A Comparative Analysis of Machine Learning and Deep Learning Models for Real Estate Stock Market Forecasting

PHAN CHI CUONG[1], NGUYEN LE KHANG[2], AND LE PHAM QUOC BAO[3]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21520673@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21520960@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21521849@gm.uit.edu.vn)

ABSTRACT This study shows a detailed comparative analysis of statistical models, machine learning, and deep learning models for predicting real estate stock prices. We check the predictive capabilities of different models like the Autoregressive Integrated Moving Average with Exogenous variables (ARIMAX), Linear Regression, Long Short-Term Memory (LSTM) networks, Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), Random Forest (RF), Fast Fourier Transform (FFT), and TimesNet. Using a carefully prepared dataset of historical stock prices and technical indicators, we train and test these models with MAPE, RMSE and Mean Directional Accuracy (MDA). Our results show that deep learning models, like LSTM and GRU, perform better than traditional time series and machine learning models in catching the complex behaviors of real estate stock markets. Also, adding technical indicators as exogenous variables in ARIMAX, using frequency domain analysis with FFT, and applying Linear Regression all improve forecasting accuracy. This research gives useful insights for investors and financial analysts aiming to make informed decisions in the real estate stock market.

INDEX TERMS Placeholder

I. INTRODUCTION

Time-series forecasting is very important in decision-making across many fields. Its value comes from its ability to provide insights about future trends and patterns in time-dependent data. For example, accurate forecasting of stock prices, foreign exchange rates, and interest rates is crucial for making good financial investments. In healthcare, organizations depend on predictions of patient demand and resource use to optimize resources and improve patient care. Energy companies also use time-series forecasting to optimize energy production, distribution, and consumption. The accuracy and efficiency of time-series forecasting models greatly affect organizational performance and decision-making.

In this paper, we explore a new approach to make time-series forecasting better using the Fast Fourier Transform (FFT). The FFT algorithm extracts frequency-domain features from time-series data, offering a promising way to improve forecast accuracy and computational efficiency. Our investigation include a comparative analysis of models trained with FFT-based features against traditional time-domain features. We apply this method to predict stock prices of real estate companies, using not only FFT but also other techniques like TimesNet and Random Forest. Through our study, we highlight the interpretability of frequency-domain features and their relationship with underlying time-series patterns, emphasizing the potential of FFT-based feature engineering to improve forecasting models.

II. RELATED WORKS

In recent years, many stock prediction models have been researched and many articles have been published, such as:

Hind Daori, Alanoud Alanazi, Manar Alharthi, Ghaida Alzahrani (2022) [**?**] used Artificial Neural Network (ANN), Random Forest Classifier, Logistic Regression,and then analyze and predict the patterns of previous stock prices and the results showed that the models were efficient and produced better results.

Hugo Souto(2023) [**?**] has researched about TimesNet for Realized Volatility Prediction. Finally, they concluded that TimesNet stands out as a reliable and effective benchmark model for researching realized volatility. Although it may not always surpass NBEATSx and NHITS in every metric, its strong performance and consistency make it a

valuable option, especially when compared to TFT. Overall, TimesNet presents a balanced and dependable choice that combines reliability with effectiveness, making it a suitable neural network model for researchers and practitioners in the field of realized volatility.

In another article by Bohumil Stádník, Jurgita Raudeliuniene, Vida Davidavičienė [**?**], they pointed out that the Fourier analysis may not be advantageous for investors forecasting stock market prices as it fails to detect existing predominant cycles. An attempt to identify significant periods in the US stock market data using FFT, a method of Fourier analysis, proved to be unacceptable. Similar failures can be expected with other liquid investment instruments or financial data series. Despite this, Fourier analysis is still used for forecasting in finance and its benefits are a topic of discussion among financial market practitioners and academicians.

