

# Average variance portfolio optimization using machine learning-based stock price prediction case of renewable energy investments

Karima Sabbar, Moad El Kharrim

Faculty of Economic and Social Legal Sciences of Tetouan, Abdemalek Esaadi University, Morocco

**Abstract:** With the progress of time series prediction, several recent developments in machine learning have shown that the integration of prediction methods into portfolio selection is a great opportunity to structure investment decisions in the renewable energy industry. In this paper, we propose a novel approach to portfolio formation strategy based on a hybrid machine learning model that combines a convolutional neural network (CNN) and long-term bidirectional memory (BiLSTM) with robust input characteristics obtained from Huber's location for stock prediction and the mean-variance (MV) Markowitz model for optimal portfolio construction. Specifically, this study first applies a prediction method for stock pre-selection to ensure high-quality stock inflows for portfolio formation. Then, the predicted results are integrated into the MV model. To comprehensively demonstrate the superiority of the proposed model, we used two portfolio models, the MV model and the equal-weighted (1/N) portfolio model, with LSTM, BiLSTM and CNN-BiLSTM, and used them as references. Between January 2016 and December 2021, historical data from the Stock Exchange of Thailand 50 Index (SET50) was collected for the study. Experience shows that integrating stock pre-selection can improve VM performance, and the results of the proposed method show that they outperform comparison models in terms of Sharpe ratio, average return and risk.

**Keywords :** portfolio optimization; mean-variance model; inventory forecasting; stock selection; machine learning; convolutional neural network; short-term long memory.

## 1 Introduction

In recent years, there has been a reluctance on the part of investors to invest in renewable energy technologies. Therefore, the most important prerequisite for assessing investment risks and creating a better situation to attract the tendency of investors to invest in this area is the optimization of portfolio selection to structure investment decisions in this area.

Portfolio optimization is one of the most interesting issues, the MV model relies on historical data to generate the optimal portfolio and can only display the optimal portfolio as far as data entry is concerned. Therefore, a number of researchers have applied machine learning to predict return and volatility in the future (Henrique et al. 2019). Investors in the financial market need to evaluate a variety of factors and perspectives to maximize their investment income (Rahiminezhad Galankashi et al. 2020). In this regard, including stock price prediction methods in portfolio optimization would be beneficial and cost-effective for investors (Kolm et al. 2014). Financial time series forecasting has long been a challenging area of study, as financial market fluctuations are inherently volatile, complex and dynamic (Paiva et al. 2019). However, several related studies claim that there is a pattern of asset price movement in financial time series data and that this pattern can be used to predict financial time series data to some extent (Wan et al. 2020; Wang et al. 2020).

Portfolio management is an analytical process of selecting and allocating a group of investment assets in which the allocated portion of the investment is constantly modified to optimize expected return and risk tolerance (Markowitz 1952). Markowitz's mean variance (MV) model, first developed in 1952, is the foundation of portfolio theory, which is widely used and recognized in portfolio management (Sharpe and Markowitz 1989).

However, based on the classical MV model, there are two main problems of practical application. The first is that the MV relies on the expected return and risk of asset inflows to produce optimal portfolios for each level of expected return and risk (Beheshti et al) to put into the optimization process, the MV model can achieve improved returns (Mitra Thakur et al 2018). Another problem is that many high-risk assets often return a large number of small-scale weightings into the optimal portfolio, making them difficult to implement, especially for individual investors (Ben Salah et al. 2018; Ortiz et al. 2021; Huang et al. 2021).

The main objective of this study is to develop a portfolio formation approach for individual investors in which a hybrid machine learning model that combines a convolutional neural network and bidirectional short-term memory with robust input characteristics (R-CNN-BiLSTM) is applied to predict future closing stock prices before using the MV model to build the optimal portfolio. In this regard, this study proposes a new approach to portfolio formation that combines R-CNN-BiLSTM and MV (R-CNN-BiLSTM+MV1). The article is organized as follows. In Section 2, reviews of some existing studies are discussed regarding inventory forecasting and portfolio optimization, as well as empirical work that uses traditional statistics and machine learning methods to solve problems related to inventory forecasting and selection. Section 3 briefly explains the underlying knowledge used in this study. Section 4 presents the

detailed experimental process. Section 5 reports the experimental results. Finally, Section 6 discusses the main findings of the work, theoretical implementations, and limitations.

## 2 Basic knowledge

### 2.1 Mean-variance optimization

Markowitz (1952) proposed the mean-variance (MV) model and was awarded the Noble Prize in Economics in 1990. The MV model used mean and variance, which are calculated from historical asset prices to quantify the expected return and risk of the generated portfolio. The MV model assumes that the investor wants to either maximize the expected return for a given level of risk or minimize the risk for a given return (Kolm et al. 2014). However, in this study we only show optimization with minimal variance. The MV model is described as follows:

Minimize:

$$\sigma^2 = \sum_{i=0}^N \sum_{j=0}^N w_i w_j C_{ij} \quad (1)$$

$$\text{Subject to: } \gamma = \sum_{i=1}^N w_i E_i \quad (2)$$

$$\sum_{i=1}^N w_i = 1 \quad (3)$$

$$w_i \geq 0 \quad (4)$$

where  $N$  is the total number of assets, which indicates the dimensionality of optimization in the portfolio.  $w_i$  is the weight of each  $i$  asset in the portfolio to be optimized;  $\sigma^2$  represents the variance of the portfolio which generally refers to the risk of the portfolio;  $C_{ij}$  is the covariance of the return between assets  $i$  and  $j$ ;  $\gamma$  is the expected or target performance; and  $E_i$  is the average return on an individual asset  $i$ .

### 2.2 CNN

A convolutional neural network (CNN) is a kind of deep learning model for processing grid model data, such as image processing and natural language processing. CNN can be applied to predict time series data (Sadouk 2019). CNN can significantly improve the quality of learning models by reducing the number of parameters. CNN is mainly composed of three types of layers: a convolution layer, a clustering layer, and a fully connected layer (Albawi et al. 2017). The first two layers, the convolution layer and the grouping layer, perform feature extraction, while the last layer, the fully connected layer, directs the extracted features to the output (Milošević and Racković 2019).

