Balancing Return Maximization and Downside Risk Control in Emerging Markets: A Machine Learning and Deep Learning Approach to Portfolio Optimization

Minh Duc Nguyen
Department of Economic Information Systems, University of Economics Hue University, Hue, Vietnam

Abstract

Background: Emerging markets pose challenges for portfolio optimization due to volatility and the non-linear behavior of stock returns. Traditional methods often fail to capture these complexities, highlighting the need for advanced forecasting combined with robust optimization. **Objectives:** This study develops a framework that integrates machine learning and deep learning forecasting models with optimization strategies to improve stock return prediction and portfolio performance. **Methods/Approach:** Using VN-100 index stocks, four models, including Support Vector Regression (SVR), Random Forest (RF), Long Short-Term Memory (LSTM), and One-Dimensional Convolutional Neural Networks (1D-CNN), were applied to forecast returns. The top 30 stocks per model were selected. Eight portfolios were constructed by combining these forecasts with two optimization methods: Mean-Variance with Forecasting (MVF) for return maximization and Maximum Drawdown Portfolio (MDP) for downside risk control. Performance was tested over three years using net asset value trajectories and monthly excess returns. **Results:** MVF portfolios achieved higher mean returns with lower volatility compared to MDP. The RF-MVF portfolio delivered the strongest risk-adjusted performance, while LSTM-based stock sets enhanced portfolio outcomes. **Conclusions:** Integrating advanced forecasting with optimization improves portfolio construction in volatile markets. The findings show that balancing return maximization and downside risk control yields superior risk-adjusted returns, offering practical guidance for investors in emerging economies.

Keywords: stock return forecasting; portfolio optimization; stock selection; machine learning; deep learning; VN-100

JEL classification: C61, G11, G17
Paper type: Research article

Received: 9 September 2025
Accepted: 25 July 2026

Citation: Nguyen, M. D. (2026). Balancing Return Maximization and Downside Risk Control in Emerging Markets: A Machine Learning and Deep Learning Approach to Portfolio Optimization. *Business Systems Research, 17*(2), 149–165.
DOI: https://doi.org/10.2478/bsrj-2026-0023

Introduction

Accurate stock return forecasting and effective portfolio optimization are critical components of modern investment decision-making, particularly in emerging markets where volatility and uncertainty are prevalent (Gu et al., 2020). Traditional approaches, such as Markowitz's mean-variance optimization (Markowitz, 1952, 1959), have long provided a foundation for constructing portfolios by balancing expected returns against risk. However, the rapid evolution of financial markets and the increasing availability of high-frequency data have spurred the adoption of advanced forecasting techniques that can capture complex market dynamics.

Recent research indicates that advanced forecasting techniques can significantly outperform traditional linear methods in capturing the complex, non-linear dynamics of financial markets. Machine learning approaches such as Support Vector Regression (SVR) and ensemble methods like Random Forest (RF) have been increasingly applied due to their ability to model intricate relationships within data (Gu et al., 2020; Hastie et al., 2001). Moreover, deep learning techniques, particularly Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN), have demonstrated considerable promise in handling sequential and temporal dependencies, leading to improved forecasting accuracy in various financial applications (Heaton et al., 2017; Zhang et al., 2024).

This study presents an integrated framework that merges advanced forecasting models with portfolio optimization techniques to improve stock return prediction and portfolio construction in emerging markets. We employ four state-of-the-art forecasting models, including SVR, RF, LSTM, and One-Dimensional CNN (1D-CNN) to capture the complex, non-linear dynamics inherent in financial time series. A dataset consisting of stocks from the VN-100 index, which represents the top companies listed on major Vietnamese exchanges, is used to assess forecasting accuracy using a broad range of metrics. Based on these metrics, we implement a stock selection process to identify the top 30 stocks for each forecasting model. These selected stocks are then used to assess portfolio performance via two distinct optimization approaches, Mean-Variance with Forecasting (MVF), which targets return maximization, and Maximum Drawdown Portfolio (MDP), which focuses on downside risk control. The resulting eight distinct portfolio configurations enable an analysis of how different stock selection methods influence overall performance. For a more detailed analysis of investment returns, we concentrate on the stock set chosen by the model achieving the best performance. This analysis provides insights into the risk-return dynamics across different model-portfolio combinations.

Our contributions are threefold. First, we provide a comprehensive evaluation of four advanced forecasting models in the context of emerging market stock return prediction, using a diverse set of performance metrics. Second, we introduce a tailored stock selection process based on forecasting accuracy to examine how different sets of stocks impact portfolio optimization outcomes. Third, we systematically compare eight unique portfolios, offering novel insights into the trade-offs between return maximization and downside risk control in volatile markets.

The remainder of this paper is organized as follows. The related work on forecasting and portfolio optimization techniques is reviewed first. Next, the comprehensive methodology is described, including data collection, preprocessing, forecasting models, and portfolio optimization strategies. The subsequent part presents the experimental results and discussion, focusing on the performance of stock return predictions, portfolio optimization with stock selection, and the analysis of investment returns. Finally, the paper concludes with key findings and outlines directions for future research.

Related work

The literature on portfolio construction originates from Markowitz's mean-variance framework, which determines asset allocations by balancing expected returns against variance-based risk (Markowitz, 1952, 1959). Although mean-variance optimization remains foundational, later studies have proposed alternative risk measures and optimization formulations to address some of its practical limitations. Konno and Yamazaki (1991), for example, introduced the mean-absolute deviation model, which replaces variance with an absolute-deviation risk measure and enables portfolio optimization through linear programming. Rockafellar et al. (2000) proposed conditional value-at-risk (CVaR) optimization to control losses in the tail of the return distribution. Chekhlov et al. (2005) developed drawdown-based portfolio measures that focus on declines from previous portfolio peaks, which are particularly relevant for investors concerned with downside risk and capital preservation. These approaches demonstrate that portfolio optimization may be designed around different investor objectives, including variance control, tail-loss reduction, and drawdown limitation.

