically, we aim to evaluate how variables such as price-to-earnings ratio (P/E), earnings per share (EPS), debt-to-equity ratio (D/E), firm size (SIZE), and stock price at time t (P_t) influence the subsequent stock price at time $t+1$ (P_{t+1}).

Given the panel nature of our dataset—comprising multiple companies observed across different quarters—we utilize three commonly used econometric techniques for panel data analysis: the Pooled Ordinary Least Squares (Pooled OLS) regression, the Fixed Effects Model (FEM), and the Random Effects Model (REM). These models allow us to control for unobserved heterogeneity and to robustly assess the relationship between financial indicators and stock price movements over time.

To select the most appropriate model, we perform a series of statistical tests. First, the F-test is conducted to compare the Pooled OLS model with the Fixed Effects Model (FEM), determining whether unobserved heterogeneity across firms is significant. Next, the Hausman test is employed to decide between FEM and the Random Effects Model (REM) by testing whether individual effects are correlated with the explanatory variables. In addition, diagnostic checks are performed to detect the presence of multicollinearity, autocorrelation, and heteroskedasticity in the regression residuals. These tests ensure the robustness and reliability of the estimated coefficients in the regression models.

D. Results

The Variance Inflation Factors (VIFs) for the independent variables in our model were calculated, as shown in Figure 3. The results indicate that all VIF values are well below the common threshold of 5, suggesting that multicollinearity is not a significant concern in our regression models. Therefore, the coefficient estimates are considered reliable.

	Variable	VIF
0	const	1.000000
1	P/E	1.003353
2	EPS	1.003845
3	D/E	1.020372
4	SIZE	1.026921

Fig. 3. VIF values for each independent variable.

The Durbin-Watson statistics for the three models are as follows:

	Model	Durbin-Watson Statistic
0	Pooled OLS	1.539020
1	Fixed Effects (FEM)	0.345235
2	Random Effects (REM)	0.334447

Fig. 4. Durbin-Watson Test Results for Autocorrelation in Models

These results suggest that autocorrelation is a concern in the FEM and REM models, potentially affecting the reliability of the estimated coefficients, while the Pooled OLS model shows weaker autocorrelation.

The Breusch-Pagan (BP) test results for three models are as follow:

	Model	LM Statistic	LM P-value	F-statistic	F-statistic P-value
0	Pooled OLS	34.069789	7.210295e-07	8.564449	6.929362e-07
1	Fixed Effects (FEM)	36.754170	2.024053e-07	9.243958	1.930165e-07
2	Random Effects (REM)	39.988346	4.352510e-08	10.063559	4.108263e-08

Fig. 5. Breusch-Pagan Test Results for Heteroscedasticity in Models.

The Breusch-Pagan (BP) test results indicate significant heteroscedasticity in the residuals of all models (Pooled OLS,

FEM, REM), as the p-values for both the LM and F-statistics are all below 0.01. This suggests that robust standard errors should be considered to account for heteroscedasticity.

The **F-statistic** for the **Fixed Effects (FEM)** model is 58.64, with a **p-value** less than 1%, suggesting that FEM is a suitable choice for the data.

However, the **Hausman test** results show a statistic of 1.41 with a **p-value** of 9%, indicating no significant difference between **FEM** and **Random Effects (REM)**. This suggests that **REM** could also be a viable alternative.

It is important to note that both models may suffer from **autocorrelation** and **heteroskedasticity**, which should be addressed, possibly using robust estimation techniques or **GMM**.

The estimation results from the Fixed Effects Model (FEM) are presented in Figure 6. Specifically, the P/E ratio has an estimated coefficient of -0.0027 with a p-value of 0.6535, indicating no statistically significant relationship between P/E and the dependent variable. The EPS variable has an estimated coefficient of 0.0251 with a p-value of 0.0074, indicating a positive and statistically significant relationship with the dependent variable. The D/E ratio has an estimated coefficient of -0.0053 with a p-value of 0.0434, showing a negative and statistically significant relationship with the dependent variable. Finally, the SIZE variable has the largest estimated coefficient of 0.3420 with a p-value of 0.0000, indicating a strong and statistically significant impact on the dependent variable. These results reflect the varying effects of financial factors and company size on the stock data in the model.

