Journal on Information Technologies & Communications

Exploring rebalancing strategies in forecast-informed portfolio optimisation with deep learning techniques on VN100 stocks

Minh Duc Nguyen[1]
[1]Department of Economic Information Systems, University of Economics, Hue University, Hue City, Vietnam

Correspondence: Minh Duc Nguyen (Email: nguyenminhduc@hueuni.edu.vn)
Communication: received 16 Oct 2025, revised 16 Nov 2025, accepted 5 Dec 2025
DOI: 10.31276/ict.2025.1416

***Abstract*-This paper proposes a forecast-driven portfolio optimisation framework that integrates deep learning models with a modified mean-variance strategy. Specifically, One-Dimensional Convolutional Neural Networks (1D-CNN) and Long Short-Term Memory (LSTM) networks are employed to predict next-day stock returns, which are then used as inputs to the Mean-Variance with Forecasting (MVF) model. The framework is evaluated on VN100 stock data from 2015 to 2024 under various rebalancing frequencies, including 30, 60, 90, and 120 trading days, as well as a no-rebalancing baseline. Empirical results show that rebalancing every 60 trading days consistently yields superior portfolio performance across both predictive models, achieving the highest cumulative returns and favourable risk-adjusted metrics. The findings highlight the importance of selecting an appropriate rebalancing frequency when leveraging model-based forecasts for portfolio allocation. The proposed approach offers a promising direction for enhancing data-driven investment strategies through the integration of deep learning and portfolio theory.**

Keywords: *deep learning, Stock return prediction, Portfolio optimisation, Rebalancing strategy, VN100 stocks.*

1. Introduction

Portfolio optimisation is a central problem in quantitative finance, where the goal is to allocate capital across a set of assets to maximise return while controlling for risk. Since the seminal work by Markowitz (1952) [1], the mean-variance framework has served as the foundation for modern portfolio theory. As a result, recent research has increasingly focused on integrating predictive models into portfolio optimisation to dynamically adapt to evolving market conditions [2], [3].

With the rise of machine learning and deep learning, return forecasting has seen substantial improvements in accuracy and generalisability. Models such as long short-term memory (LSTM) networks [4] and one-dimensional convolutional neural networks (1D-CNN) have demonstrated strong performance in capturing temporal dependencies in financial time series [5]. These models can be trained on sequences of historical returns and deployed to forecast short-term price movements, making them promising tools for enhancing portfolio allocation strategies. However, effective integration of predicted returns into optimisation frameworks remains an open challenge. Forecast errors and estimation uncertainty can lead to unstable allocations and overfitting, particularly when rebalancing is performed too frequently. To address this, the mean-variance with forecasting (MVF) model introduces a modified objective function that accounts for predicted returns and their associated errors, in addition to traditional risk measures [2]. This approach aims to better align allocation decisions with forecast reliability, while preserving the tractable structure of mean-variance optimisation.

Despite extensive research on deep learning-based return forecasting and portfolio optimisation, the role of rebalancing frequency within a forecast-driven optimisation framework remains largely unexplored. Existing studies often employ a fixed rebalancing interval or overlook rebalancing design entirely, even though allocation changes based on unreliable or excessively frequent forecasts can substantially degrade performance. Moreover, prior work predominantly focuses on developed markets, leaving a lack of evidence on how forecast-integrated optimisation behaves in emerging markets where volatility and structural shifts are more pronounced. This gap motivates our examination of how different fixed rebalancing intervals influence portfolio outcomes when combined with deep learning forecasts within the MVF model.