Alongside developments in portfolio optimization, financial forecasting research has increasingly examined machine learning and deep learning methods because asset returns may exhibit non-linear relationships, interactions among predictors, and temporal dependence that are difficult to represent using simpler linear specifications. SVR has been applied to financial time-series forecasting because its kernel-based structure can model non-linear relationships while controlling model complexity. Kim (2003), for example, applied support vector machines (SVM) to stock index forecasting and compared their performance with neural-network and case-based reasoning approaches. RF has also been used in stock-market prediction because its ensemble structure can capture non-linear relationships and interactions while reducing the instability associated with individual decision trees. Patel et al. (2015) compared SVM, RF, artificial neural networks, and naive-Bayes methods for predicting stock and stock-index movements, illustrating the use of machine learning models for financial prediction tasks.

Deep learning models further extend this line of research by learning representations directly from sequential financial data. LSTM networks are designed to capture temporal dependencies through their gated memory structure and have been used extensively in financial-market prediction. Fischer and Krauss (2018) applied LSTM networks to directional prediction for S&P 500 stocks and found stronger predictive and trading performance than selected memory-free benchmark models in their empirical setting. CNN-based models have also been considered for financial time-series forecasting because convolutional filters can extract local sequential patterns from return or price histories. Reviews by Sezer et al. (2020), Gao et al. (2024), Li and Law (2024), and Zhang et al. (2024) document the increasing use of LSTM, CNN, and other machine learning and deep learning architectures in financial forecasting applications. However, the empirical effectiveness of these models remains context-dependent and may vary according to market, sample period, input features, and evaluation design.

A related stream of research connects return prediction with portfolio construction. Gu et al. (2020) show that machine learning methods can improve the modeling of expected returns and support economically meaningful portfolio analysis in asset-pricing applications. Krauss et al. (2017) employ deep neural networks, gradient-boosted trees, and RF for equity selection in a statistical-arbitrage setting, demonstrating how model-generated signals can inform investment decisions.

Heaton et al. (2017) further discuss the use of deep learning in portfolio-oriented financial applications. These studies indicate that forecasting models can contribute to portfolio decisions either by identifying promising assets or by supplying expected-return inputs for portfolio allocation.

Despite this progress, relatively limited evidence is available on frameworks that simultaneously compare multiple machine learning and deep learning forecasting models, use forecasting accuracy to construct alternative stock-selection sets, and evaluate both a return-oriented optimization approach and a downside-risk-oriented approach within the same emerging-market setting. This study addresses this gap using stocks from the VN-100 index. Specifically, SVR, RF, LSTM, and 1D-CNN are used to generate predicted returns and forecasting-based stock-selection sets, while MVF and MDP are applied to examine the trade-off between forecast-based return generation and maximum-drawdown control. In doing so, the study provides evidence on how forecasting-model choice, stock-selection basis, and portfolio-optimization objective jointly influence portfolio outcomes in the Vietnamese equity market.

Methodology

In this section, we present our comprehensive methodological framework that underpins the study. We begin by detailing our data collection and preprocessing procedures, which ensure the reliability and suitability of the financial data for subsequent analysis. This is followed by a description of the advanced forecasting models and their hyperparameter optimization, and culminates in the integration of these forecasts into portfolio optimization models.

Data collection and preprocessing

The dataset for this study is sourced from the VN-100 index, a prominent benchmark in the Vietnamese stock market that reflects the performance of the top 100 companies listed on the country's major exchanges. Serving as a key indicator of market performance and investor sentiment in one of Asia's emerging markets, the VN-100 provides a robust basis for analysis. For our study, we extracted the daily closing prices for stocks within this index from 2015 to 2024. To ensure data quality, we excluded stocks that were not continuously listed or that contained missing values, resulting in a final dataset of 50 stocks.

To effectively capture recent market trends, we adopt a sliding window approach that uses the most recent 30 days of daily returns as input for our prediction models. This technique emphasizes short-term market movements while keeping the input size manageable for our machine learning and deep learning algorithms. Financial return series may contain extreme observations, particularly in emerging markets where episodes of sharp price adjustment and elevated volatility are common. We therefore applied an outlier-capping procedure based on the interquartile range (IQR). Let Q_1 and Q_3 represent the first and third quartiles of a feature, respectively, and define the IQR as $IQR = Q_3 - Q_1$. Each feature value d_i is then clipped to the interval: $[Q_1 - 5 * IQR, Q_3 + 5 * IQR]$. In other words, the capped value $\tilde{d}_i$ is given by:

$$\tilde{d}_i = clip(d_i, Q_1 - 5 * IQR, Q_3 + 5 * IQR) = max\{Q_1 - 5 * IQR, min\{d_i, Q_3 + 5 * IQR\}\} \quad (1)$$

Daily stock returns may contain unusually large but economically meaningful movements, especially in a volatile emerging-market setting. Therefore, we adopted a relatively wide $(5 * IQR)$ threshold to cap only highly extreme observations while preserving substantial return movements that may carry relevant market information. Under this procedure, observations falling below $(Q_1 - 5 * IQR)$ or above $(Q_3 + 5 * IQR)$

were replaced by the corresponding lower or upper boundary, respectively, thereby limiting the influence of extreme values on subsequent analyses. Using this rule, 5,584 out of 124,950 return observations were clipped, corresponding to 4.47% of the dataset. After outlier capping, each feature is standardized to have zero mean and unit variance. Specifically, each original feature value x is transformed into a standardized value z using:

$$z = \frac{x-\mu}{\sigma} \quad (2)$$

where μ and σ denote the mean and standard deviation of the feature, respectively. This transformation places all features on a common scale, enhancing the stability and efficiency of the model training process.

The year-based split was constructed chronologically from the start of the sample period. The first five years, 2015-2019, were allocated to the training set to provide a sufficiently large historical sample for estimating the forecasting models. The following two years, 2020-2021, were used as the validation set to select hyperparameters and assess model generalization on subsequent observations before final testing. The remaining three years, 2022-2024, were reserved for out-of-sample testing, providing a sufficiently long period for evaluating both forecasting accuracy and portfolio performance. This allocation balances the data requirements for model estimation, validation, and final evaluation while preserving the chronological structure of the financial time series.

