*Corresponding Author's Email: yqzhao7@connect.hku.hk
Proceedings of the International Conference on Applied Research in Business, Management and Economics, Vol. 2, Issue. 1, 2025, pp. 17-28
DOI:https://doi.org/10.33422/bmeconf.v2i1.1015

ISSN: 3030-0754 online

Integrating Machine Learning Methods for the Prediction in Online Portfolio Selection Problems

Zhonglin Liu, Yuqiao Zhao*, Benmeng Lyu and Wai-Ki Ching

Department of Mathematics, The University of Hong Kong, Hong Kong

Abstract

Online portfolio selection (OLPS) is a critical issue in computational finance. It sequentially updates portfolio allocations across multiple investment periods as new information becomes available. The main objective of OLPS is to maximize the final cumulative return, typically achieved through asset price prediction and portfolio optimization steps in each investment period. The properties of financial data, such as non-linearity, make certain machine learning methods applicable to the problem with potential benefits. To explore the effectiveness of integrating machine learning methods on OLPS, this work employs two machine learning models, the Long Short-Term Memory Networks (LSTM) and Extreme Gradient Boosting (XGBoost), on the asset price forecasting stage. These models are integrated with three optimization models: Mean-Variance, Max-Return, and On-Line Moving Average Reversion (OLMAR) to facilitate the decision-making process. For comparison purpose, a traditional price forecasting approach, the Exponential Moving Average (EMA) model, is utilized with the same optimization models as control groups. Numerical experiments are conducted using three commonly used public datasets, and the performance of the OLPS models is evaluated in terms of both final cumulative wealth and risk-adjusted return. The results indicate the advantages of incorporating machine learning models in various circumstances. Among the nine OLPS models, LSTM-based models outperform others in most scenarios. However, the effectiveness of XGBoost-based models varies depending on the optimization models and datasets used.

Keywords: Online Portfolio Selection, Machine Learning, LSTM, XGBoost, Exponential Moving Average (EMA)

1. Introduction

Online portfolio selection (OLPS) is a primary issue in computational finance, and research of it has extended to other areas such as statistics and artificial intelligence (Li and Hoi, 2014). OLPS determines the optimal portfolio allocations over multiple investment periods sequentially (Li & Hoi, 2015). Before making each investment decision using the optimization algorithm, accurately forecasting asset values to estimate future returns is an essential step. Capturing price patterns and numerically combining price trends are two widely developed approaches in the prediction stage of OLPS (Xi et al., 2023). However, due to the elaborate and non-linear property of financial data, traditional methods are not sufficient for financial analysis (Dai et al., 2024). This work aims to explore the potential benefits of integrating machine learning techniques into the prediction stage of OLPS strategies.

The Long Short-Term Memory Networks (LSTM) (Schmidhuber & Hochreiter, 1997) and Extreme Gradient Boosting (XGBoost) (Chen & Guestrin, 2016) from two categories of machine learning models are selected for asset price anticipating. LSTM is a specialized Recurrent Neural Network (RNN), which effectively processes sequential data, such as time series data (Graves, 2012). However, standard RNN has limitations on long-term dependencies due to the vanishing gradient problem (Bengio et al., 1994). The architecture of LSTM is designed to deal with the issue (Miao et al., 2015), making it suitable for forecasting financial time series (Martelo et al., 2022). XGBoost is built on gradient boosted regression tree, whose mechanism is integrating weak information to recognize complex patterns and relations that are difficult for linear algorithms to detect (Moghar & Hamiche, 2020). With the improvement in both speed and performance (Hongjoong, 2021), XGBoost is more applicable to practical problems. As noted by Chen (2023), XGBoost excels in predicting stock prices due to its sophisticated handling of complex data relationships and its high predictive accuracy. To better understand the effectiveness of these two machine learning models, we apply the traditional Exponential Moving Average (EMA) (Li & Hoi, 2015) used in OLPS literature as a comparison method. Three optimization models: Mean-Variance (Markowitz, 1952), Max-Return, and On-Line Moving Average Reversion (Li & Hoi, 2012) are integrated for the second stage of OLPS.