This paper explores a hybrid strategy that combines deep learning-based return forecasts with the MVF optimisation model. Specifically, we apply both 1D-CNN and LSTM models to predict next-day returns of VN100 stocks, and use these forecasts within the MVF framework to construct optimised portfolios. The performance of the resulting portfolios is evaluated across multiple rebalancing frequencies and a no-rebalancing baseline. The goal is to assess the effect of rebalancing frequency on forecast-based portfolio performance and identify the interval that best balances responsiveness and stability. The main contributions of this work are as follows: (1) we develop and test a forecasting-integrated MVF framework using deep learning return predictions; (2) we empirically analyse the role of rebalancing frequency in determining portfolio outcomes; and (3) we provide comparative insights into the behaviour of 1D-CNN and LSTM-based investment strategies on real-world financial data.

2. Related work

Portfolio optimisation seeks to allocate capital among a group of assets in a way that maximises expected performance while adhering to a defined risk profile. The seminal framework in this domain is the mean–variance optimisation (MVO) model introduced by Markowitz (1952) [1], which formulates the problem as minimising portfolio variance for a specified expected return, or equivalently maximising return for a given level of risk. Classical MVO assumes that both expected returns and the covariance matrix of asset returns are known beforehand and remain stable throughout the investment horizon. This assumption, however, is often unrealistic in practical settings. As a result, numerous extensions have been developed to enhance MVO's robustness. For instance, robust portfolio optimisation [6] mitigates the impact of parameter uncertainty by optimising for worst-case scenarios, while Bayesian frameworks [7] blend sample-based estimates with prior beliefs to produce more stable estimates of mean returns. Nonetheless, accurate return forecasting continues to be a key determinant of portfolio performance, as even small prediction errors can substantially distort optimal weight allocations [8], [9].

Forecasting financial returns remains highly challenging due to market efficiency, stochastic noise, and structural regime shifts. Traditional econometric models such as ARIMA [10], GARCH [11], and VAR [12] capture serial dependence and volatility dynamics but generally fail to model complex nonlinear interactions. Factor-based approaches, including the Fama-French three-factor model [13], introduce systematic risk factors to improve explanatory capability, though their predictive performance for short-term returns remains limited. The emergence of machine learning and deep learning has revolutionised forecasting by enabling the modelling of nonlinear, high-dimensional relationships [14]. Algorithms such as Random Forests [15], Gradient Boosting Machines [16], and deep architectures like LSTM networks [4] have demonstrated strong capabilities in uncovering intricate temporal and cross-sectional dependencies in financial datasets [17], [18]. In particular, deep learning-based forecasts have been shown to enhance risk-adjusted returns when embedded in portfolio optimisation processes [2], [19]. Furthermore, recent research has started to integrate considerations such as transaction costs and liquidity constraints into optimisation frameworks, although these studies predominantly focus on developed markets [3].

Integrating forecasts into portfolio optimisation entails refining the allocation process so that it accounts not only for the level of expected returns but also for the reliability of those predictions. Unlike the traditional mean-variance approach, which assumes that expected returns are known and fixed, forecast-sensitive models adjust the optimisation objective by introducing penalties for assets with higher prediction uncertainty [2], [3], [20]. Similar ideas of incorporating model uncertainty have also been explored in adaptive predictive control frameworks, where online model updating is used to maintain robustness under disturbances and parameter variation [21]. This adjustment can be implemented using indicators such as past forecasting errors or confidence intervals generated from predictive models [3], [22]. By assigning weights to forecasts based on their estimated accuracy, these models aim to mitigate overreliance on volatile or unreliable signals, resulting in enhanced robustness and improved out-of-sample performance.

3. Research methodology

3.1. Workflow of the proposed framework

To provide an intuitive overview of the entire experimental process, Figure 1 illustrates the workflow of the proposed forecast-driven portfolio optimisation framework.

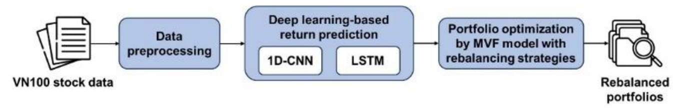

Fig. 1. Overview of the probing framework.