Machine learning and deep learning models

In this study, machine learning and deep learning models are used to forecast stock returns based on daily returns. Their purpose is to capture potentially non-linear and temporal patterns in return movements and to provide forecasting performance measures for selecting stocks prior to portfolio optimization. For each forecasting model, hyperparameters determine important aspects of model structure, complexity, and learning behavior. Appropriate hyperparameter selection is therefore necessary to improve predictive accuracy and reduce the risk of overfitting, particularly when modeling noisy financial time series.

Support Vector Regression (SVR)

SVR is a widely used machine learning technique for stock return prediction due to its ability to capture non-linear relationships while maintaining robust generalization performance. Unlike traditional regression models that minimize squared error, SVR uses a margin-based approach, fitting a function within a predefined tolerance level ε. This approach enables the model to focus on significant deviations, making it particularly effective for financial time series forecasting, where market data can be highly complex and noisy. The prediction function is defined as:

$$f(x_j) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\, K(x_i, x_j) + b \quad (3)$$

where $f(x_j)$ is the predicted stock return for observation x_j, x_i denotes the (i)-th training input vector, α_i and α_i^* are the estimated coefficients, $K(x_i, x_j)$ is the kernel function, and (b) is the intercept term.

In this study, the Radial Basis Function (RBF) kernel is employed to enhance the model's capacity to capture non-linear dependencies in stock return data. The RBF kernel transforms the input space into a higher-dimensional feature space, allowing SVR to model intricate market trends more effectively. It is defined as:

$$K(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right) \quad (4)$$

where γ is a hyperparameter that determines the flexibility of the decision function. A larger γ value results in a more complex model that can fit the training data closely, while a smaller γ yields a smoother function that might underfit the data.

To optimize the SVR model's performance, hyperparameters including the regularization parameter C, the kernel coefficient γ, and the margin ε are tuned using a grid search in combination with cross-validation. This systematic approach evaluates different combinations of hyperparameters to identify the optimal settings that minimize the prediction error and achieve a balance between bias and variance. Table 1 summarizes the range of values considered for the grid search of the RBF kernel parameters.

Table 1
Hyperparameter grid for SVR with RBF kernel

Hyperparameter	Values explored	Effect on the model
C	1, 5, 10, 15, 20, 25	Controls the penalty for errors outside the margin; higher values fit training errors more strictly.
γ	0.001, 0.01, 0.1, 1	Controls the RBF kernel sensitivity; higher values produce a more flexible, localized fit.
ε	0.001, 0.01, 0.1, 1	Defines the error-tolerance margin; higher values ignore larger small errors.

Source: Author's work

Random Forest (RF)

RF is an ensemble learning method that constructs multiple decision trees and aggregates their predictions to improve regression accuracy. In contrast to single-tree models, RF reduces variance by averaging the outcomes of many trees, each trained on different random subsets of the data and features. This mechanism allows RF to capture complex interactions among financial indicators while remaining robust to overfitting when modeling the noisy behavior of stock returns. For stock return forecasting, RF leverages its inherent randomness to model non-linear relationships without extensive preprocessing. The model's performance is largely influenced by its hyperparameters, which govern aspects such as the number of trees, depth of each tree, and the minimum number of samples required for splits. These hyperparameters are systematically tuned to enhance predictive accuracy using the hyperparameter grid presented in Table 2.

Table 2
Hyperparameter grid for RF

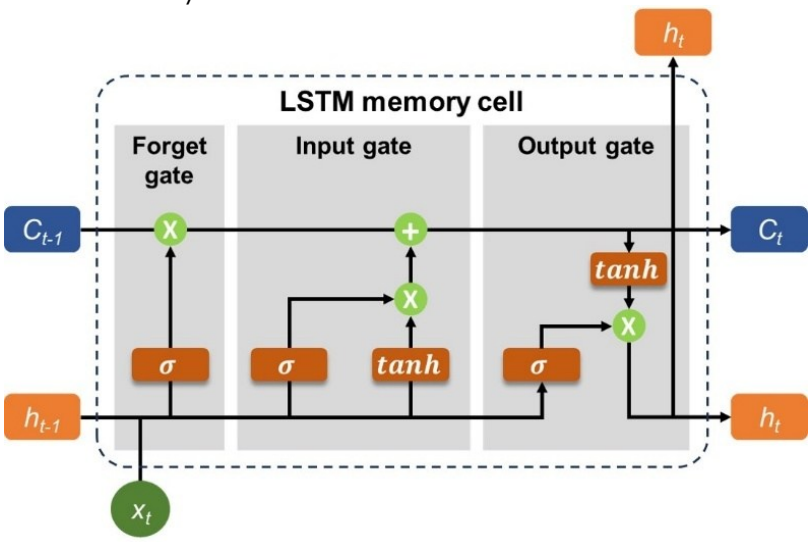
Hyperparameter	Values explored	Effect on the model
n_estimators	100, 250, 500	Number of trees; more trees generally improve stability but increase computation.
max_depth	10, 20, 30	Maximum tree depth; larger values capture more complex patterns but may overfit.
min_samples_split	5, 10, 15	Minimum samples needed to split a node; higher values produce more conservative trees.
min_samples_leaf	5, 10, 15	Minimum samples in a leaf; higher values smooth predictions and reduce overfitting.
max_features	10, 20, 30	Number of features considered per split; controls tree diversity and model flexibility.

Source: Author's work

Long Short-Term Memory Network (LSTM)

LSTM networks have become a cornerstone in forecasting stock returns due to their capability to capture long-term dependencies in sequential data. Unlike traditional recurrent neural networks, LSTMs overcome the vanishing gradient problem by employing a memory cell that retains relevant information over extended periods (Hochreiter & Schmidhuber, 1997). This mechanism is particularly beneficial in financial time series analysis, where capturing historical trends and temporal dependencies is crucial for accurate predictions. At the core of the LSTM architecture is the memory cell (as illustrated in Figure 1), which utilizes three gates to regulate the flow of information. The forget gate discards outdated information, the input gate selectively updates the cell state with new information, and the output gate controls the exposure of the cell's state to subsequent network layers. This structured approach allows the LSTM to learn which patterns are significant for forecasting future stock returns.

Figure 1
LSTM memory cell architecture



Source: Adapted from Hochreiter and Schmidhuber (1997)

To optimize the LSTM model, a systematic grid search is conducted across various hyperparameters. Table 3 summarizes the ranges and values explored for each parameter.