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	-9.952e-16	0.0065	-1.53e-13	1.0000	-0.0128	0.0128
P/E	-0.0027	0.0061	-0.4490	0.6535	-0.0146	0.0092
EPS	0.0251	0.0094	2.6781	0.0074	0.0067	0.0435
D/E	-0.0053	0.0026	-2.0204	0.0434	-0.0105	-0.0002
SIZE	0.3420	0.0237	14.425	0.0000	0.2955	0.3885

Fig. 6. FEM Parameter Estimates

The results from the Random Effects Model (REM) are similar to those from the FEM, as shown in Figure 7:

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	0.0135	0.0642	0.2097	0.8339	-0.1125	0.1394
P/E	-0.0023	0.0057	-0.3993	0.6897	-0.0135	0.0089
EPS	0.0264	0.0097	2.7090	0.0068	0.0073	0.0454
D/E	-0.0059	0.0027	-2.1572	0.0310	-0.0112	-0.0005
SIZE	0.3280	0.0221	14.833	0.0000	0.2847	0.3714

Fig. 7. REM Parameter Estimates

E. Conclution

In conclusion, the results indicate that EPS, D/E, and SIZE have statistically significant effects on the dependent variable, while P/E does not appear to have a significant impact.

The positive relationship between EPS and the dependent variable suggests that higher earnings per share are associated with a higher stock value. This is because EPS is a key indicator of a company's profitability, and higher earnings per share signal a more efficient and profitable company, which in turn attracts investors and drives up stock prices.

The negative relationship between D/E and the dependent variable indicates that companies with lower debt relative to equity tend to have higher stock values. This is because a lower debt-to-equity ratio generally reflects better financial health, reduced risk of bankruptcy, and increased investor confidence, which can lead to higher stock prices.

Finally, the SIZE variable shows a strong positive impact on stock value, suggesting that larger companies tend to have higher stock prices. Larger companies often benefit from economies of scale, greater market recognition, and more stable cash flows, making them more attractive to investors and thus resulting in higher stock values.

Based on these findings, we will use these factors to train a stock price prediction model, leveraging their influence to improve the accuracy of the model's forecasts.

VI. LSTM-BASED APPROACH

In this section, we propose using a Long-Short-Term Memory (LSTM) model to predict stock prices, incorporating the financial factors identified in Section 5. The LSTM model will take advantage of additional factors such as EPS, D/E, and SIZE to improve prediction accuracy.

A. Data Preprocessing

Our goal is to predict future stock prices based on factors observed previously. To achieve this, we first apply feature normalization using the `StandardScaler`, which standardizes each feature by removing the mean and scaling to the unit variance.

After normalization, we transform the time-series data into supervised learning samples. Specifically, for each sample, we use the observed values of all relevant factors from the previous n_steps_in time steps as input, and the stock prices from the following n_steps_out time steps as the corresponding output. This sliding-window approach allows the model to learn temporal dependencies and capture patterns in the sequential data.

The resulting input-output pairs are then fed into the LSTM model, which is designed to learn from sequences and make accurate multistep-ahead predictions.

B. Model Selection

To determine the most effective architecture for stock price prediction, we experimented with several variants of the long-short-term memory (LSTM) model. These included the standard stacked LSTM, bidirectional LSTM, CNN-LSTM, as well as hybrid models that combine Bidirectional and CNN layers.

In the stacked LSTM architecture, multiple LSTM layers are placed on top of each other to increase the model's capacity to learn complex temporal dependencies. The bidirectional LSTM model processes the input sequence in both forward and backward directions, allowing the network to capture patterns that depend on both past and future context within the input window.

We also explored the CNN-LSTM architecture, where convolutional layers are applied before LSTM layers. These convolutional layers are responsible for extracting local temporal patterns and reducing the input dimensionality, which can enhance the performance of subsequent LSTM layers.

Furthermore, we developed hybrid architectures that combine both CNN and Bidirectional LSTM layers. In these models, the CNN layers first extract relevant features, and the resulting feature maps are then processed by Bidirectional LSTM layers to capture comprehensive temporal dependencies. This combination aims to leverage the strengths of both architectures—local pattern extraction from CNNs and contextual sequence modeling from Bidirectional LSTMs.

In all architectures, we experimented with combinations of LSTM, Convolutional, Dense, and Dropout layers to optimize performance and mitigate overfitting. The final selection of the model was based on the loss of validation and the accuracy of the forecast.

C. Experimental Setup

To evaluate the performance of the proposed models, we divided the dataset into training and testing sets using a 70/30 split. Prior to splitting, the data was shuffled to ensure a more robust and generalized training process by reducing any time-dependent biases that may exist in the sequence.