The workflow consists of three main stages. First, VN100 stock data undergo preprocessing, including the computation of daily returns and normalisation. Second, two deep learning models (1D-CNN and LSTM) are trained to forecast returns using rolling windows of historical data. Finally, the predicted returns are integrated into the MVF model, which performs portfolio optimisation under different rebalancing strategies. This end-to-end process enables a systematic evaluation of how rebalancing frequency impacts the performance of forecast-informed portfolios.

3.2. Data collection and preprocessing

This study utilised daily closing price data of VN100 stocks from January 2015 to December 2024. After removing stocks with insufficient historical data, the final dataset comprised 50 stocks. Daily returns were computed from the price series, and all return values were normalised to ensure a consistent scale and facilitate model convergence. Each model input consisted of a rolling window of the past 30 trading days' returns, used to predict the return for the next day. This sliding-window approach captures short-term temporal patterns in the data. Standardisation was applied to each return sequence using the following transformation:

$$\hat{d}_i = \frac{d_i - \mu}{\sigma}$$

where d_i is the return at time i, and μ and σ are the mean and standard deviation of the stock's returns during the training period.

The dataset was partitioned chronologically into three subsets: the training set (January 2015–December 2021), the validation set (January 2022–December 2022), and the test set (January 2023–December 2024). The deep learning models described in the following sections were trained to forecast next-day returns using the standardised return sequences. Forecasts generated from the test set were then used as inputs for portfolio optimisation via the MVF model. All implementations were carried out in Python, leveraging libraries such as NumPy, Pandas, TensorFlow, Scikit-learn, and Matplotlib for data processing, model training, and visualisation.

3.3. One-dimensional convolutional neural network (1D-CNN)

The 1D-CNN model was designed to capture localised temporal patterns in the return sequences of individual stocks. Unlike traditional CNNs for images, 1D-CNNs process one-dimensional time-series inputs. The architecture includes a convolutional layer with 64 filters and a kernel size of 3, followed by a max-pooling layer (pool size = 2) to reduce dimensionality. The output is flattened and passed through a dense layer with 64 units and a dropout rate of 0.2, before reaching a final linear output unit. ReLU is used in all hidden layers. The model was trained using the Adam optimiser (learning rate = 0.001) with mean squared error (MSE) as the loss function, and early stopping (patience = 100 epochs) was applied to prevent overfitting.

Hyperparameters such as filter size, kernel size, learning rate, and dropout rate were tuned via trial and error. The selected configuration demonstrated the best validation performance and was used to generate return forecasts for portfolio optimisation.

3.4. Long short-term memory (LSTM) network

The LSTM model was employed to capture sequential dependencies in stock return time series. As a type of recurrent neural network (RNN), LSTM is designed to retain long-term information via memory cells and gating mechanisms, making it suitable for modelling financial time series. Each input consists of a standardised sequence of past daily returns for a given stock. The LSTM layer processes these sequences and outputs a single prediction for the next day's return. This output is passed through fully connected layers with ReLU activation to refine the forecast, ending with a linear output layer for regression. Dropout regularisation and early stopping were applied to mitigate overfitting.

Hyperparameters were tuned through trial and error. The best performance was achieved using 64 LSTM units, a batch size of 64, a dropout rate of 0.2, and a learning rate of 0.001, optimised with the Adam optimiser and MSE loss function.

3.5. Mean-Variance with Forecasting (MVF) model

To allocate portfolio weights based on predictive signals, we adopt the MVF model [2], which extends the classical Markowitz framework by integrating return forecasts and their associated uncertainties. This formulation allows the optimiser to consider not only historical risk but also the expected return and forecast confidence for each asset. The optimisation problem is defined as follows:

$$\min\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma_{ij} - \sum_{i=1}^{n} x_i \hat{r}_i - \sum_{i=1}^{n} x_i \bar{\varepsilon}_i\right)$$

subject to the constraints:

$$\sum_{i=1}^{n} x_i = 1 \text{ and } 0 \le x_i \le 1 \text{ for } i = 1,2,\dots,n$$

In this formulation, x_i is the portfolio weight allocated to asset i, σ_{ij} is the covariance between the returns of assets i and j, $\hat{r}_i$ is the predicted return of asset i, and $\bar{\varepsilon}_i$ is the average prediction error (uncertainty) of asset i.