Table 3
Hyperparameter grid for LSTM

Hyperparameter	Values explored	Effect on the model
Hidden nodes	5, 10, 15, 20	Controls the number of learned features within each LSTM layer.
Hidden layers	1, 2, 4, 8	Controls network depth and capacity to learn complex temporal patterns.
Learning rate	0.0001, 0.001, 0.01	Determines the step size during model training.
Patience	0, 5, 10	Controls early stopping when validation performance no longer improves.

Batch size	50, 100	Determines the number of observations used in each training update.
Dropout rate	0.1, 0.2, 0.3	Reduces overfitting by randomly dropping standard connections during training.
Recurrent dropout rate	0.1, 0.2, 0.3	Reduces overfitting in recurrent connections across time steps.
Loss function	Mean absolute error	Measures prediction error used to train the model.
Optimizer	Adam, SGD, RMSprop	Determines how model weights are updated during training.

Source: Author's work

One-Dimensional Convolutional Neural Network (1D-CNN)

1D-CNNs are increasingly employed in time series forecasting, including stock return prediction, due to their capacity to automatically extract local patterns from sequential data. By convolving filters along the temporal axis, 1D-CNNs can capture short-term dependencies in stock price movements, technical indicators, and other financial signals. These extracted features are then fed into fully connected layers to generate the final prediction. Compared to deeper recurrent models, 1D-CNNs often train more efficiently and can handle large-scale datasets effectively. To optimize the 1D-CNN's predictive performance, this study also performs a grid search over a set of hyperparameters. Table 4 summarizes the main hyperparameters and the ranges explored.

Table 4

Hyperparameter grid for 1D-CNN

Hyperparameter	Values explored	Effect on the model
Number of filters	32, 64, 128	Controls the number of local patterns extracted from the return sequence.
Kernel size	2, 3, 5	Determines the temporal length of patterns captured by each filter.
Number of conv layers	1, 2, 3	Controls the depth and complexity of extracted temporal features.
Pooling type	Max, Average	Determines how extracted features are summarized.
Pooling size	2, 4	Controls the reduction of temporal feature dimensions.
Dropout rate	0.0, 0.2, 0.5	Reduces overfitting by randomly dropping units during training.
Dense layer units	32, 64, 128	Controls the capacity of the final prediction layers.
Activation	ReLU, Tanh	Determines how non-linear patterns are transformed.
Optimizer	SGD, Adam, RMSprop	Determines how model weights are updated during training.
Learning rate	0.0001, 0.001, 0.01	Determines the step size during model training.
Batch size	32, 64	Determines the number of observations used per training update.
Epochs	50, 100	Determines the maximum number of training cycles.

Source: Author's work

Portfolio optimization models

Portfolio optimization is the final stage of our investment decision process, in which the predicted returns generated by the machine learning and deep learning models are translated into portfolio allocations. Specifically, each forecasting model is first used to identify a corresponding set of selected stocks. For each selected stock set, the predicted returns generated by SVR, RF, LSTM, and 1D-CNN are separately used as inputs to two portfolio optimization approaches: MVF and MDP. In MVF, predicted returns enter the objective function together with variance-based risk. In MDP, predicted returns are used to construct the portfolio value trajectory, and portfolio weights are selected to minimize the resulting maximum drawdown.

Mean-Variance with Forecasting (MVF)

The MVF model extends the traditional mean-variance framework by incorporating model-predicted returns into the portfolio allocation objective (Markowitz, 1952; Yu et al., 2020). In the present study, for each selected stock set, predicted returns generated by SVR, RF, LSTM, or 1D-CNN are used as the return inputs of the MVF optimization model. In addition, the model accounts for prediction reliability through the average forecast residuals. The MVF optimization problem is formulated as:

$$\min_{(\boldsymbol{x})}\left(\sum_{(i=1)}^{n}\sum_{(j=1)}^{n} x_i\, x_j \sigma_{ij} - \sum_{(i=1)}^{n} x_i\, \hat{r}_i - \sum_{(i=1)}^{n} x_i\, \bar{\varepsilon}_i\right) \quad (5)$$

subject to the constraints:

$$\sum_{(i=1)}^{n} x_i = 1, \quad 0 \le x_i \le 1, \quad i = 1,2,\dots,n \quad (6)$$

where x_i is the portfolio weight assigned 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 generated by the forecasting model used in the optimization stage, and $\bar{\varepsilon}_i$ is the average forecast residual for asset i, representing the adjustment for historical forecast error.

Maximum Drawdown Portfolio (MDP)

In contrast to MVF, the MDP model focuses on controlling downside risk by directly addressing the maximum drawdown which is the largest peak-to-trough decline in portfolio value over a specified period (Chekhlov et al., 2005). This metric is particularly valuable for investors prioritizing capital preservation during volatile market conditions. The MDP approach uses forecasted returns not only to estimate potential gains but also to evaluate scenarios that might lead to severe losses. Formally, let P_t denote the portfolio value at time t, which is defined as:

$$P_t = P_0 \prod_{i=1}^{t}(1 + w^T r_i) \quad (7)$$

where P_0 is the initial portfolio value, w represents the vector of portfolio weights, and r_i is the vector of asset returns at time i. The maximum drawdown (MDD) is then defined as:

$$\mathrm{MDD} = \max_{t\in[0,T]}\left(1 - \frac{P_t}{\max_{s\in[0,t]} P_s}\right) \quad (8)$$

The MDP model seeks to minimize this maximum drawdown by optimizing the asset allocation. The corresponding optimization problem is formulated as:

$$\min_{w} \mathrm{MDD}(w) \quad (9)$$

subject to the portfolio constraints as introduced in Eq. (5).

In this setup, forecasted returns are employed to simulate the evolution of the portfolio over time, allowing the model to identify periods where severe declines might occur. By directly minimizing the potential maximum loss, the MDP model provides a more conservative asset allocation strategy that explicitly targets downside risk. This approach is particularly attractive for risk-averse investors who seek to preserve capital during adverse market conditions.

Together, the MVF and MDP models offer complementary strategies for portfolio optimization. While MVF aims to maximize risk-adjusted returns by balancing expected gains against variance, MDP focuses on mitigating extreme losses, thereby providing a robust framework that addresses both performance and risk management in dynamic financial markets.

