FFT, TimesNet, and Random Forest in Real Estate Stock Market Analysis

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 investigates the effectiveness of Fast Fourier Transform (FFT), Time Series Network (TimeSNet), and Random Forest (RF) models in predicting stock prices within the Vietnamese real estate market. We apply these models independently to historical daily closing prices of three major real estate companies from 2019 to 2024, exploring how each method contributes to understanding and forecasting stock price movements. FFT is utilized to reveal underlying periodic patterns, TimeSNet to capture temporal dependencies, and RF to provide robust predictions. The results offer insights into the strengths and weaknesses of each model for this specific market, providing valuable information for investors and policymakers.

INDEX TERMS Placeholder

I. INTRODUCTION

Time-series forecasting plays a crucial role in decision-making across various domains. Its significance lies in its ability to provide valuable insights into future trends and patterns in time-dependent data. For instance, accurate predictions of stock prices, interest rates, and foreign exchange rates are essential for informed investment decisions in finance. Similarly, healthcare organizations rely on forecasting patient demand and resource utilization to allocate resources effectively and improve patient care. Energy management companies use time series forecasting to optimize energy production, distribution, and consumption. The accuracy and efficiency of time-series forecasting models significantly impact organizational performance and decision-making processes.

In this paper, we explore an innovative approach to enhance time-series forecasting using the Fast Fourier Transform (FFT). The FFT algorithm extracts frequency-domain features from time series data, offering a promising avenue for improving forecast accuracy and computational efficiency. Our investigation involves a comparative analysis of models trained with FFT-based features against traditional time domain features. We apply this approach to predict stock prices of real estate companies, leveraging not only FFT but also other techniques such as TimesNet and Random Forest. Through our study, we shed light on the interpretability of frequency domain features and their relationship with underlying time series patterns, emphasizing the potential of FFT-based feature engineering in enhancing 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) [1] 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) [2] 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 [3], 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

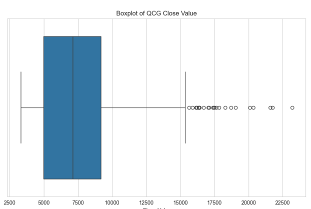

FIGURE 1. QCG stock price's boxplot

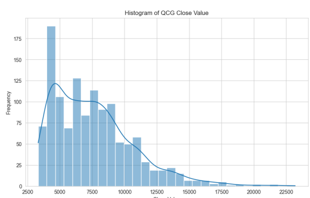

FIGURE 2. QCG stock price's histogram

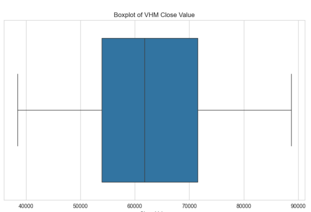

FIGURE 3. VHM stock price's boxplot

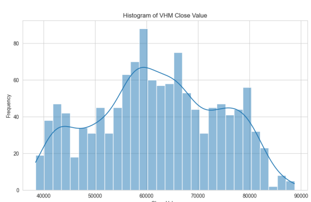

FIGURE 4. VHM stock price's histogram

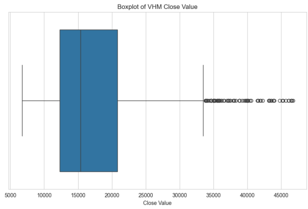

FIGURE 5. DXG stock price's boxplot

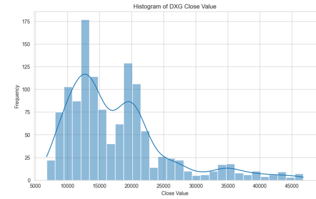

FIGURE 6. DXG stock price's histogram

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.
- $\beta_1, \ldots, \beta_k$ 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.

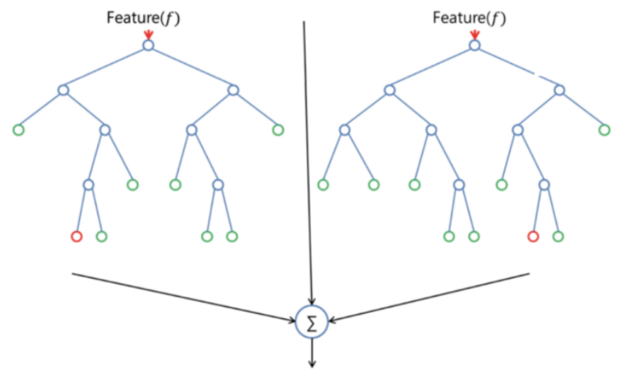

FIGURE 7. Random forest models

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.

TABLE 2. 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 (Table 2), 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. ARIMA

F. RNN

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to recognize patterns in sequences of data, such as text, time series data, and speech. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles, allowing information to persist. This characteristic enables RNNs to exhibit temporal dynamic behavior, making them suitable for tasks where context and sequential data play a critical role.