The model aims to balance risk reduction with return enhancement and prediction confidence. This optimisation is performed at each rebalancing point using the most recent forecasts $\hat{r}_i$, historical covariance σ_{ij}, and average forecast errors $\bar{\varepsilon}_i$. Between rebalancing dates, the asset weights are held fixed. This dynamic process allows the portfolio to adapt over time in response to updated market conditions.

4. Empirical evaluation

4.1. Performance metrics

This study evaluates both the prediction accuracy of two deep learning models (1D-CNN and LSTM), and the portfolio performance under various rebalancing strategies when using predicted returns in the MVF model. To assess how well the 1D-CNN and LSTM models predict stock returns, we use the following metrics as shown in Table 1.

Table 1: Forcasting performance metrics

Metric	Formula	Metric	Formula
Mean Absolute Error (MAE)	$\frac{1}{n}\sum_{t=1}^{n}\lvert y_t - \hat{y}_t\rvert$	Total hit rate (H_R)	$\frac{count(y_t \cdot \hat{y}_t > 0)}{count(y_t \cdot \hat{y}_t \neq 0)}$
Mean Squared Error (MSE)	$\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2$	Positive hit rate (H_{R+})	$\frac{count(y_t > 0 \; and \; \hat{y}_t > 0)}{count(\hat{y}_t > 0)}$
Root Mean Squared Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$	Negative hit rate (H_{R-})	$\frac{count(y_t < 0 \; and \; \hat{y}_t < 0)}{count(\hat{y}_t < 0)}$

Let y_t denote the actual return and $\hat{y}_t$ denote the predicted return at time t. The total number of time steps in the evaluation period is represented by n. The MAE calculates the average absolute difference between the predicted and actual returns, offering a straightforward interpretation of error magnitude. The MSE measures the average of the squared differences and penalizes larger deviations more heavily, making it more sensitive to outliers. The RMSE is the square root of MSE, expressed in the same units as the returns, allowing for easier interpretation. The H_R measures the proportion of times the model correctly predicts the direction of return. The H_{R+} computes the fraction of actual positive returns that are also predicted as positive, while the H_{R-} evaluates the percentage of correct predictions among all actual negative return instances.

For each model (1D-CNN and LSTM), the predicted returns are fed into the MVF model. The resulting portfolios are then evaluated under different rebalancing frequencies using the metrics presented in Table 2.

Table 2: Portfolio optimization performance metrics

Metric	Formula
Mean Daily Return (MDR)	$\frac{1}{n}\sum_{t=1}^{n} r_t^{(p)}$
Standard Deviation of Daily Return (SDDR)	$\sqrt{\frac{1}{n-1}\sum_{t=1}^{n}\left(r_t^{(p)} - \bar{r}^p\right)^2}$
Total Return (TR)	$\prod_{t=1}^{n}\left(1 + r_t^{(p)}\right) - 1$

Let $r_t^{(p)}$ denote the return of the optimized portfolio on trading day t, and let n be the total number of trading days in the evaluation period. The term $\bar{r}^p$ represents the mean daily return of the portfolio over n days. The MDR is calculated as the arithmetic average of daily portfolio returns over the evaluation period. The SDDR quantifies the variability in daily portfolio returns and serves as a proxy for portfolio risk. Lastly, the TR reflects the overall percentage gain or loss of the portfolio across the entire period, accounting for the compounding of daily returns.

4.2. Results

4.2.1. Forecasting stock returns

Table 3 reports the forecasting performance metrics for the 1D-CNN and LSTM models used to predict daily stock returns. For each model, both the mean and standard deviation (Std) of the metric values are presented.