### 2.3 LSTM

Long-term short-term memory (LSTM) was proposed by Hochreiter and Schmidhuber (Hochreiter and Schmidhuber 1997). The model is a class of RNN but has a memory function, which allows LSTM to recycle data over a long period of time compared to RNN (Fischer and Krauss 2018). The LSTM model filters the information that enters through door structures consisting of an entrance door, a forgotten door, and an exit door to improve and maintain memory cells. LSTM is particularly popular in the field of financial time series forecasting, as the model can effectively manage the redundancy of historical data (Gao et al. 2021). The operating equation of LSTM is as follows:

$$\text{Forget gate : } \quad ft = (w_f \sigma_f[ht-1, xt] + bf) \quad (5)$$

$$\text{Input gate : } \quad it = \sigma(w_i[ht-1, xt] + bi) \quad (6)$$

$$Ot = (w_o \sigma_o[ht-1, xt] + b_o) \quad (7)$$

$$\tilde{C}_t = \tanh(w_c[ht-1, xt] + bc) \quad (8)$$

Output gate :

$$ct = ft \cdot ct-1 + it \cdot \tilde{c}_t \quad (9)$$

$$ht = Ot \cdot \tanh(ct) \quad (10)$$

where  $f_t$ ,  $i_t$  and  $O_t$  denote respectively the forgotten door, the entrance door and the exit door;  $w$  is the weight of the matrix;  $b_t$ ,  $b_i$  and  $b_o$  indicate the polarization of the forgetting door, the entrance door and the exit door, respectively;  $\sigma$  means the sigmoid function;  $x_t$  and  $h_t$  designate respectively the input and output of current at time  $t$ ; This is the value of the entrance gate at time  $t$ ; and the hyperbolic function (tangent) is represented by  $\tanh$ .

## 2.4 BiLSTM

Long-term bidirectional memory (BiLSTM) is an improved version of LSTM with the ability to access the front and rear directions of the input feature (Dong et al. 2014). The main difference between BiLSTM and LSTM is that it uses two hidden layers. BiLSTM was found to be better than LSTM in terms of predicting time series data (Siami-Namini et al. 2019). The output of the hidden layer of BiLSTM has the activation function both forward and backward. The BiLSTM equations (Yang and Wang 2022) are described as follows:

$$\vec{h}_t = \sigma(W_{xh} \rightarrow x_t + W_{\vec{h}\vec{h}} \vec{h}_{t-1} + b_{\vec{h}}) \quad (11)$$

$$\overleftarrow{h}_t = \sigma(W_{xh} \leftarrow x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \quad (12)$$

$$H_t = W_{xh} \rightarrow \vec{h} + W_{\vec{h}y} \vec{h} + b_y \quad (13)$$

where  $\sigma$  represents the function for activating the model.  $W$  is the weight of the matrix;  $W_{xh}$  is the weight of the input ( $x$ ) in the hidden layer ( $h$ );  $H_t$  indicates the entrance of the hidden layer; and  $b_x$  denotes the polarization of the respective gates ( $x$ ). The output is carried out by updating forward and backward structures.  $\vec{h}_t, \overleftarrow{h}_t$

## 2.5 Robust statistics

In real-world applications, data collection often includes atypical observations that deviate from the majority or mass of data, in which these observations are referred to as outliers, especially in financial time series data. For example, for stock prices, outliers deviate from the general trend in the data. Therefore, it is very difficult to predict future stock prices.

To overcome these limitations, this study uses a robust statistical theory (Maronna et al. 2019) to estimate the appropriate dataset to use in the machine learning training process.

### 2.5.1 The Classic Robust Location Estimator

The mean and median of the sample are considered location estimators of the data distribution. The main difference is that the sample mean is not robust to extreme outliers. For example, the closing price of a certain stock in a week falls immediately for unexpected reasons. For example, provided that the historical closing prices of a certain stock are {243, 190, 150, 80, 56, 28, 142}, in this case, the sample average is 125, which is not considered a good location estimator for these observations. Then the median of the sample is 150, which is a robust data location estimator. However, if the distribution of data is considered approximately normal, then the sample mean would be considered a better estimator than the sample median. The robust location estimator is a combination of these two classical estimators. To put it precisely, when there are extreme outliers in observations, the robust estimator approaches the median of the sample. On the other hand, the estimated location is close to the sample average.

### 2.5.2 Huber Location Estimator Tool

Huber (1964) proposed a good combination of mean and median, called a robust location estimator or M estimator of a location, which can be described as follows:

$$\hat{\mu} = \min_{\mu} \sum_{i=1}^n \rho(x_i - \mu) \quad (14)$$

where  $\hat{\mu}$  is a robust location estimator of the observation;  $x_i$  is the observation variable  $I$ ;  $\rho$  represents the error function. Robust location is a parameter that minimizes function to ensure that the parameter provides the minimum error between the location estimator and all observations. Several methods have been proposed (Maronna et al. 2006) to find the local minimum of the function such as the maximum likelihood estimator (MLE). In this paper, we use a numerical method, the Newton-Raphson method, to find the robust location estimator.  $\rho, \mu, \rho, \rho$

The Newton-Raphson method is an iterative method for solving nonlinear equations. To solve the equation  $h(y) = 0$ ,  $h$  is set to be linearized at each iteration. In a location M-estimator, it is necessary to solve the equation

Iterations are defined as follows:  $h(\mu) = 0$  for  $h(\mu) = \arg \{ \Psi (x - \mu) \}$

$$\mu_{m+1} = \mu_m + \frac{\sum_{i=1}^n \Psi(x_i - \mu_m)}{\sum_{i=1}^n \Psi'(x_i - \mu_m)} \quad (15)$$

Where is the value of the location estimator at iteration  $\mu_m$ .  $x$  is defined by the following function with respect to a given positive constant of  $k$  as follows:

$$\Psi_k(x) = \begin{cases} k, & x > k \\ x, & -k \leq x \leq k \\ -k, & x < -k \end{cases} \quad (16)$$

if is bounded, its derivative tends to be infinitely zero (Hampel et al. 2011). If  $\Psi_k k$  tends to infinity, then is the mean. On the other hand, if  $\Psi_k k$  tends to zero, then the acts as the median. In this article, we use  $\Psi_k k = 1.435$  in our proposed method. This value is similar to that used by Fox and Weisberg (2019).