Evaluation metrics for stock return prediction

We evaluate the predictive performance of our stock return forecasting models using three widely adopted metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics quantify the difference between the actual returns and the predictions generated by each model, providing insight into their accuracy and reliability. Specifically, MAE is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (10)$$

where y_i is the actual stock return, $\hat{y}_i$ is the predicted return, and N is the total number of observations. MAE provides a straightforward measure of the average magnitude of the errors in a set of predictions, without considering their direction. Next, MSE is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (11)$$

MSE squares the errors before averaging them, thereby penalizing larger errors more heavily than MAE. This metric is particularly sensitive to outliers. Lastly, RMSE is the square root of MSE. RMSE retains the same units as the original data, facilitating direct interpretation of the error magnitude. Lower values for these metrics indicate a model with higher predictive accuracy.

Results and discussion

Performance of stock return prediction

The forecasting performance of SVR, RF, LSTM, and 1D-CNN was evaluated over the 2022–2024 test period by comparing the predicted daily returns with the corresponding actual daily returns for each stock. For each model, MAE, MSE, and RMSE were calculated separately for each stock, and Table 5 reports the mean and standard deviation of these stock-level metric values. The mean indicates the model's overall prediction error across the stock universe, whereas the standard deviation indicates how consistently the model performs across different stocks.

Table 5
The predictive performance of different models

Model		MAE	MSE	RMSE
SVR	**Mean**	$1.90*10^{-2}$	$7.89*10^{-4}$	$2.71*10^{-2}$
	Std	$4.59*10^{-3}$	$5.41*10^{-4}$	$7.50*10^{-3}$
RF	**Mean**	$2.31*10^{-2}$	$1.10*10^{-3}$	$3.24*10^{-2}$
	Std	$6.01*10^{-3}$	$4.97*10^{-4}$	$7.56*10^{-3}$
LSTM	**Mean**	$1.88*10^{-2}$	$7.14*10^{-4}$	$2.61*10^{-2}$
	Std	$4.73*10^{-3}$	$3.03*10^{-4}$	$5.84*10^{-3}$
1D-CNN	**Mean**	$1.91*10^{-2}$	$7.33*10^{-4}$	$2.65*10^{-2}$
	Std	**$4.76*10^{-3}$**	**$3.10*10^{-4}$**	$5.87*10^{-3}$

Source: Author's work
Note: SVR = Support Vector Regression; RF = Random Forest; LSTM = Long Short-Term Memory; 1D-CNN = One-Dimensional Convolutional Neural Network; MAE = Mean Absolute Error; MSE = Mean Squared Error; RMSE = Root Mean Squared Error; Std = Standard Deviation.

Table 5 shows that LSTM achieves the strongest overall forecasting performance, recording the lowest mean MAE ($1.88*10^{-2}$), MSE ($7.14*10^{-4}$), and RMSE ($2.61*10^{-2}$) among the four models. The 1D-CNN model performs closely behind LSTM, with mean MAE, MSE, and RMSE values of $1.91*10^{-2}$, $7.33*10^{-4}$, and $2.65*10^{-2}$, respectively. SVR also produces competitive results, particularly in terms of MAE, and records the lowest MAE standard deviation ($4.59*10^{-3}$), indicating stable absolute prediction errors across stocks. In contrast, RF reports the highest mean errors across all three measures, with an MAE of $2.31*10^{-2}$, an MSE of $1.10*10^{-3}$, and an RMSE of $3.24*10^{-2}$. Overall, the results indicate that LSTM provides the most accurate predictions in this study, while 1D-CNN and SVR remain relatively competitive and RF performs less effectively over the test period.

Performance of portfolio optimization with stock selection

In order to enhance our portfolio optimization, a stock selection process was applied to the initial dataset of 50 stocks. For each forecasting model, namely SVR, RF, LSTM, and 1D-CNN, the stocks were ranked according to their RMSE values, and the 30 stocks with the lowest RMSE were selected. This procedure produced four stock-selection sets: SVR-based, RF-based, LSTM-based, and 1D-CNN-based. The term "SVR-based," for example, refers to the set of 30 stocks selected according to the forecasting accuracy of the SVR model. Within each stock-selection set, eight portfolio types were examined by combining predicted-return inputs from the four forecasting models with the two portfolio optimization approaches, MVF and MDP. For example, RF-MVF refers to a portfolio optimized using the MVF approach with RF-predicted returns, whereas RF-MDP refers to a portfolio optimized using the MDP approach with RF-predicted returns. The performance of these eight portfolio types was compared across the four stock-selection sets over the 2022-2024 test period using monthly excess return (MER) relative to the VN-100 index, information ratio (IR), and total return (TR). Table 6 presents the performance of the eight portfolio types across the four stock-selection sets. Within each stock-selection set, the portfolio with the highest MER is highlighted in boldface.

Table 6

Performance metrics for portfolios with different stock selection approaches

Stock selection	Portfolio	MER		IR	TR (%)
		Mean (%)	Std		
SVR-based	SVR-MVF	0.289	0.050	0.058	10.103
	RF-MVF	**0.553**	**0.052**	**0.106**	**19.364**
	LSTM-MVF	0.550	0.049	0.113	19.262
	1D-CNN-MVF	0.230	0.054	0.043	08.041
	SVR-MDP	0.292	0.055	0.053	10.236
	RF-MDP	0.348	0.057	0.061	12.165
	LSTM-MDP	0.277	0.055	0.050	09.696
	1D-CNN-MDP	0.290	0.055	0.052	10.154
RF-based	SVR-MVF	0.907	0.042	0.215	31.761
	RF-MVF	**1.156**	**0.044**	**0.264**	**40.457**
	LSTM-MVF	0.861	0.048	0.178	30.123
	1D-CNN-MVF	0.613	0.049	0.124	21.456
	SVR-MDP	0.462	0.053	0.087	16.181
	RF-MDP	0.602	0.053	0.113	21.087
	LSTM-MDP	0.542	0.05	0.109	18.953
	1D-CNN-MDP	0.472	0.048	0.098	16.520
LSTM-based	SVR-MVF	0.971	0.04	0.243	33.987