While forecasting accuracy is not the primary focus of this study, the results confirm that both models perform reasonably well and are stable across different runs. The LSTM model exhibits slightly lower error values and marginally higher directional hit rates than the 1D-CNN, suggesting modest improvements in predictive capability. Nonetheless, both models provide sufficiently reliable forecasts to support the subsequent portfolio optimisation analysis.

Table 3: Forecasting performance metrics for the 1D-CNN and LSTEM models

Metric		MAE	MSE	RMSE	H_R	H_{R+}	H_{R-}
1D-CNN	Mean	1.58×10^{-2}	4.93×10^{-4}	2.16×10^{-2}	48.27%	43.98%	43.62%
	Std	4.16×10^{-3}	2.31×10^{-4}	5.36×10^{-3}	2.57×10^{-2}	3.95×10^{-2}	3.14×10^{-2}
LSTM	Mean	1.55×10^{-2}	4.73×10^{-4}	2.12×10^{-2}	48.28%	44.02%	43.67%
	Std	4.00×10^{-3}	2.11×10^{-4}	5.07×10^{-3}	2.47×10^{-2}	3.86×10^{-2}	3.93×10^{-2}

Table 4: Effect of rebalancing frequency on forecast-based MVF for pỏtfolio performance

Strategy		MDR (%)	SDDR	TR (%)
1D-CNN+MVF	No rebalancing	0.049	0.006105	24.735
	Every 30 days	0.065	0.009233	30.259
	Every 60 days	**0.104**	**0.011087**	**48.964**
	Every 90 days	0.053	0.010976	19.392
	Every 120 days	0.027	0.009110	8.386
LSTM+MVF	No rebalancing	0.049	0.006121	24.804
	Every 30 days	0.049	0.009083	21.803
	Every 60 days	**0.075**	**0.010295**	**33.022**
	Every 90 days	0.033	0.011094	10.760
	Every 120 days	0.037	0.009501	11.883

4.2.2. Optimising portfolios under different rebalancing frequencies

Building on the return forecasts, we evaluate portfolio optimisation performance by combining each deep learning model with the MVF framework. For both models, optimised portfolios are tested under various rebalancing strategies as well as a no-rebalancing baseline where initial asset weights remain fixed. This experimental setup enables a systematic assessment of how rebalancing frequency influences investment outcomes when guided by forecasted returns. Transaction costs are excluded to isolate the pure effect of rebalancing and model accuracy. The analysis aims to evaluate not only the predictive models' effectiveness in portfolio construction but also the trade-offs between responsiveness and stability in rebalancing decisions.

Table 4 presents the impact of different rebalancing frequencies on portfolio performance using the MVF framework with return forecasts from 1D-CNN and LSTM models. The strategies vary in rebalancing intervals from no rebalancing to every 30, 60, 90, and 120 trading days.

For the 1D-CNN+MVF strategy, rebalancing every 60 days yields the highest performance across all metrics, with an MDR of 0.104%, an SDDR of 0.011087, and a TR of 48.964%. This suggests that a moderate rebalancing frequency captures the benefit of updated allocations while avoiding excessive turnover or instability. Similarly, for the LSTM+MVF strategy, the 60-day rebalancing interval again achieves the best results, with an MDR of 0.075%, an SDDR of 0.010295, and a TR of 33.022%. Compared to both more frequent (30-day) and less frequent (90-day, 120-day) rebalancing, this configuration strikes a better balance between return and risk. Overall, the results indicate that 60-day rebalancing is the most effective strategy for both forecasting models within the MVF framework, delivering superior total returns and improved return-to-risk trade-offs.

To further illustrate the effect of rebalancing frequency on portfolio performance, Figures 2 and 3 present the cumulative return curves of MVF-optimised portfolios using forecasts generated by 1D-CNN and LSTM models, respectively. Each figure compares five strategies along with a no-rebalancing baseline.