### 3. Experimental process

#### 3.1. Data preparation

One of the biggest challenges in forecasting inventories is capturing the financial data schema of the time series between the past and the future (Wang et al. 2020). As a result, it is easier to forecast stable inventories than volatile stocks. Wang et al. (2020) randomly select 21 FTSE100 stocks as a sample for the machine learning prediction process before optimization. Chen et al. (2021) randomly selected 24 stocks from the SSE50 as candidate assets in the stock prediction process before forming a portfolio. Ma et al. (2021) used 49 stocks from the SSE100 as a dataset for predicting stocks using machine learning before building a portfolio. In addition, many researchers agree to hold about 10 different stocks in the portfolio. For example, Soeryana et al. (2017) chose five different stocks from the optimal portfolio. Abrami and Marsoem (2021) built a portfolio of eight assets. Therefore, our study randomly selected 25 stocks that fully traded between January 1, 2015 and December 30, 2020 covering 1462 trading days from the SET50 index and used the closing price as an experimental dataset, large enough for individual investors to build a portfolio. (Zaimovic et al. 2021).

Table 1 provides summary closing price statistics for 25 stocks. The stocks with the highest and lowest returns are clearly Delta and IRPC, while the stocks with the highest returns and the lowest standard deviation are SCC and LH, respectively.

**Table 1.** Summary statistics for experimental data.

Stock	Maximum	Minimum	Average	Standard deviation
AOT	81	25,4	52.74	15.72
COM7	27.25	18.5	22.26	2.27
HMPRO	12	4.07	7.8	2.06
BJC	66	27.22	44.42	9.12
BTS	14.2	7.85	9.79	1.53
CPALL	90	37.5	63.81	13.46
CPN	86.25	33.25	60.99	13.53
DELTA	684	30	81.12	44.62
DTAC	96.25	27.75	48.93	15.18
SHE	69.5	19.1	36.46	11.48
GLOBAL	19.61	6.33	13.24	3.58
INTUCH	83.5	43	59.31	8.8
IRPC	8.15	1.88	4.77	1.38
IVL	62.5	16.9	36.27	12.24
KCE	64.5	12	34.	12.31

KTC	60	6.22	23.28	13.15
LH	12.2	7	9.53	1.26
AS	45.25	14.49	34.28	6.63
MTC	68.25	11	36.75	15.01
PTT	58.8	19.8	39.54	8.32
PTTEP	160	42.5	101.83	23.64
PTTGC	103	24	63.98	14.4
RATCH	81.5	46	57.01	5.73
SAWAD	80.75	19.49	42.52	11.55
SCC	550	267	454.1	59.68

### 3.1.1. R-CNN-BiLSTM Architecture

This study proposes a hybrid R-CNN-BiLSTM model to improve prediction accuracy. The proposed model consists of three parts: data transformation, feature extraction, and price forecasting. First, the data transformation component converts the closing prices of stocks into the robust domain, which is the non-noisy version of the data. In this study, direct stock closing price data is not suitable for machine learning training due to the high standard deviations. Therefore, we need to transform data to make it more relevant to the training process. The closing prices of the shares are divided into a small time series of 4 days, called latency.

The latency times overlap 1 day with each other. The Huber location estimator of each latency is calculated using equations (14) and (15).

Secondly, feature extraction is performed using a CNN network. CNN has the ability to identify important factors in the data, called "characteristics." The purpose of this step is to preserve historical data in time series data and feed it into BiLSTM. Therefore, input data is converted by performing convolution operations on the time series data time steps using a sequence aliasing layer. In the next step, the two-dimensional convolutional layer is used to extract characteristics from the data. The filtering size of the first convolutional layer is  $3 \times 3$  and the stride parameter is set to  $\{a = 1, b = 1\}$ , where a is a vertical step size and b is a horizontal step size.

The first convolutional layer is followed by Batch Normalize (BN), which normalizes a mini batch of data across all data points to speed up the CNN formation process. In our model, the exponential linear unit (ELU) is used as the activation function. This function performs the identity operation on positive inputs and exponential nonlinearity on negative inputs. The filtering size of the second and third convolutional layers increases to  $5 \times 5$  and  $7 \times 7$ , respectively. The next layer is the grouping layer. It subsamples by dividing the data into sub-regions and then calculating the maximum for each region.

The sequence structure of the input is restored using the sequence unfold layer. Finally, the spatial dimensions of the data are reduced with the flattened layer. An overview of the proposed framework for the CNN model is shown in Figure 1.

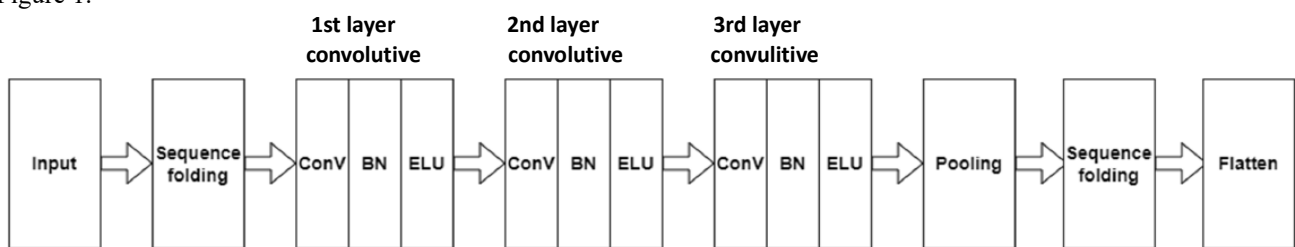
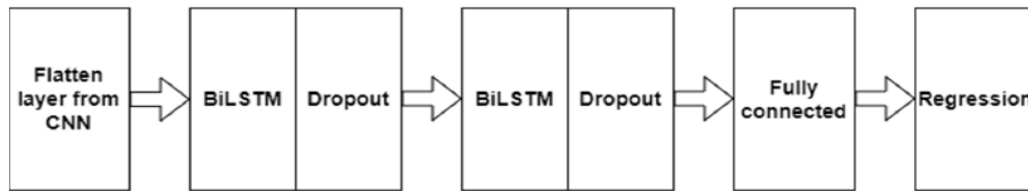