III. MATERIALS

A. DATASET

The dataset comprises historical daily closing stock prices (in Vietnamese Dong - VND) for three prominent Vietnamese real estate companies:

- Quoc Cuong Gia Lai Joint Stock Company (QCG)
- Dat Xanh Group Joint Stock Company (DXG)
- Vinhomes Joint Stock Company (VHM)

The data spans a five-year period from March 1, 2019, to March 1, 2024. While the raw data includes additional attributes such as opening price, high, low, volume, and change, this study focuses solely on the "Close" price to develop predictive models for future closing price movements.

B. DESCRIPTIVE STATISTICS

TABLE 1. QCG, VHM, DXG's Descriptive Statistics

	DXG	VHM	QCG
Observations	1252	0	1252
Mean	17676	62065.92	7586.17
Median	15348	61768	7105
Std	7862	11877.69	3102.55
Min	6739	38450	3320
Max	46750	88722	23200
25%	12303	53900	4960
50%	15348	61768	7105
75%	20806	71569	9182.5
Skewness	1.41	-0.04	1.13
Kurtosis	1.97	-0.85	1.68

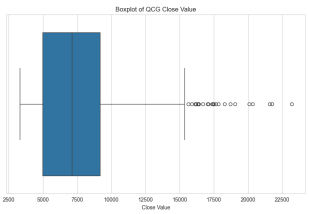

FIGURE 1. QCG stock price's boxplot

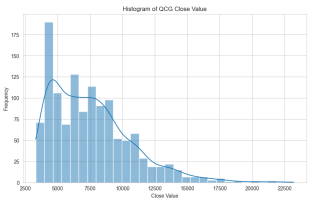

FIGURE 2. QCG stock price's histogram

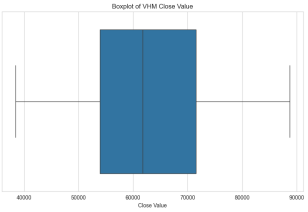

FIGURE 3. VHM stock price's boxplot

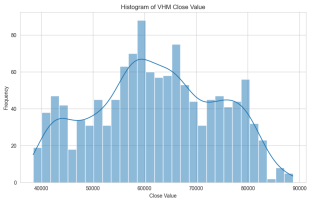

FIGURE 4. VHM stock price's histogram

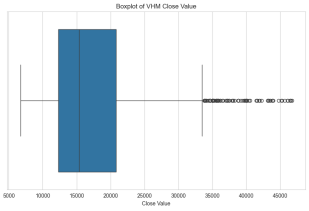

FIGURE 5. DXG stock price's boxplot

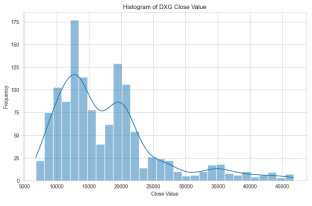

FIGURE 6. DXG stock price's histogram

IV. METHODOLOGY

A. DATA PREPROCESSING

The initial dataset of daily closing stock prices was incomplete, lacking data for weekends and potentially other non-trading days, resulting in a non-consecutive time series. Recognizing the importance of a continuous time series for accurate analysis, we took steps to fill these gaps. We assumed that the market doesn't experience significant changes over non-trading days and used the closing price of the preceding Friday to fill the missing values for weekends and holidays.

Furthermore, to enhance the predictive power of our models, we calculated several technical indicators from the closing prices. These indicators included Simple Moving Average (SMA), Exponential Moving Average (EMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Bollinger Bands (BB), Average True Range (ATR), and On-Balance Volume (OBV). These widely-used indicators provide valuable information about market trends, momentum, and volatility, serving as potential predictors in our linear regression model.

By addressing the missing data and incorporating technical indicators, we created a more comprehensive and informative dataset for our subsequent analysis. This enhanced dataset enabled us to explore the relationships between stock

prices and various market factors, ultimately contributing to the development of more accurate predictive models.