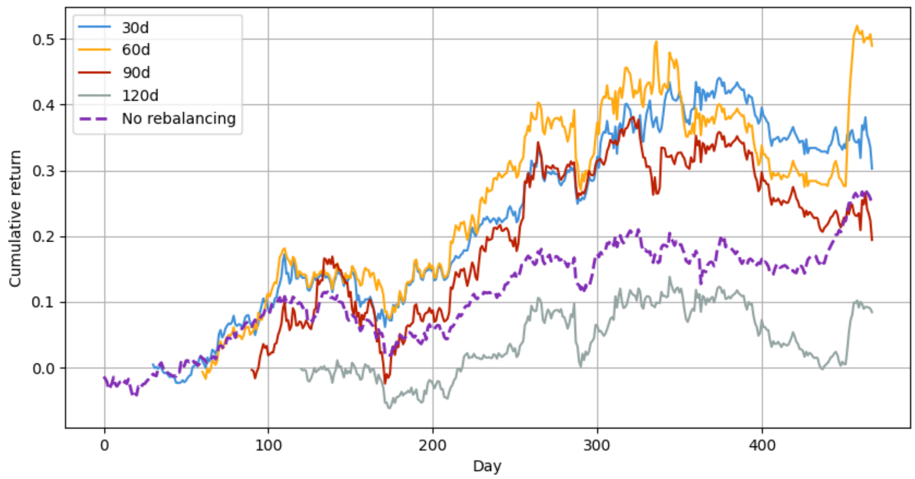

Fig. 2. Cumulative return of MVF-optimised portfolios using 1D-CNN forecasts under different rebalancing frequencies.

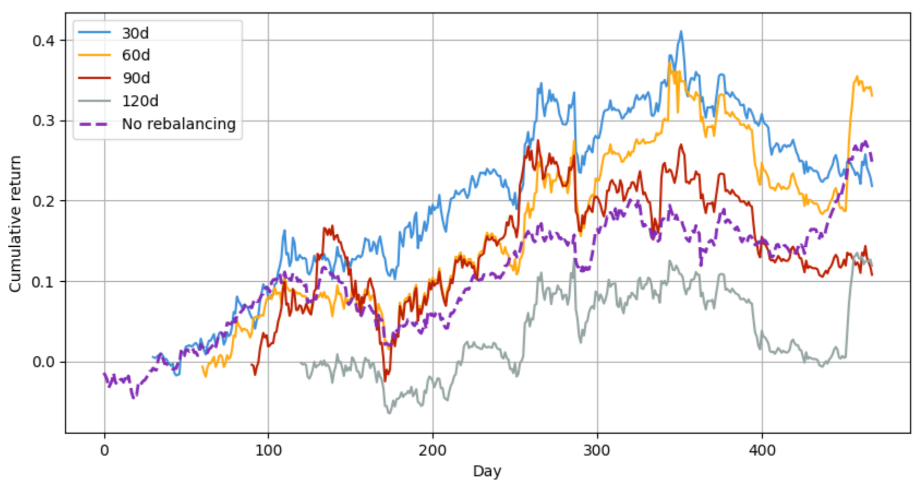

Fig. 3. Cumulative return of MVF-optimised portfolios using LSTM forecasts under different rebalancing frequencies.

These two figures clearly illustrate that rebalancing every 60 trading days consistently produces the strongest cumulative performance for both the 1D-CNN and LSTM-based MVF portfolios. The trajectories show that the 60-day strategy not only achieves the highest peak returns but also maintains steadier growth over time compared with both more aggressive and more conservative schedules. In contrast, rebalancing too frequently (such as every 30 days) introduces higher volatility. This occurs because the portfolio reacts too quickly to short-term forecast noise, resulting in excessive turnover and unstable allocation adjustments that diminish long-term performance. On the other hand, infrequent rebalancing (90 or 120 days) causes the portfolio to lag behind evolving market conditions, particularly during periods of trend reversals or emerging price momentum. The no-rebalancing baseline shows smooth but noticeably lower returns, confirming that periodic updating of the MVF-optimised weights is essential when forecasts provide meaningful signals. Overall, the comparative patterns across all strategies highlight that rebalancing frequency plays a decisive role in balancing responsiveness and stability, with the 60-day interval emerging as the optimal compromise for maximising risk-adjusted portfolio growth.