Fig.1. The framework of the CNN model

Finally, price forecasting using BiLSTM is performed. CNN's flattened data is used as input to BiLSTM. The input size of the first BiLSTM layer is 500 and the number of hidden units is 128. Tanh is used as the state activation function and Sigmoid as the door activation function. To prevent the network from overadapting, we place the abandonment layer next to the BiLSTM layer. Its operation changes the underlying network architecture between interactions by randomly setting input elements to zero with uniform probability. For the second BiLSTM layer, we decrease the input size and the number of hidden units to 256 and 16, respectively. The last two layers are fully connected and regression layers are the standard of the CNN architecture. The BiLSTM framework is shown in Figure 2.

1st layer BILSTM                      2nd layer BILSTM layer



**Fig. 2.** The framework of the BiLSTM model.

### 3.1.2. Training and testing process

One of the most important factors that determine the success of machine learning is the process of training and testing. In this study, we divided the closing price of each chosen stock into training and test sets according to the ratio of 80:20. Therefore, the first 1201 days of data are used in the training process and the last 262 days are used as a test set.

### 3.1.3. Hyperparameter parameter

The training data set is transmitted to the proposed training model. In this step, the different hyperparameters of the neural network are specified. These include the number of hidden layers, the number of eras, and the size of batch entries. Finding optimal hyperparameters remains a major challenge in the field of deep learning. In this study, hyperparameters are defined manually by trial and error with the selection of the best parameters of the experiment. The following is a detailed description of hyperparameters and their value parameters.

1. Number of eras: An era is a complete training cycle. In our experiments, we set the number of eras at 100 and carried out our training. After training, we found that all workouts stop at a maximum of 100 to 120 eras. Therefore, 100 is selected as the value for this hyperparameter.
2. The number of hidden layers: This is the number of layers between the input and output layers. For the CNN network, we set the number of hidden convolutional layers to 100, 100 and 50. In the BiLSTM network, we have set these numbers to 128 and 16.
3. Learning rate: This value is defined for the precise convergence of the model in the prediction. In our experiment, we set an apprenticeship rate at 0.0001. Many researchers recommend using a learning value below 0.01 (Hastie et al. 2017).
4. Optimizer: This is the optimization function used to get the best results. In our work, we use the Adam optimizer, as it works well for LSTM-based networks.
5. Loss function: The mean square error (RMS) was used as the loss function. Our implementation was written using MATLAB with GPU compute.

### 3.1.4. Selection of actions

Once all stock prices are successfully predicted, high-quality stocks are selected to perform the optimization process one by one by ranking them in descending order based on expected (average) return. The expected share price is used to calculate the return on the stock using equation (17).

$$\hat{R}_t = \frac{\hat{p}_t - \hat{p}_{t-1}}{\hat{p}_{t-1}} \quad (17)$$

where is the return of the stock  $\hat{R}_t$  at time  $t$ , while is the expected share price at time  $t$  and is the expected share price at time  $\hat{p}_{t-1}$ .

As a result, we select the top number (N) of stocks with a higher potential return in ranking order. Only the selected securities are qualified to build the portfolio in the next step. The MV model is used in this process to build the optimal portfolio with different proportions of asset allocation depending on the qualified stocks. The optimization process is carried out using the MS Excel solver in which the minimum variance is defined as the objective function and the weight of each asset is adjusted using the Excel solver. Therefore, each of the optimal portfolios with the lowest variance is found and used for analysis.

## 4. Experimental results

This section first presents the prediction performance of the LSTM, BiLSTM, CNNBiLSTM and R-CNN-BiLSTM models. In the following, this study constructs different portfolio sizes using the classic MV model to compare the prediction result of different machine learning models without transaction fees.

## 5.1. Prediction performance results

### 5.1.1. Machine Learning Metrics

In this section, the predictive accuracy of machine learning models is evaluated by three criteria, mean absolute error (MAE), mean square error (MSE), and mean absolute error in percentage (SMAPE), as they are widely used as performance measures (Jierula et al. 2021; Singh et al. 2021). These measures are described as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{p}_i - p_i)^2 \quad (18)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{p}_i - p_i| \quad (19)$$

$$SMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|\hat{p}_i - p_i|}{|p_i| + |\hat{p}_i|} \quad (20)$$

where  $\hat{p}_i$  refers to the predicted price,  $p_i$  represents the actual value, and  $\hat{p}_i p_i n$  indicates the total number of shares used in the experiment.

### 4.1.2. Prediction performance

Tables 2 and 3 present the results of each model that was applied in accordance with the performance measures used. From both tables, it is clear that the R-CNN-BiLSTM model provides most of the best results compared to the other models in the experiment. However, there are still a few exceptions in which some comparison models work better. For example, the prediction error of BTS, LH, PTTGC and SAWAD shares in terms of BiLSTM's MAE, MSE and SMAPE is lower than that of R-CNN-BiLSTM. Another example is that the EAWs, MSEs and SMAPE in the SCC stock that have been predicted using CNN-LSTM are smaller than R-CNN-BiLSTM.