	RF-MVF	**1.212**	**0.04**	**0.300**	**42.429**
	LSTM-MVF	0.891	0.046	0.195	31.181
	1D-CNN-MVF	0.670	0.046	0.145	23.439
	SVR-MDP	0.406	0.049	0.083	14.193
	RF-MDP	0.663	0.056	0.119	23.193
	LSTM-MDP	0.587	0.048	0.122	20.529
	1D-CNN-MDP	0.633	0.048	0.132	22.169
1D-CNN-based	SVR-MVF	0.978	0.043	0.226	34.223
	RF-MVF	**1.261**	**0.044**	**0.286**	**44.135**
	LSTM-MVF	0.989	0.050	0.198	34.611
	1D-CNN-MVF	0.722	0.051	0.142	25.268
	SVR-MDP	0.733	0.050	0.147	25.638
	RF-MDP	0.559	0.053	0.106	19.564
	LSTM-MDP	0.664	0.049	0.135	23.246
	1D-CNN-MDP	**0.417**	**0.051**	**0.082**	14.578

Source: Author's work

Note: SVR = Support Vector Regression; RF = Random Forest; LSTM = Long Short-Term Memory; 1D-CNN = One-Dimensional Convolutional Neural Network; MVF = Mean-Variance with Forecasting; MDP = Maximum Drawdown Portfolio; MER = Monthly Excess Return; Std = Standard Deviation; IR = Information Ratio; TR = Total Return. SVR-based, RF-based, LSTM-based, and 1D-CNN-based refer to the stock sets selected using the respective forecasting models. Within each selected stock set, the portfolio labels indicate the forecasting model providing predicted-return inputs and the optimization method applied.

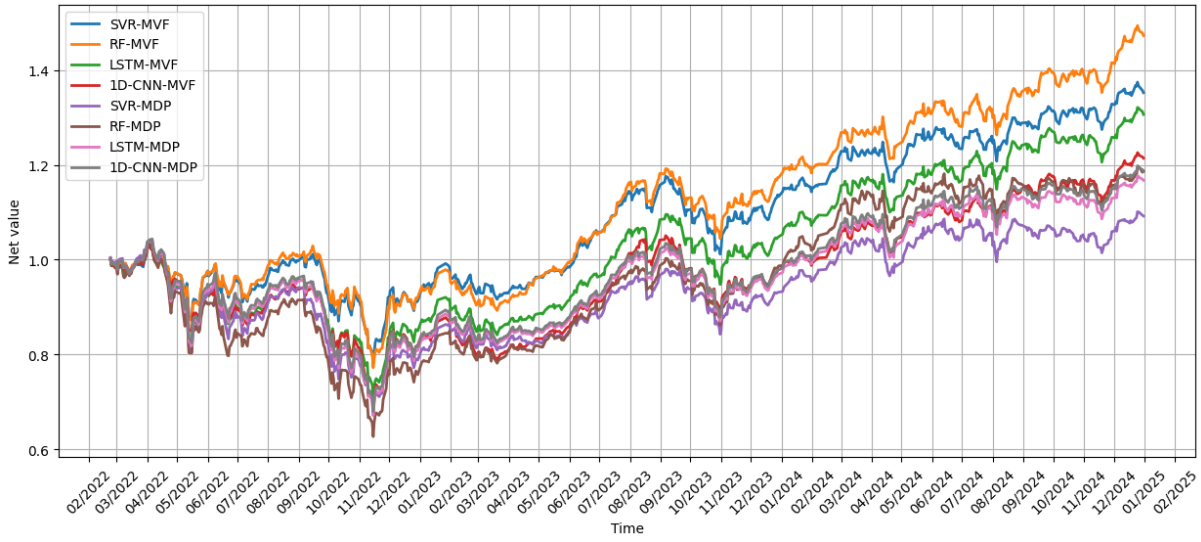
Table 6 shows that MVF portfolios generally outperform MDP portfolios in terms of mean MER, although this pattern is not uniform across all configurations. In the RF-based, LSTM-based, and 1D-CNN-based stock-selection groups, the MVF portfolios produce higher mean MERs than their corresponding MDP portfolios. Under SVR-based stock selection, RF-MVF records the highest mean MER (0.553%) and TR (19.364%), while LSTM-MVF achieves the highest information ratio (IR = 0.113), slightly exceeding RF-MVF (IR = 0.106). Across all configurations, RF-MVF under 1D-CNN-based stock selection achieves the highest total return (TR = 44.135%) and mean MER (1.261%), whereas RF-MVF under LSTM-based stock selection achieves the highest risk-adjusted performance (IR = 0.300), together with a mean MER of 1.212% and a TR of 42.429%. Overall, RF-MVF records the highest mean MER and TR within all four stock-selection groups and the highest IR in three of the four groups, indicating the most consistently favorable portfolio performance in this analysis.

Analysis on investment returns

Building on the performance evaluation in the previous section, attention is now turned to a detailed analysis of investment returns. The evolution of net asset values and MERs over the test period is examined to provide deeper insights into the risk-return dynamics across different portfolio configurations. Given its superior RMSE performance, as detailed in Table 5, the LSTM-based stock selection method was applied to provide the top 30 stocks. Figure 2 illustrates the trajectory of net asset values from 2022 to 2024, capturing the cumulative growth of different portfolios.

As shown in Figure 2, while all portfolios exhibit overall growth, those optimized using the MVF approach tend to achieve higher final net values. In particular, MVF-based portfolios that incorporate RF and LSTM models show strong cumulative performance, whereas portfolios employing the MDP approach display more moderate performance, with SVR-MDP yielding the lowest returns among them.