V. Discussion and conclusion

This study investigated a forecasting-integrated portfolio optimisation framework that combines deep learning models, 1D-CNN and LSTM, with the MVF strategy. By leveraging model-generated return predictions, the framework dynamically reallocates portfolio weights at fixed rebalancing intervals to enhance performance over a traditional static allocation. Experimental results on VN100 stock data from 2015 to 2024 show that rebalancing frequency significantly affects portfolio outcomes. In particular, rebalancing every 60 trading days consistently yielded the highest cumulative returns and favourable risk-return profiles for both 1D-CNN and LSTM-based forecasts. This frequency appears to strike an effective balance between responsiveness to new forecast information and portfolio stability, outperforming both more frequent (30-day) and less frequent (90- and 120-day) rebalancing strategies. The no-rebalancing baseline, while more stable, achieved relatively lower returns.

Although the LSTM model achieved marginally better forecasting accuracy in terms of both error magnitude and directional prediction, the overall portfolio outcomes generated by LSTM and 1D-CNN were largely comparable once integrated into the MVF optimisation framework. This convergence in performance indicates that forecast accuracy alone does not dominate portfolio behaviour; instead, the interaction between forecasting signals and the rebalancing mechanism plays a decisive role. Across both models, the empirical evidence consistently shows that a 60-day rebalancing interval yields smoother growth trajectories, fewer drawdowns, and the highest cumulative returns. This pattern suggests that moderate rebalancing effectively captures meaningful forecast updates while avoiding excessive turnover or sensitivity to short-term fluctuations. In contrast, overly frequent (30-day) rebalancing amplifies noise and leads to unstable adjustments, while infrequent (90- or 120-day) updates fail to incorporate evolving market conditions. These findings highlight that, within forecast-informed portfolio strategies, the choice of rebalancing frequency is as critical as the choice of predictive model, with moderate intervals offering the best balance between responsiveness and stability.

Future work may extend this framework by incorporating transaction costs, turnover constraints, and alternative risk measures such as Conditional Value at Risk (CVaR). In addition, exploring more sophisticated forecasting ensembles or adapting the model to different market regimes could further enhance its robustness and applicability.

Acknowledgment

This study was funded by the University of Economics, Hue University (project number: DHKT2025-04-34).

REFERENCES

[1] H. Markowitz (1952), "Portfolio Selection", *J. Finance*, **7(1)**, pp.77-91, DOI: 10.2307/2975974.

[2] J.R. Yu, W.J.P. Chiou, W.Y. Lee, et al (2020), "Portfolio models with return forecasting and transaction costs", *International Review of Economics & Finance*, **66**, pp.118-130, DOI: 10.1016/j.iref.2019.11.002.

[3] Y. Ma, R. Han, W. Wang (2021), "Portfolio optimization with return prediction using deep learning and machine learning", *Expert Syst. Appl.*, **165**, DOI: 10.1016/j.eswa.2020.113973.

[4] S. Hochreiter, J. Schmidhuber (1997), "Long short-term memory", *Neural Comput.*, **9(8)**, pp.1735-1780, DOI: 10.1162/neco.1997.9.8.1735.

[5] T. Kim, H. Y. Kim (2019), "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLoS One*, **14(2)**, DOI: 10.1371/journal.pone.0212320.

[6] A. Ghahtarani, A. Saif, A. Ghasemi (2022), "Robust portfolio selection problems: a comprehensive review," *Operational Research*, **22(4)**, pp.3203-3264, DOI: 10.1007/s12351-022-00690-5.

[7] P. Jorion (1986), "Bayes-Stein Estimation for Portfolio Analysis", *The Journal of Financial and Quantitative Analysis*, **21(3)**, pp.279-292, DOI: 10.2307/2331042.