**Table 2.** Comparison of prediction performance between LSTM and BiLSTM.

Stock	LSTM	BiLSTM				
	MAE	MSE	SMAPE	MAE	MSE	SMAPE
AOT	1.8492	6.0365	1.4947	1.3972	3.8694	1.1617
COM7	0.3848	0.2955	0.9173	0.3652	0.2743	0.8640
HMPRO	0.2085	0.0845	1.1439	0.2095	0.0897	1.1524
BJC	0.9812	1.7666	1.2952	1.1242	2.1052	1.4813
BTS	0.2837	0.1599	1.2865	0.2313	0.1095	1.0495
CPALL	0.9605	1.6107	0.7317	0.9228	1.5193	0.7042
CPN	1.8462	7.1009	1.9229	1.7676	6.5828	1.8486
DELTA	4.0368	8.0218	13.123	3.8862	7.7699	12.226
DTAC	0.9254	1.5075	1.1961	0.9695	1.5263	1.2567
SHE	1.0341	2.0658	1.2442	1.0010	1.9209	1.2035
GLOBAL	0.4385	0.3665	1.5382	0.4237	0.3466	1.4796
IRPC	0.7735	0.6996	1.3212	0.6152	0.4566	1.0894
INTUCH	0.7602	1.3049	0.7142	0.7457	1.2532	0.7002
IVL	1.1677	2.4750	2.2526	1.1137	2.1696	2.1337
KCE	1.3583	2.9136	3.0482	1.2504	2.6176	2.9376
KTC	1.6230	7.4166	2.0531	1.5991	7.5624	2.0145
LH	0.4567	0.3537	3.0500	0.3830	0.2645	2.5767
AS	4.2719	2.5754	9.3568	3.5595	1.8810	7.9947
MTC	2.9454	1.3472	2.7505	2.2480	8.7324	2.1013
PTT	0.9246	2.2412	1.2935	0.9153	2.0173	1.2794
PTTEP	2.6926	2.1457	1.5534	2.5247	2.0952	1.4822
PTTGC	4.7754	4.1757	5.5750	3.2552	2.5053	3.9878

RATCH	1.0472	2.4033	0.8817	1.3077	3.1643	1.0816
SAWAD	3.8613	3.1299	3.2622	3.5476	2.5694	2.9785
SCC	3.6151	1.7271	4.9859	2.6922	1.0819	3.7868

Mean absolute error (AEM): As shown in Tables 2 and 3, the mean AEM value for each of the machine learning models is decreasing as follows: 1.7219 for LSTM, 1.5350 for CNN-BiLSTM, 1.5222 for BiLSTM, and 1.4582 for R-CNN-BiLSTM. Stock PTTGC has the highest EAW of 4.7754, which is found for the LSTM model. The lowest EAW is the HMPRO stock, which was predicted using CNN-BiLSTM, with a value of 0.1651. RMS: According to Tables 2 and 3, the mean ESM values for each of the machine learning models are reported as follows: 2.9000 for CNN-BiLSTM, 2.5794 for BiLSTM, 2.5570 for LSTM and 1.8081 for R-CNN-BiLSTM. The largest EAW is 9.3412, which is found on the MTC stock generated from CNN-BiLSTM. Using R-CNN-BiLSTM in HMPRO stock, the MAE minus of 0.0523 was predicted. Average absolute percentage error (SMAPE): From Tables 2 and 3, the mean values for each of the machine learning models are described from top to bottom as follows: 2.7197 for LSTM, 2.4589 for CNN-BiLSTM, 2.4229 for BiLSTM and 2.3332 for R-CNN-BiLSTM. The largest SMAPE is 13.487 which is associated with the predicted DELTA stock using CNNBiLSTM. The lowest SMAPE is found on the CPALL stock, model R-CNN-BiLSTM, with a value of 0.5713.

In conclusion, most of the R-CNN-BiLSTM results outperform the LSTM, BiLSTM and CNN-BiLSTM models for the inventory forecasting process in terms of EAW, MSE and SMAPE. Specifically, 14 stocks, HMPRO, BJC, CPALL, CPN, DELTA, EA, GLOBAL, IVL, KCE, KTC, MINT, MTC, PTT and PTTEP, which were predicted using R-CNN-BiLSTM, scored best on all three measures, followed by BiLSTM and CNN-BiLSTM. What's more, a unique traditional machine learning model, LSTM, is the worst, with several predictive errors in this experiment. Specifically, only the original RATCH is the most efficient in terms of MAE and SMAPE for the LSTM model. It can be seen that the proposed model, R-CNN-BiLSTM, which uses robust entry characteristics instead of the direct closing price of stocks in the machine learning training process, achieves a majority of better results than machine learning models that use the direct closing price of stocks. Seize.

**Table 3.** Comparison of prediction performance between CNN-BiLSTM and R-CNN-BiLSTM.

Stock	LSTM	BiLSTM				
	MAE	MSE	SMAPE	MAE	MSE	SMAPE
AOT	1.3178	3.3801	1.0902	1.3700	3.1541	1.1104
COM7	0.4277	0.3319	1.0172	0.3292	0.1934	0.7758
HMPRO	0.1651	0.0585	0.9034	0.1747	0.0523	0.9577
BJC	1.0290	1.8592	1.3327	0.8566	0.8566	1.1343
BTS	0.3006	0.2078	1.3639	0.2627	0.1243	1.1716
CPALL	1.0490	2.0394	0.7842	0.7528	1.0157	0.5713
CPN	1.8837	6.6585	1.9341	1.2533	2.8825	1.3019
DELTA	4.1104	8.1768	13.487	3.4162	5.0508	12.143
DTAC	1.1074	1.9661	1.4468	0.8781	1.2819	1.1335
SHE	0.9578	1.6883	1.1501	0.8770	1.3480	1.0603
GLOBAL	0.3636	0.2451	1.2451	0.3325	0.2011	1.1091
IRPC	0.7817	1.2752	0.7347	0.7434	1.1314	0.6960
INTUCH	0.7479	0.7327	1.2883	0.9645	1.0976	1.5837
IVL	0.9090	1.5083	1.7234	0.8079	1.0826	1.5329
KCE	1.2586	2.5306	2.8828	0.7679	0.9975	1.7225
KTC	1.8072	8.4518	2.3359	0.9608	2.2998	1.2831
LH	0.4215	0.3194	2.8259	0.4915	0.3743	3.2514
AS	3.1798	1.5235	7.2280	3.1251	1.3556	7.0846
MTC	2.2111	9.3412	2.0425	1.8174	6.4816	1.6748
PTT	0.7665	1.2902	1.0616	0.6891	0.9916	0.9468
PTTEP	2.0733	1.1074	1.1690	2.0274	1.0305	1.1313
PTTGC	4.7207	4.3939	5.5390	5.1528	4.9323	5.9606
RATCH	1.1446	2.8712	0.9626	1.0720	2.0291	0.8916
SAWAD	3.5888	2.8325	2.9873	3.6310	3.0988	3.0133