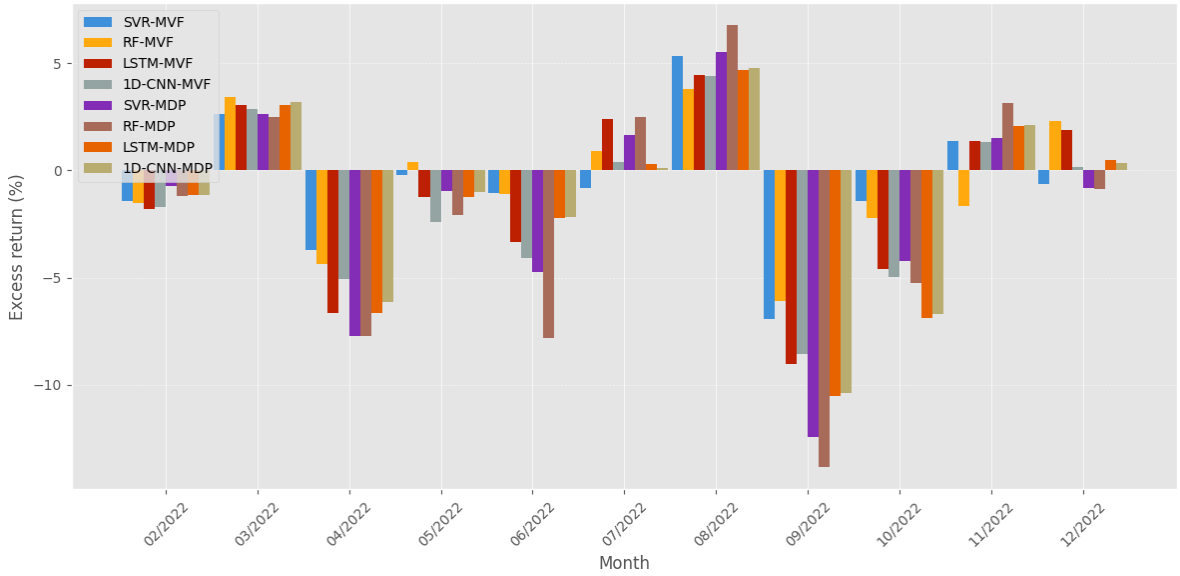
Figure 2
Net asset value trajectories of portfolios using the LSTM-based stock-selection set



Source: Author's work

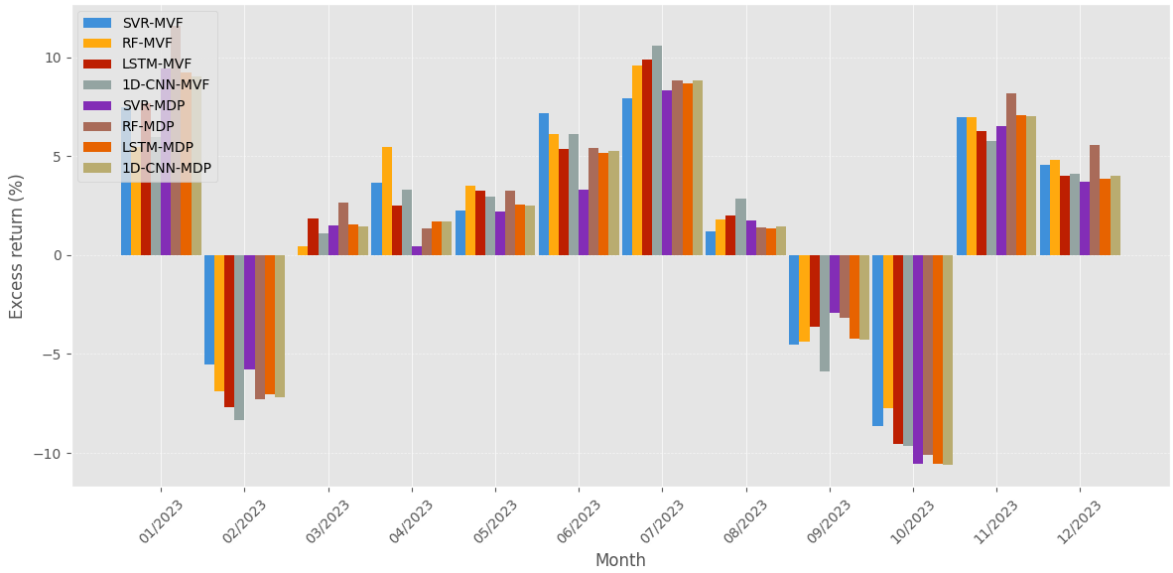
To analyze monthly performance in greater detail, Figures 3-5 display MERs for each year within the test period. Across these figures, MERs vary considerably across all eight portfolios, with no single model-portfolio combination consistently dominating on a monthly basis. However, in downtrend months, portfolios optimized using the MDP approach tend to record more pronounced losses compared to those optimized using the MVF approach. This observation suggests that while the MDP strategy is designed to limit drawdowns, it may be more susceptible to severe market declines, leading to larger negative excess returns during such periods. In contrast, MVF-based portfolios appear to experience relatively less severe losses during downtrends.