B. LINEAR REGRESSION

A linear regression model was employed to analyze the relationship between the closing price of real estate company stocks and various technical indicators. Linear regression is a statistical method that models the linear relationship between a dependent variable and one or more independent variables. In this context, the closing price of real estate stocks was chosen as the dependent variable, while several technical indicators derived from the stock's price and volume data were considered as potential independent variables.

The mathematical representation of a multiple linear regression model is as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

Where:

- Y is the predicted closing price of the real estate stock.
- $X_1, X_2, \ldots, X_k$ are the independent (explanatory) variables.
- β_0 is the intercept term.
- $\beta_1, ..., \beta_k$ are the regression coefficients for the independent variables.
- ε is the error term.

The dataset used for this analysis included stock price data for various real estate companies, spanning a specific time period. The dataset included various technical indicators such as Simple Moving Average (SMA), Exponential Moving Average (EMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Bollinger Bands (BB_High, BB_Middle, BB_Low), Average True Range (ATR), and On-Balance Volume (OBV). These indicators were selected as potential independent variables due to their established relevance in technical stock analysis.

C. RANDOM FOREST

Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result.

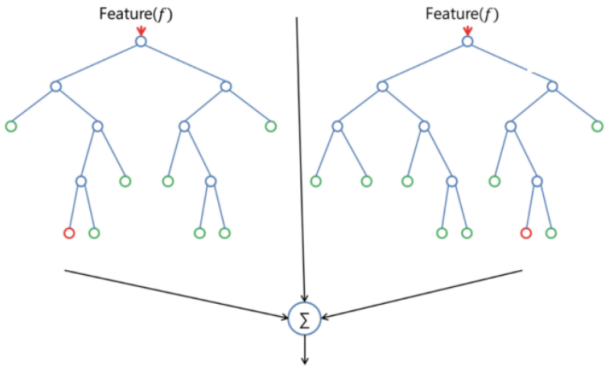

FIGURE 7. Random forest models

Random forests are also very hard to beat performance-wise. Of course, you can probably always find a model that can perform better — like a neural network, for example — but these usually take more time to develop, though they can handle a lot of different feature types, like binary, categorical and numerical. Overall, random forest is a (mostly) fast, simple and flexible tool, but not without some limitations.

D. GRU

GRU is a simplified version of LSTM (Long Short-Term Memory) and has fewer parameters, which helps reduce the time and computational resources required during model training. Both GRU and LSTM belong to the family of advanced recurrent neural network architectures that can retain information over long sequences without encountering gradient degradation issues. The structure of GRU consists of two main gates:

- Update Gate: Controls the amount of information from the previous hidden state that needs to be carried over to the current state.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

- Reset Gate: Decides how much of the past information to forget.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

- Current memory content : determines the potential contribution to the updated hidden state, allowing the network to retain or update information effectively.

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h)$$

- Final memory at current time step : is the updated hidden state that combines the previous hidden state and the new candidate hidden state based on the update gate's decision. This updated hidden state effectively balances retaining information from the past and incorporating new information from the current time step.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

Where:

- h_t is the final hidden state at time step t. This represents the updated memory of the network at the current time step.
- z_t is the update gate vector at time step t. The update gate controls how much of the previous hidden state should be carried forward to the current hidden state.
- h_{t-1} is the hidden state from the previous time step $t-1$. This is the memory of the network from the prior time step.
- $\tilde{h}_t$ is the candidate hidden state at time step t. It represents the new information that could be added to the hidden state, calculated using the current input and the reset-modified previous hidden state.
- $\odot$ represents element-wise multiplication. This operation is applied element-wise to vectors or matrices.
- $1 - z_t$ is the complement of the update gate vector. It represents the proportion of the previous hidden state that should be retained.