SCC	2.0515	7.7104	2.9372	3.7010	1.8256	5.0882
-----	--------	--------	--------	--------	--------	--------

## 4.2. Results of portfolio optimization

### 4.2.1. Portfolio metrics

In this section, the performance of different optimal portfolios is measured and compared using three criteria, the Sharpe ratio, the average return and the risk of the portfolio. These metrics are widely used to evaluate and compare the performance of equity portfolios (Lefebvre et al. 2020; Sikalo et al. 2022; Mba et al. 2022). Another measure of the portfolio is the Sharpe ratio, which can be described as follows:

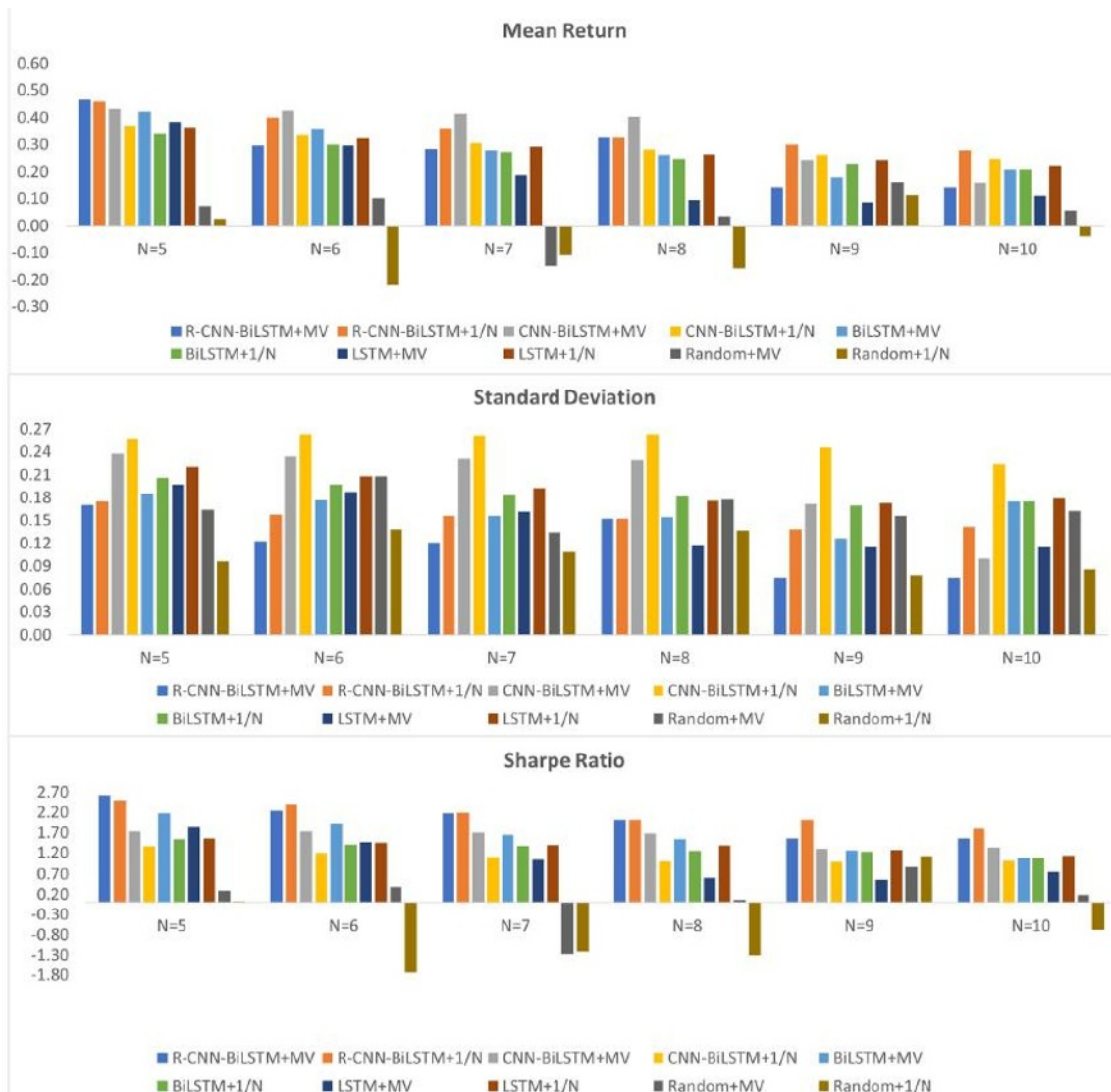
$$\text{Ratio de Sharpe} = \frac{E_p - R_f}{\sigma} \quad (21)$$

where  $E_p$  is the expected (average) return or average return of the portfolio;  $\sigma$  is the standard deviation or risk of the portfolio; and  $R_f$  refers to risk-free assets. In this study, we use a risk-free asset rate of 0.022, based on the Thai 10-year Treasury rate.