Figure 3
Monthly excess returns for 2022



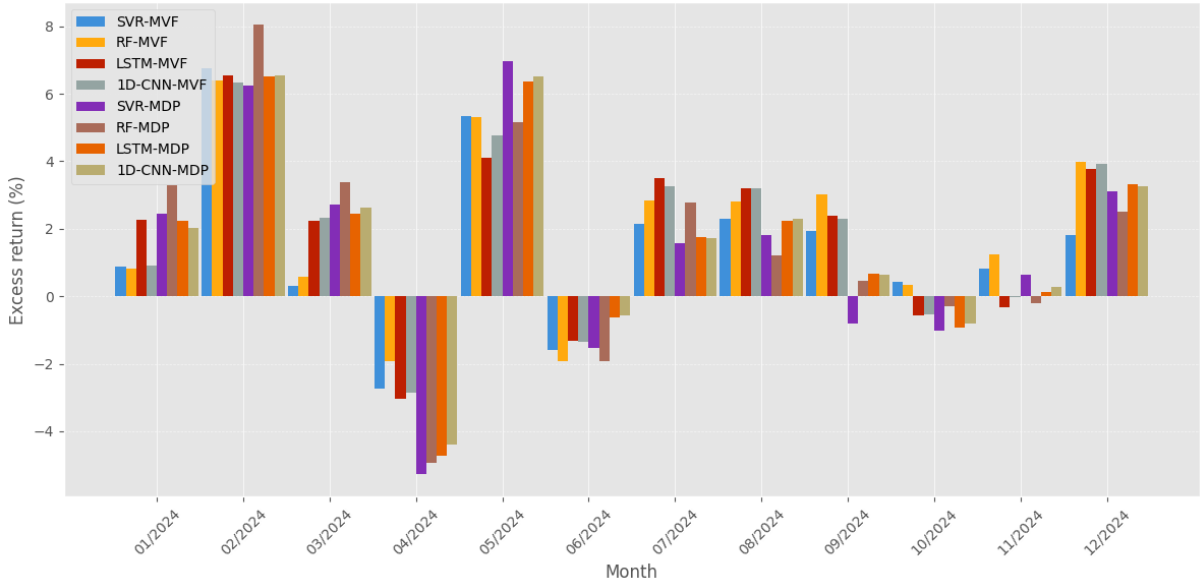
Source: Author's work

Figure 4
Monthly excess returns for 2023



Source: Author's work

Figure 5
Monthly excess returns for 2024



Source: Author's work

Based on the evaluation, MVF portfolios tend to outperform MDP portfolios overall in terms of net asset value growth and MER performance. Although MDP is designed to limit downside risk, our results suggest that, in this context, the MVF approach delivers superior returns without compromising volatility.

Conclusion

This study presents an integrated framework that combines advanced forecasting models with portfolio optimization techniques to enhance stock return prediction and portfolio construction. By employing four forecasting models (SVR, RF, LSTM, and 1D-CNN) alongside two portfolio optimization approaches (MVF and MDP), we developed eight distinct portfolio configurations: SVR-MVF, RF-MVF, LSTM-MVF, 1D-

CNN-MVF, SVR-MDP, RF-MDP, LSTM-MDP, and 1D-CNN-MDP. These portfolio types were evaluated across four stock-selection sets (SVR-based, RF-based, LSTM-based, and 1D-CNN-based), each consisting of 30 stocks selected according to the forecasting performance of the corresponding model. Our evaluation using MAE, MSE, and RMSE metrics revealed that LSTM achieved the lowest mean prediction errors across all three metrics, while 1D-CNN also demonstrated competitive forecasting performance. Building on these results, we applied a stock selection process based on RMSE performance to choose the top 30 stocks for each forecasting method. The portfolio results show that MVF generally provides stronger return performance than MDP, although this pattern is not uniform across all configurations. RF-MVF demonstrates the most consistently favorable performance, recording the highest mean monthly excess return and total return within all four stock-selection sets. In particular, RF-MVF under 1D-CNN-based stock selection achieves the highest overall total return of 44.135%, while RF-MVF under LSTM-based stock selection records the highest information ratio of 0.300. These findings suggest that portfolio outcomes depend not only on the optimization approach, but also on the forecasting model supplying predicted-return inputs and the stock-selection basis.

This study contributes to the growing body of research on hybrid forecasting-optimization frameworks, particularly in the context of emerging markets such as Vietnam. Nevertheless, several limitations warrant acknowledgment. First, the analysis is restricted to stocks from the VN-100 index and includes only continuously listed stocks without missing observations, which may limit generalizability to the broader Vietnamese equity market. Second, the study focuses on a single emerging-market context and a specific sample period; therefore, the relative performance of the forecasting and optimization methods may vary across other markets and market regimes. Third, the empirical framework considers only four forecasting models and two portfolio optimization approaches. Finally, the study does not examine broader robustness checks, such as alternative stock-selection criteria, additional risk measures, transaction costs, or dynamic rebalancing strategies. Future research may explore additional risk measures, dynamic rebalancing strategies, and further integration of alternative forecasting models to adapt to evolving market conditions.

References

1. Chekhlov, A., Uryasev, S., & Zabarankin, M. (2005). Drawdown measure in portfolio optimization. *International Journal of Theoretical and Applied Finance, 8*(1), 13–58. https://doi.org/10.1142/S0219024905002767
2. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054
3. Gao, H., Kou, G., Liang, H., Zhang, H., Chao, X., Li, C.-C., & Dong, Y. (2024). Machine learning in business and finance: A literature review and research opportunities. *Financial Innovation, 10*, Article 86. https://doi.org/10.1186/s40854-024-00629-z
4. Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies, 33*(5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009
5. Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction* (1st ed.). Springer. https://doi.org/10.1007/978-0-387-21606-5

6. Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry, 33*(1), 3–12. https://doi.org/10.1002/asmb.2209
7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
8. Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1–2), 307–319. https://doi.org/10.1016/S0925-2312(03)00372-2
9. Konno, H., & Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management Science, 37*(5), 519–531. https://doi.org/10.1287/mnsc.37.5.519
10. Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research, 259*(2), 689–702. https://doi.org/10.1016/j.ejor.2016.10.031
11. Li, W., & Law, K. L. E. (2024). Deep learning models for time series forecasting: A review. *IEEE Access, 12*, 92306–92327. https://doi.org/10.1109/ACCESS.2024.3422528
12. Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77–91. https://doi.org/10.2307/2975974
13. Markowitz, H. M. (1959). *Portfolio selection: Efficient diversification of investments*. John Wiley & Sons.
14. Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications, 42*(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040
15. Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *The Journal of Risk, 2*(3), 21–41. https://doi.org/10.21314/JOR.2000.038
16. Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing, 90*, Article 106181. https://doi.org/10.1016/j.asoc.2020.106181
17. Yu, J.-R., Chiou, W.-J. P., Lee, W.-Y., & Lin, S.-J. (2020). Portfolio models with return forecasting and transaction costs. *International Review of Economics & Finance, 66*, 118–130. https://doi.org/10.1016/j.iref.2019.11.002
18. Zhang, C., Sjarif, N. N. A., & Ibrahim, R. (2024). Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. *WIREs Data Mining and Knowledge Discovery, 14*(1), Article e1519. https://doi.org/10.1002/widm.1519

About the author

Minh Duc Nguyen is a lecturer at the Department of Economic Information Systems, University of Economics, Hue University, Vietnam. As a leading institution in economics and business education in Central Vietnam, the university fosters research and innovation in economic information systems, digital transformation, and financial technology. 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, finance and investment, machine learning and artificial intelligence, blockchain, legal informatics, knowledge management, and management information systems. His work focuses on applying data-driven and blockchain-based solutions to financial markets, risk management, legal compliance, and decision support systems to enhance investment strategies and business intelligence. The author can be contacted at **nguyenminhduc@hueuni.edu.vn**.