The core concept behind Recurrent Neural Networks (RNNs) is the introduction of a hidden state that captures information about previous inputs. This hidden state is updated at each time step as new inputs are processed. Mathematically, this process can be represented as follows:

$$h_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t + b)$$

where:

- h_t is the hidden state at time t.
- W_h and W_x are weight matrices.
- x_t is the input at time t.
- b is the bias term.
- σ is a non-linear activation function (e.g., $\tanh$ or ReLU).

Recurrent Neural Networks have significantly advanced the field of sequence modeling, enabling numerous applications across diverse domains. Despite their challenges, ongoing research and innovations continue to enhance their performance and applicability. By understanding and addressing the limitations of RNNs, we can unlock their full potential and pave the way for even more sophisticated and capable sequential data models.

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.

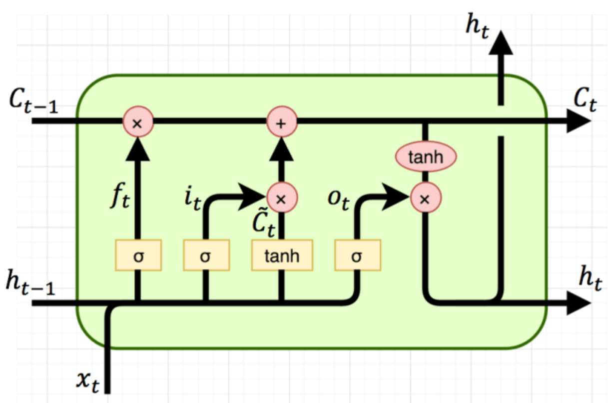

FIGURE 8. LSTM Model

- **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)

The fast Fourier transform (FFT) is a computational tool that transforms time-domain data into the frequency domain by deconstructing the signal into its individual parts: sine and cosine waves. This computation allows engineers to observe

the signal's frequency components rather than the sum of those components.
The FFT forecasting model leverages the fact that any periodic time series can be represented as a sum of sinusoidal functions (sines and cosines) of different frequencies. By transforming the time series data into the frequency domain, we can isolate significant frequencies that capture the underlying periodic patterns.
Given a time series $x(t)$, the Fast Fourier Transform (FFT) converts it into the frequency domain $X(f)$:

$$X(f) = \sum_{t=0}^{N-1} x(t)e^{-i2\pi ft/N}$$

where:

- N is the number of data points.
- f represents different frequency components.

To reconstruct the time series from significant frequencies:

$$x(t) = \sum_{f \in F} X(f)e^{i2\pi ft/N}$$

where F is the set of significant frequencies.

The Fast Fourier Transform Forecasting Model is a powerful tool for analyzing and forecasting time series data with periodic components. By transforming data into the frequency domain, it enables the identification of significant patterns and trends, offering an efficient and effective approach to time series forecasting.

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 (MAE):is the relative difference between the log-transformed actual and predicted values.

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

where:

- n is the number of observations.
- $\hat{y}_i$ represents the predicted values.
- y_i represents the actual values.

B. DXG DATASET

DXG Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MSLE
LN	7:2:1	608.64	4.27	46.33
	8:2	11729.2	10.825	0.019
	9:1	**7933.49**	**7.47**	**0.007**
ARIMA	7:3	11864.3	7.52	0.021
	8:2	8521.33	5.01	0.009
	9:1	**7006.54**	**3.73**	**0.006**
GRU	7:3	648.57	1.066	0.0002
	8:2	**612.68**	**1.0074**	**0.00017**
	9:1	613.20	0.01023	0.00015
RNN	7:3	1185.91	2.26	0.00066
	8:2	**580.55**	**0.95**	**0.00015**
	9:1	617.77	1.030	0.015
RF	7:3	734.07	1.252	0.00027
	8:2	**696.60**	**1.22**	**0.00021**
	9:1	813.82	1.464	0.00025
LTSM	7:2:1	608.64	4.27	46.33
	7.5-1.5-1	604.48	3.86	45.81
	8-1.5-0.5	399.35	1.95	52.26
TimesNet	7:2:1	8119.99	10.67	42.17
	7.5-1.5-1	11519.35	15.87	41.93
	8-1.5-0.5	3700.29	4.21	43.01
FFT	7:3	941.7588	1.7384	0.0005
	8:2	939.7588	1.6546	0.0005
	9:1	**936.8374**	**1.6273**	**0.0005**