Given the high correlations between the *Close* price and other price-based indicators (*Open*, *High*, *Low*, *SMA_20*, *EMA_20*, *BB_High*, *BB_Middle*, *BB_Low*), we will exclude these to avoid multicollinearity issues. The remaining indicators (*Volume*, *Change %*, *RSI*, *MACD*, *MACD_Signal*, *MACD_Diff*, *ATR*, *OBV*) will be considered as potential independent variables in the linear regression model.

bibliography/Figure/correlation_arimax.png

FIGURE 8. Correlation Matrix of Filtered Data

The model was then trained using the preprocessed dataset, and its performance was evaluated using appropriate metrics such as Mean Squared Error (MSE), R-squared, and adjusted R-squared.

The choice of independent variables for the final model was guided by the correlation matrix (Figure **??**), which revealed the strength and direction of linear relationships between the closing price and each indicator. Variables exhibiting higher correlation with the closing price were considered more influential and were prioritized for inclusion in the model.

By analyzing the estimated coefficients $(\beta_1, \beta_2, ..., \beta_n)$ of the linear regression model, we can quantify the impact of each technical indicator on the predicted closing price of real estate stocks. This analysis provides valuable insights into the factors that drive the stock's price movements and can inform investment decisions.

E. ARIMAX

Stock prices, much like weather patterns, exhibit both inherent trends and reactions to external forces. To capture this duality, we employed an Autoregressive Integrated Moving Average with Exogenous variables (ARIMAX) model. Building upon the established ARIMA framework, ARIMAX allows us to weave external factors, or "exogenous variables," into our forecasting tapestry. Mathematically, the ARIMAX model is expressed as:

$$y(t) = \alpha + \sum_{i=1}^{p} \beta_i y(t-i) + \sum_{j=1}^{q} \phi_j \varepsilon(t-j) + \sum_{k=1}^{r} \gamma_k x_k(t) + \varepsilon(t)$$

In our context, $y(t)$ represents the real estate stock price at time (t). The terms $(\sum_{i=1}^{p} \beta_i y(t-i))$ and $(\sum_{j=1}^{q} \phi_j \varepsilon(t-j))$ capture the autoregressive (AR) and moving average (MA) components, respectively, similar to ARIMA. The added dimension lies in $(\sum_{k=1}^{r} \gamma_k x_k(t))$, where $x_k(t)$ are our carefully curated exogenous variables, namely the technical indicators derived from our preprocessed dataset.

The selection of the ARIMAX model's parameters – the number of AR and MA terms, the order of differencing, and the specific exogenous variables – was a meticulous process. It involved scrutinizing autocorrelation and partial autocorrelation plots, along with consulting information criteria like AIC and BIC.

With the model's architecture finalized, we trained it on our refined dataset, where the closing prices served as the main melody, and the technical indicators provided the counterpoint. We assessed the model's performance using various metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), seeking to minimize these measures of forecasting error.

This ARIMAX model, enriched by the inclusion of exogenous variables, allowed us to not only decipher the inherent rhythm of real estate stock prices but also to anticipate their movements based on the external influences captured by the technical indicators. It represents a step towards a more holistic understanding of the stock market's complex dance.

F. RNN

G. LONG SHORT TERM MEMORY (LSTM)

Long Short Term Memory networks (LSTM), often known as LSTMs, are a special type of recurrent neural network (RNN) with the ability to learn and remember long-term dependencies. LSTMs were introduced by Hochreiter and Schmidhuber in 1997, and have since been refined and developed further by many researchers and experts in the field. Thanks to their exceptional performance on various tasks, LSTMs have become increasingly popular.

LSTMs are designed to address the problem of long-term dependencies. Retaining information over extended periods is an inherent characteristic of LSTMs, requiring no special training to achieve this capability. In other words, the ability to remember long-term information is built into LSTMs.

Unlike traditional RNNs, which have a simple structure with a single tanh activation layer, LSTMs have a more complex chain-like structure, with modules that contain up to four layers interacting in a special way.

In the t-th state of the LSTM model:

Output: c_t, h_t, where c is the cell state, and h is the hidden state.