### 4.2.2. Performance of portfolios of different sizes

Numerous studies have shown that owning too many different stocks makes them difficult to control and manage, especially for individual investors. Several studies related to portfolio optimization consider building a portfolio with less than 10 stocks (Almahdi and Yang 2017). Paiva et al. (2019) found that a portfolio with an average of seven stocks outperforms other portfolios with a different number of stocks. Wang et al. (2020) showed that the optimal portfolio with ten stocks performs better than a portfolio with other numbers of stocks. Chen et al. (2021) argued that having seven stocks in a portfolio is the most appropriate number for portfolio formation. As a result, this study decides to build portfolios corresponding to the number of shares  $N = 5, 6, 7, 8, 9$  and  $10$  and to comprehensively evaluate the performance of the proposed models. The annualized average return, annualized standard deviation and annualized Sharpe ratio are used as indicators. Figure 4 shows annualized returns for different portfolio sizes  $N = 5, 6, 7, 8, 9$  and  $10$  are presented in three sub-graphs in which the y-axis of each sub-graph shows the amount of the average return, the standard deviation and the Sharpe ratio calculated annually, while the X axis of each sub-graph represents the different models. which are formed by a different number of actions. It can be clearly seen in Figure 4 that when the number of  $N = 5$ , R-CNN-BiLSTM + MV is the best performer in terms of average yield, standard deviation and Sharp ratio. Specifically, when  $N = 5$ , R-CNN-BiLSTM+MV has the highest average efficiency of 0.47, followed by 0.46 for R-CNN-BiLSTM+1/N, 0.43 for CNNBiLSTM+MV, 0.42 for BiLSTM+MV, 0.38 for LSTM+MV, 0.37 for CNN+BiLSTM+1/N, 0.36 for LSTM+1/N, 0.34 for BiLSTM+1/N, 0.07 for Random+MV and 0.02 for Random+1/N. In addition, when  $N = 9$  and  $10$ , R-CNN-BiLSTM+MV outperforms other models in terms of standard deviation. Specifically, R-CNN-iLSTM+MV has the lowest standard deviation of 0.07 for  $N = 9$  and  $10$ . Regarding the measurement of the Sharpe ratio, when  $N = 5, 8$ , R-CNN-BiLSTM+MV provides the best Sharpe ratios of 2.62 and 1.99, respectively. However, R-CNN-BiLSTM+1/N also has the same Sharpe ratio as R-CNN-BiLSTM+MV when  $N = 8$ .

In summary, a clear advantage of R-CNN-BiLSTM+MV is found with a portfolio size  $N = 5$  in which all three indicators outperform the other models except for the risk of Random+MV and Random+1/N; however, these two models have low expected returns of 0.07 and 0.02, respectively, but it is still reasonable to consider R-CNN-BiLSTM+MV superior. In addition, most models tend to perform better in terms of annualized expected return, annualized Sharpe ratio, and standard deviation or annualized risk when the number of stocks in the portfolio is five.



**Fig. 3.** Annualized portfolio performance for different portfolio sizes

## 5. Discussion

The article aims to expand the existing literature on portfolio optimization with stock selection. The proposed prediction model is developed based on the use of robust statistical theory and the CNN-BiLSTM machine learning model to advance the MV model, which integrates the benefits of machine learning into stock selection. This study has several results.

First, this article compares the predictive performance of LSTM, BiLSTM, CNN BiLSTM and inventory forecasting. The experimental results show that BiLSTM is superior to other models, indicating that it is more suitable for predicting financial time series than the other machine models applied in this experiment, confirming the study by Wang et al. (2020) showing that this traditional LSTM was superior in terms of prediction performance. Second, this study improves the predictive accuracy of CNN-BiLSTM by transforming the closing price of shares into a robust entry feature that can effectively reduce prediction error before the model predicts the future price. After comparing the results of the R-CNN-BiLSTM to LSTM, BiLSTM and CNN-BiLSTM, it was discovered that the robust input is appropriate to be used as an input feature for the machine learning training process to capture time series financial data that can overcome other comparison models when the direct closing price of shares is used as an input characteristic.

Finally, the result of the prediction process is integrated into the stock selection for portfolio optimization; Stocks with higher returns calculated from predicted prices are chosen to build the optimal portfolio. The experimental results show that holding five shares is appropriate and realistic for individual investors, which is different from the results of Wang et al. (2020) and Chen et al. (2021). In addition, most of the results of R-CNN-BiLSTM+MV, R-CNN-BiLSTM+1/N, CNN-BiLSTM+MV, CNN-BiLSTM+1/N, BiLSTM+MV, BiLSTM+1/N, LSTM+MV and LSTM+1/N are superior to Random+MV and Random+1/N in terms of Sharpe ratio, average return and standard deviation,

indicating the importance of selecting high-quality stocks in portfolio optimization. The importance of stock preselection is similar to the findings of Wang et al. (2020), Ta et al. (2020), and Chen et al. (2021).

## 6. Conclusion

This study enriches theoretical research on stock price forecasting and portfolio optimization. This paper uses four prediction models, which can capture financial time series data to ensure high-quality assets before starting portfolio optimization. Specifically, LSTM, BiLSTM, CNN-BiLSTM and R-CNN-BiLSTM are adopted to predict the daily future closing price of the stock and compare R-CNNBiLSTM's forecast results with LSTM, BiLSTM and CNN-BiLSTM to show the predictability of R-NNBiLSTM using robust data instead of the direct closing price of the shares to more accurately predict financial time series data.

While this study provides useful information, there are some limitations. First, we only use stock data in Thailand. Due to economic differences between countries, this method may not be suitable for stock markets in other countries. Second, several external factors have an impact on the financial market and can be added as input indicators to improve the method, such as the COVID-19 crisis, interest rates and politics. Third, this study does not consider temporal complexity as a constraint for comparing results. Finally, since this study manually defines hyperparameters based on trial and error, the application of hyperparameter optimization algorithms can provide better hyperparameters. In future research, temporal complexity should be considered to further demonstrate the applicability of the proposed method.

## References

1. Le Caillec, Jean-Marc, Alya Itani, Didier Guriot, and Yves Rakotondratsimba. 2017. Stock picking by probability–possibility approaches. *IEEE Transactions on Fuzzy Systems* 25: 333–49.
2. Lefebvre, William, Gregoire Loeper, and Huyen Pham. 2020. Mean-variance portfolio selection with Tracking Error Penalization. *Mathematics* 8: 1915.
3. Li, Ting, Weiguo Zhang, and Weijun Xu. 2015. A fuzzy portfolio selection model with background risk. *Applied Mathematics and Computation* 256: 505–13.
4. Lozza, Sergio Ortobelli, Enrico Angelelli, and Daniele Toninelli. 2011. Set-portfolio selection with the use of market stochastic bounds. *Emerging Markets Finance and Trade* 47: 5–24.
5. Ma, Yilin, Ruizhu Han, and Weizhing Wang. 2021. Portfolio optimization with return prediction using Deep Learning and machine learning. *Expert Systems with Applications* 165: 113973.
6. Markowitz, Harry. 1952. Portfolio selection\*. *The Journal of Finance* 7: 77–91.
7. Maronna, Ricardo A., Douglas Martin, and Víctor J. Yohai. 2006. *Robust Statistics: Theory and Methods*. Hoboken: John Wiley & Sons.
8. Maronna, Ricardo A., Douglas Martin, Victor J. Yohai, and Matías Salibián-Barrera. 2019. *Robust Statistics Theory and Methods (with R)*. Hoboken: John Wiley & Sons.
9. Mba, Jules Clement, Kofi Agyarko Ababio, and Samuel Kwaku Agyei. 2022.
10. Markowitz mean-variance portfolio selection and optimization under a behavioral spectacle: New empirical evidence. *International Journal of Financial Studies* 10: 28.
11. Milošević, Nemanja, and Milos Racković. 2019. Classification based on missing features in deep convolutional neural networks. *Neural Network World* 29: 221–34.