During model construction, we experimented with different activation functions. The primary activation functions used were `ReLU` (Rectified Linear Unit) and `GELU` (Gaussian Error Linear Unit), both of which are known for their ability to handle non-linearity and enhance deep learning performance.

For training, we employed two loss functions: Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). MSE emphasizes larger errors and is commonly used in regression tasks, while MAPE provides a more interpretable percentage-based error metric, which is particularly useful when evaluating forecasting accuracy.

In addition to architectural variations, we also fine-tuned several hyperparameters to optimize model performance. These included `n_steps_in`, which defines the number of past time steps used as input; `n_steps_out`, which specifies the number of future time steps the model is required to predict. Careful tuning of these parameters helped improve the model's ability to generalize and make accurate multi-step forecasts.

All experiments were conducted under the same preprocessing pipeline and architectural tuning strategies described in earlier sections to ensure consistency in model comparisons.

D. Results

Overall, our experiments indicate that the choice of activation function does not significantly impact the final performance of the models. Across various settings, the differences in both MSE and MAPE between activation functions such as `ReLU`, `GELU` and `linear` were relatively small, as shown in Figure 8 and Figure 9. This suggests that the model's predictive capability is more strongly influenced by other architectural or training factors rather than the activation function itself.

However, we observed a consistent trend where increasing the values of `n_steps_in` and `n_steps_out` generally led to higher loss values. This can be attributed to the increased complexity of the forecasting task. As `n_steps_in` grows, the input sequences become longer, potentially introducing more noise and making it harder for the model to learn meaningful temporal patterns. Similarly, as `n_steps_out` increases, the model is required to make longer-term predictions, which are inherently more uncertain and thus more prone to error. These observations, supported by Figures 8 and 9, highlight the trade-off between prediction horizon and accuracy in time series forecasting tasks.

	model	relu	gelu	linear
0	bidirectional_cnn_lstm	0.107466	0.115837	0.195768
1	bidirectional_lstm	0.106912	0.112567	0.122175
2	bidirectional_stacked_lstm	0.096889	0.106698	0.114320
3	cnn_lstm	0.109664	0.167955	0.150103
4	double_stacked_lstm	0.105925	0.106555	0.110962
5	triple_stacked_lstm	0.097214	0.116044	0.119007

Fig. 8. Average MSE values for each model using different activation functions

	model	relu	gelu	linear
0	bidirectional_cnn_lstm	70.459150	70.828076	65.439762
1	bidirectional_lstm	62.832488	62.707595	63.505055
2	bidirectional_stacked_lstm	60.749939	58.391257	58.296277
3	cnn_lstm	70.153949	70.784424	67.852990
4	double_stacked_lstm	63.824276	59.945210	57.947841
5	triple_stacked_lstm	58.128208	58.362542	58.332726

Fig. 9. Average MAPE values for each model using different activation functions

E. Final Model

Based on our observations from the experimental results, we selected the `Triple Stacked LSTM` architecture with

`ReLU` activation and the `MSE` loss function for the final stock price prediction model. This configuration used `n_steps_in` set to 5 and `n_steps_out` set to 3.

The choice of the triple-stacked structure is motivated by its ability to capture complex temporal dependencies through a deeper network. Each additional LSTM layer allows the model to progressively abstract more meaningful temporal patterns from raw input data. The stacking helps the network learn hierarchical time-based features, which is especially beneficial in financial time series where patterns may exist at multiple time scales.

The values for `n_steps_in` and `n_steps_out` were selected after evaluating the trade-off between prediction accuracy and forecasting range. As illustrated in Figure 10, we observed that increasing `n_steps_in` generally led to a decrease in the final loss, as the model could leverage a longer temporal context to better capture recent patterns. On the other hand, increasing `n_steps_out` resulted in higher loss values, likely due to the added complexity of making longer-term predictions, which are more prone to uncertainty.

By choosing `n_steps_in` = 5 and `n_steps_out` = 3, we strike a balance between ensuring sufficient temporal context and limiting the prediction horizon to avoid excessive error accumulation.

This configuration provides a balanced approach, ensuring that the model is deep enough to extract meaningful signals while remaining responsive and accurate over a short prediction horizon—qualities that are essential for real-world stock price forecasting tasks.