[8] R. C. Merton (1980), "On estimating the expected return on the market: An exploratory investigation", *J. financ. econ.*, **8(4)**, pp.323-361, DOI: 10.1016/0304-405X(80)90007-0.

[9] V. DeMiguel, L. Garlappi, F. J. Nogales, et al. (2009), "A Generalized Approach to Portfolio Optimization: Improving Performance by Constraining Portfolio Norms", *Manage. Sci.*, **55(5)**, pp.798-812, DOI: 10.1287.mnsc.1080.0986.

[10] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, et al. (2015), *Time series analysis: forecasting and control*, pp.161-215, John Wiley & Sons.

[11] T. Bollerslev (1986), "Generalized autoregressive conditional heteroskedasticity", *J. Econom.*, **31(3)**, pp.307-327, DOI: 10.1016/0304-4076(86)90063-1.

[12] C.A. Sims (1980), "Macroeconomics and Reality", *Econometrica*, **48(1)**, pp.1-48, DOI: 10.2307/1912017.

[13] E.F. Fama, K.R. French (1993), "Common risk factors in the returns on stocks and bonds", *J. financ. econ.*, **33(1)**, pp.3-56, DOI: 10.1016/0304-405X(93)90023-5.

[14] M. Abdelsattar, A. AbdelMoety, A. Emad-Eldeen (2025), "Advanced machine learning techniques for predicting power generation and fault detection in solar photovoltaic systems", *Neural Comput. Appl.*, **37(15)**, pp.8825-8844, DOI: 10.1007/s00521-025-11035-6.

[15] L. Breiman (2001), "Random Forests", *Mach. Learn.*, **45(1)**, pp.5-32, DOI: 10.1023/A:1010933404324.

[16] J.H. Friedman (2001), "Greedy Function Approximation: A Gradient Boosting Machine", *The Annals of Statistics*, **29(5)**, pp.1189-1232, DOI: 10.1214/aos/1013203451.

[17] T. Fischer, C. Krauss (2018), "Deep learning with long short-term memory networks for financial market predictions", *Eur. J. Oper. Res.*, **270(2)**, pp.654-669, DOI: 10.1016/j.ejor.2017.11.054.

[18] S. Gu, B. Kelly, D. Xiu (2020), "Empirical Asset Pricing via Machine Learning", *Rev. Financ. Stud.*, **33(5)**, pp.2223-2273, DOI: 10.1093/rfs/hhaa009.

[19] C. Krauss, X. A. Do, N. Huck (2017), "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500", *Eur. J. Oper. Res.*, **259(2)**, pp.689-702, DOI: 10.1016/j.ejor.2016.10.031.

[20] E.S. Oda, M. Ebeed, A.M.A. Hamed, A. Ali, et al. (2021), "Optimal allocation of a hybrid photovoltaic-based DG and DSTATCOM under the load and irradiance variability", *International Transactions on Electrical Energy Systems*, **31(2)**, 25pp, DOI: 10.1002/2050-7038.13131.

[21] A.A.Z. Diab, M.A.E. Sattar (2018), "Adaptive model predictive based load frequency control in an interconnected power system", *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp.604-610, DOI: 10.1109/EIConRus.2018.8317170.

[22] A. Kourtis, G. Dotsis, R. N. Markellos (2012), "Parameter uncertainty in portfolio selection: Shrinking the inverse covariance matrix", *J. Bank. Financ.*, **36(9)**, pp.2522-2531, DOI: 10.1016/j.bankfin.2012.05.005.

Minh Duc Nguyen is a lecturer at the Department of Economic Information Systems, University of Economics, Hue University, Hue, Vietnam. He received his M.Eng. and Ph.D. degrees in Information Systems from Inje University, Korea, in 2017 and 2021, respectively.

His research interests include data mining, ontology engineering, semantic technologies, knowledge management, and blockchain systems.