Input: c_{t-1}, h_{t-1}, x_t, where x_t is the input at state t of the model, and c_{t-1} and h_{t-1} are the outputs from the previous layer. The hidden state h is similar to s in RNN, while c is the unique aspect of LSTM.

Reading the diagram: The symbols σ and tanh indicate that the step uses the sigmoid and tanh activation functions, respectively. The multiplication is element-wise, and the addition is matrix addition.

Gates: f_t, i_t, o_t correspond to the forget gate, input gate, and output gate, respectively.

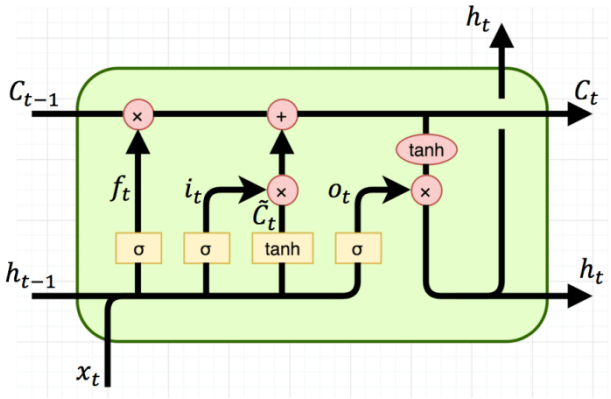

FIGURE 9. LSTM Model

- **Forget gate**:

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

- **Input gate**:

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

- **Output gate**:

$$o_t = \sigma(U_o \cdot x_t + W_o \cdot h_{t-1} + b_o)$$

Thus, the expressions for each gate of the LSTM illustrate how each gate manages the information flowing in and out of the model's states.

H. TIMESNET

I. FAST FOURIER TRANSFORM FORECASTING MODEL (FFT)

V. RESULT

Placeholder line

A. EVALUATION METHODS

Mean Percentage Absolute Error (MAPE): is the average percentage error in a set of predicted values.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| = 1$$

Root Mean Squared Error (RMSE): is the square root of average value of squared error in a set of predicted values.

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

Mean Absolute Error (MSLE):is the relative difference between the log-transformed actual and predicted values.

$$MSLE = \frac{1}{n} \sum_{i=1}^{n} (log(1 + \hat{y}_i) - log(log(1 + y_i))^2$$

) Where:

- n is the number of observations in the dataset.
- y_i is the true value.
- $\hat{y}_i$ is the predicted value.

B. DXG DATASET

DXG Dataset's Evaluation				
Model	Training-Testing	RMSE	MAPE (%)	MSLE
ARIMA	7-3			
	8-2			
	9-1			
GRU	7-3			
	8-2			
	9-1			
Linear Regression	7-3			
	8-2			
	9-1			
LSTM	7-3	240.04897	3.15	48.59
	8-2	262.16156	3.28	50.31
	9-1	205.6617	2.05	50.0
RF	7-3			
	8-2			
	9-1			
RNN	7-3			
	8-2			
	9-1			
TimesNet	7-3	8119.47	10.67	42.17
	8-2	3175.16	21.21	44.08
	9-1	2398.97	16.05	47.31

TABLE 2. DXG's Evaluation

C. VHM DATASET

D. QCG DATASET

VI. CONCLUSION

Placeholder line

A. SUMMARY

Placeholder

B. FUTURE CONSIDERATIONS

Placeholder line

ACKNOWLEDGMENT

Placeholder line

REFERENCES

[1] Hind Daori, Alanoud Alanazi, Manar Alharthi, Ghaida Alzahrani , "Predicting Stock Prices Using the Random ForestClassifier", November, 14th, 2022.
[2] Hugo Souto, 2023, "TimesNet for Realized Volatility Prediction ".
[3] Bohumil Stádník, Jurgita Raudeliuniene, Vida Davidavičienė , 2016. Fourier Analysis For Stock Price Forecasting: Assumption And Evidence .