12. Mitra Thakur, Gour Sundar, Rupak Bhattacharyya, and Seema Sarkar (Mondal). 2018. Stock portfolio selection using Dempster–Shafer Evidence theory. *Journal of King Saud University-Computer and Information Sciences* 30: 223–35.
13. Nguyen, Than Thi. 2014. Selection of the right risk measures for portfolio allocation. *International Journal of Monetary Economics and Finance* 7: 135.
14. Ortiz, Roberto, Mauricio Contreras, and Cristhian Mellado. 2021. Improving the volatility of the optimal weights of the Markowitz model. *Economic Research-Ekonomiska Istraživanja*, September 29.
15. Paiva, Felipe D., Rodrigo T. Cardoso, Gustavo P. Hanaoka, and Wendel M. Duarte. 2019. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications* 115: 635–55.
16. Rahiminezhad Galankashi, Masoud, Farimah Mokhtab Rafiei, and Maryam Ghezelbash. 2020. Portfolio selection: A fuzzy-ANP approach. *Financial Innovation* 6: 17.
17. Rather, Akhter M., Arun Agarwal, and V. N. Sastry. 2015. Recurrent neural network and a hybrid model for prediction of Stock returns. *Expert Systems with Applications* 42: 3234–41.
18. Sadouk, Lamyaa. 2019. CNN approaches for Time Series Classification. *Time Series Analysis-Data, Methods, and Applications*, November 5.
19. Sharpe, William F., and Harry M. Markowitz. 1989. Mean-variance analysis in portfolio choice and capital markets. *The Journal of Finance* 44: 531.
20. Siami-Namini, Sima, Neda Tavakoli, and Akbar S. Namin. 2019. The performance of LSTM and BiLSTM in forecasting time series. Paper presented at 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9–12.
21. Sikalo, Mirza, Almira Arnaut-Berilo, and Azra Zaimovic. 2022. Efficient Asset Allocation: Application of game theory-based model for superior performance. *International Journal of Financial Studies* 10: 20.
22. Singh, Upma, Mohammad Rizwan, Muhannad Alaraj, and I. Alsaidan. 2021. A machine learning-based gradient boosting regression approach for wind power production forecasting: A step towards Smart Grid Environments. *Energies* 14: 5196.
23. Soeryana, E., N. Fadhlina, Sukono, E. Rusyaman, and S. Supian. 2017. Mean-variance portfolio optimization by using time series approaches based on logarithmic utility function. *IOP Conference Series: Materials Science and Engineering* 166: 012003.
24. Ta, Van-Dai, Chuan-Ming Liu, and Direselign A. Tadesse. 2020. Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences* 10: 437.
25. Tu, Juntu, and Guofu Zhou. 2010. Incorporating economic objectives into bayesian priors: Portfolio choice under parameter uncertainty. *Journal of Financial and Quantitative Analysis* 45: 959–86.
26. Wan, Yuqing, Raymond Y. Lau, and Yain-Whar Si. 2020. Mining subsequent trend patterns from financial time series. *International Journal of Wavelets, Multiresolution and Information Processing* 18: 2050010.
27. Wang, Wuyu, Weizi Li, Ning Zhang, and Kecheng Liu. 2020. Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications* 143: 113042.
28. Yang, Mo, and Jing Wang. 2022. Adaptability of Financial Time Series prediction based on bilstm. *Procedia Computer Science* 199: 18–25.
29. Zaimovic, Azra, Adna Omanovic, and Almira Arnaut-Berilo. 2021. How many stocks are sufficient for equity portfolio diversification? A review of the literature. *Journal of Risk and Financial Management* 14: 551.
30. Khan, Ameer Tamoor, Xinwei Cao, Shuai Li, Bin Hu, and Vasilio N. Katsikis. 2021. Quantum beetle antennae search: A novel technique for the constrained portfolio optimization problem. *Science China Information Sciences* 64: 152204.
31. Khan, Ameer Tamoor, Xinwei Cao, Inova Brajevic, Predrag S. Stanimirovic, Vasilio N. Katsikis, and Shuai Li. 2022. Non-linear activated beetle antennae search: A novel technique for non-convex tax-aware portfolio optimization problem. *Expert Systems with Applications* 197: 116631.
32. Vineet Tirth, Ali Algahtani, Abdulaziz Alghtani, Tawfiq Al-Mughanam, and Kashif Irshad, 2023. Sustainable Nanomaterial-Based technologies for renewable energy production and efficient storage based on Machine learning. *Techniques*.
33. U. Sakthi, T. AnilKumar, Kuraluka Vimala Kumar, Shamimul Qamar, Girish Kumar Sharma, and Abdul Azeem. 2023. Power grid based renewable energy analysis by photovoltaic cell machine learning architecture in wind energy ybridization.
34. Martin János Mayer, Bence Biró, Botond Szücs, and Attila Aszódi. 2023. Probabilistic modeling of future electricity systems with high renewable energy penetration using machine learning.
35. Rasha M. Abd El-Aziz.2022, Renewable power source energy consumption by hybrid machine learning model.
36. Milad Izanloo, Alireza Aslani, and Rahim Zahedi. 2022. Development of a Machine learning assessment method for renewable energy investment decision making.