TABLE 3. DXG Dataset's Evaluation

C. VHM DATASET

VHM Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MSLE
LN	7:2:1	608.64	4.27	46.33
	8:2	11729.2	10.825	0.019
	9:1	**7933.49**	**7.47**	**0.007**
ARIMA	7:3	11864.3	7.52	0.021
	8:2	8521.33	5.01	0.009
	9:1	**7006.54**	**3.73**	**0.006**
GRU	7:3	1789.84	2.03	0.001
	8:2	**1383.71**	**1.59**	**0.0004**
	9:1	1412.4	1.54	0.0004
RNN	7:3	1919.39	2.15	0.0011
	8:2	1516.528	1.769	0.0005
	9:1	**1389.80**	**1.555**	**0.0004**
RF	7:3	1751.89	2	0.00094
	8:2	**1539.70**	**1.83**	**0.00056**
	9:1	1620.2	1.81	0.00055
LTSM	7:2:1	608.64	4.27	46.33
	7.5-1.5-1	604.48	3.86	45.81
	8-1.5-0.5	399.35	1.95	52.26
TimesNet	7:2:1	8119.99	10.67	42.17
	7.5-1.5-1	11519.35	15.87	41.93
	8-1.5-0.5	3700.29	4.21	43.01
FFT	7:3	941.7588	1.7384	0.0005
	8:2	939.7588	1.6546	0.0005
	9:1	**936.8374**	**1.6273**	**0.0005**

TABLE 4. VHM Dataset's Evaluation

D. QCG DATASET

QCG Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MSLE
LN	7:2:1	608.64	4.27	46.33
	8:2	11729.2	10.825	0.019
	9:1	**7933.49**	**7.47**	**0.007**
ARIMA	7:3	11864.3	7.52	0.021
	8:2	8521.33	5.01	0.009
	9:1	**7006.54**	**3.73**	**0.006**
GRU	7:3	648.57	1.535	0.00039
	8:2	**563.91**	**0.009**	**0.00016**
	9:1	926.86	1.614	0.00044
RNN	7:3	1919.39	2.15	0.0011
	8:2	1516.528	1.769	0.0005
	9:1	**1389.80**	**1.555**	**0.0004**
RF	7:3	827.19	1.129	0.000317
	8:2	**566.2**	**0.92**	**0.00017**
	9:1	838.29	1.426	0.00036
LTSM	7:2:1	608.64	4.27	46.33
	7.5-1.5-1	604.48	3.86	45.81
	8-1.5-0.5	399.35	1.95	52.26
TimesNet	7:2:1	8119.99	10.67	42.17
	7.5-1.5-1	11519.35	15.87	41.93
	8-1.5-0.5	3700.29	4.21	43.01
FFT	7:3	941.7588	1.7384	0.0005
	8:2	939.7588	1.6546	0.0005
	9:1	**936.8374**	**1.6273**	**0.0005**

TABLE 5. QCG Dataset's Evaluation

VI. CONCLUSION

A. SUMMARY

In the pursuit of forecasting stock prices, various methodologies have been explored, ranging from traditional statistical models to advanced machine learning algorithms. Among the examined models, Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), Random Forest, Fast Fourier Transform (FFT), TimesNet, and Auto Regressive Integrated Moving Average (ARIMA) stand out. Notably, RNN, GRU, LSTM, and Random Forest have emerged as the most promising and effective models for predicting stock prices.

The intricacies of stock price forecasting, rooted in the complexity and unpredictability of financial markets, demand models capable of capturing nuanced patterns and relationships within the data. Recurrent Neural Networks (RNN) excel at handling sequential data, offering robust predictions. Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) models, with their enhanced ability to capture sequential dependencies, exhibit notable performance in stock price forecasting. Random Forest, with its ensemble learning approach, further refines predictive capabilities by combining multiple decision trees to provide collective insights

As evidenced by evaluation metrics such as RMSE, MAPE, and MSLE, the RNN, GRU, LSTM, and Random Forest models consistently demonstrate superior performance across various aspects of forecasting accuracy. Their adaptability in managing the inherent uncertainties of stock markets positions them as formidable tools for investors and analysts seeking reliable predictions.

B. FUTURE CONSIDERATIONS

The exploration of forecasting stock prices using Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), Random Forest, Fast Fourier Transform (FFT), TimesNet, and Auto Regressive Integrated Moving Average (ARIMA) models has yielded initial insights, yet the accuracy of these models remains a challenge. To enhance the predictive performance and applicability of these models, several future considerations are warranted:

- Integrating more diverse and high-frequency data sources, such as social media sentiment, macroeconomic indicators, and real-time news feeds, could enrich the dataset, allowing models to capture a more comprehensive range of market dynamics and improve forecasting accuracy.
- Utilizing and possibly developing more comprehensive evaluation metrics beyond RMSE, MAPE, and MSLE to assess model performance can provide deeper insights into their effectiveness and areas needing improvement.
- Developing hybrid models that combine the strengths of different approaches could yield better performance. For example, integrating the sequential learning capabilities of RNNs with the ensemble approach of Random Forests could result in models that better capture both short-term and long-term market patterns.

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 .