F. Using the Final Model for Forecasting

After selecting the best-performing architecture based on evaluation results, we applied the trained model to forecast stock prices for the upcoming three years: 2025, 2026, and 2027. We use this final model to generate predictions for five representative corporations: **FPT**, **MB Bank**, **Hoa Phat Group**, **Hoang Anh Gia Lai**, and **Vinamilk**

The purpose of this forecasting step is to illustrate the model's ability to generalize beyond the training period and provide insights into future stock price trends. While no model can fully capture the uncertainty of financial markets, these forecasts offer a data-driven perspective on potential stock movements assuming current patterns persist.

The predicted stock prices for each company over the three-year horizon are visualized in Figure 11.

VII. Limitations and Future Work

Despite the promising results achieved in this study, there are several limitations that need to be addressed in future research. First, the models used in this paper are relatively simple, with the LSTM architecture being relatively straightforward and lacking higher levels of complexity. While this simplicity allows for efficient training and testing, more sophisticated models could potentially improve the predictive performance by capturing deeper patterns in the data.

	n_steps_in	n_steps_out	final_loss_relu_mse
0	1	1	0.086093
1	1	2	0.073564
2	1	3	0.101923
3	1	4	0.101353
4	1	5	0.132203
5	2	1	0.127618
6	2	2	0.115108
7	2	3	0.100739
8	2	4	0.087739
9	2	5	0.169487
10	3	1	0.055462
11	3	2	0.074585
12	3	3	0.096009
13	3	4	0.112805
14	3	5	0.099406
15	4	1	0.069798
16	4	2	0.098389
17	4	3	0.089082
18	4	4	0.116734
19	4	5	0.136539
20	5	1	0.063438
21	5	2	0.065513
22	5	3	0.087274
23	5	4	0.079281
24	5	5	0.090214

Fig. 10. Triple stacked LSTM model performance for different (n_steps_in, n_steps_out) combinations

	Company	2025_Predict	2026_Predict	2027_Predict
0	FPT	52788.222656	49155.734375	45951.867188
1	HAG	7814.091797	6582.179199	5849.959473
2	HPG	28994.101562	24349.318359	21633.583984
3	VNM	87652.828125	74076.859375	71850.343750
4	MBB	13179.325195	11102.770508	9820.454102

Fig. 11. Predicted stock prices of five corporations from 2025 to 2027

Another limitation is the dataset used in this study. The sample size is relatively small, which may limit the generalizability of the model. A larger and more diverse dataset, including more extensive historical stock prices, would provide a more robust foundation for training and could help improve the accuracy of predictions. Additionally, the model has not incorporated a wide range of economic factors, such as macroeconomic indicators, market sentiment, or geopolitical events, which are known to have a significant impact on stock prices. Integrating these factors into the model could provide a more holistic understanding of stock price movements.

Looking forward, there are several directions for future work. First, we plan to explore more complex architectures, such as deeper LSTM networks, hybrid models combining LSTM with other neural network types (e.g., convolutional neural networks or attention mechanisms), and even transformer-based models, which have shown great promise in sequential data tasks. Additionally, we intend to experiment with other machine learning models, such as Random Forests, Gradient Boosting, and XGBoost, to compare their performance and determine which models best capture the underlying dynamics of stock price movements.

Furthermore, future research will incorporate additional economic variables, such as inflation rates, interest rates, and employment data, to provide a more comprehensive feature set for the prediction models. Finally, we aim to explore the use of Large Language Models (LLMs) to assess market sentiment and investor psychology, which can significantly influence stock prices. By integrating these external factors and utilizing more advanced models, we hope to further improve the accuracy and robustness of stock price predictions.

VIII. Acknowledgments

The code and datasets used in this study are publicly available on GitHub at
https://github.com/hype1524/PredictStockPrice.

References

[1] P. S. Nirmala, P. S. Sanju, M. Ramachandran, Determinants of share prices in india, Journal of Emerging Trends in Economics and Management Sciences 2 (2011) 124–130.

[2] M. Islam, T. Khan, T. Choudhury, A. Adnan, S. Lecturer, How earning per share (eps) affects on share price and firm value, European Journal of Business and Management 6 (01 2014).

[3] T. Sharif, H. Purohit, R. Pillai, Analysis of factors affecting share prices: The case of bahrain stock exchange, International Journal of Economics and Finance 7 (2015) 207–207. doi:10.5539/ijef.v7n3p207.

[4] CafeF, Vietnam finance and stock market information, accessed: 2025-05-07 (2024).
URL https://s.cafef.vn

[5] Yahoo Finance, Stock market live, quotes, business & finance news, accessed: 2025-05-07 (2024).
URL https://finance.yahoo.com