The remainder of the work is structured as follows. In Section 2, we present the problem formulation based on several assumptions. Then, we explore the structures of three prediction models for OLPS in Section 3. In Section 4, we present three optimization models and the merging of a prediction model with an optimization model. Section 5 displays the settings of numerical experiments and the corresponding results. Ultimately, we conclude the work in Section 6.

2. Problem formulation

This section expounds the decision-making framework of OLPS, representing a classical example of sequential optimization. The assumptions following some literature (Li et al., 2015) are made to simplify the analysis of developing and evaluating the machine learning models for OLPS. First, no transaction costs or taxes are incurred while trading. Second, assets can be bought or sold in any quantity at the closing price. Third, the implementation of portfolio selection strategies does not affect the market behavior and other assets' prices. We acknowledge that these aspects may have an impact on practical applications (Li et al., 2015).

Considering an investor plans to invest in n different assets over T periods, where the first K periods data is treated as historical data and the investment behavior starts from the $K+1$ period. The historical closing price over period t is denoted as $\boldsymbol{p_t} = (p_{t1}, p_{t2}, \dots, p_{tn})^{\top}$, where p_{ti} is the closing price of asset i at period t, for $i = 1, 2, \cdots, n$ and $t = 1, 2, \cdots, T$. The price

relative vector for all assets in period *t* is denoted as $\boldsymbol{r}_t = (r_{t1}, r_{t2}, \dots, r_{tn})^\top$, where $r_{ti} = \frac{p_{ti}}{p_{(t-1)i}}$ for $t = 2, \cdots, T$ and $r_{1i} = 1$. The asset return vector in period t is $\boldsymbol{y}_t = (y_{t1}, y_{t2}, \dots, y_{tn})^\top$, where $y_{ti} = r_{ti} - 1$. The predicted price vector, predicted price relative vector, and predicted return vector at period t is denoted by $\hat{\boldsymbol{p}}_t, \hat{\boldsymbol{r}}_t, \hat{\boldsymbol{y}}_t \in \mathbb{R}^{n\times 1}$, respectively. Based on the latest information, at the beginning of each period, the investor decides on an investment strategy $\boldsymbol{x}_t \in \mathbb{R}^{n\times 1}$, where x_{ti} denotes the fraction of capital allocated to asset *i* at period *t*. Given the initial wealth at the end of period K, W_K, the final cumulative wealth by the end of period *T* is given by Eq. (1).

$$W_T = W_K \prod_{t=K+1}^{T} \boldsymbol{r}_t^\top \boldsymbol{x}_t. \tag{1}$$

3. Prediction models

This section is devoted to explaining how to apply the three models EMA, LSTM, and XGBoost to the stock price prediction part of OLPS in three subsections, respectively. Note that since the price is predicted for each asset individually, we simplify the notation in this section by omitting i, e.g. we use p_t instead of p_{ti}. For a fair comparison, all three models use the same latest K historical price $\mathbf{p}_h = [p_{t-K+1}, p_{t-K+2}, \dots, p_t]$ to predict $\hat{p}_{t+1}$.

3.1 EMA model

As a variation of the Weighted Moving Average (WMA), the EMA model employs all historical data, with more recent data having a higher weight. It is commonly used in time series forecasting, particularly in financial markets for tracking stock prices and trading volumes (Singla & Malik, 2016). The predicted price at time t+1 can be calculated by the Model (2).

$$\begin{aligned} &EMA_1 = p_{t-K+1}, \\ &EMA_k = \alpha \cdot p_{t-K+k} + (1-\alpha) \cdot EMA_{k-1}, \\ &\quad \text{for } k = 2, 3, \cdots, K. \end{aligned} \tag{2}$$

Here, α is the smoothing factor ranging from 0 to 1, which is set as $\alpha = \frac{2}{K+1}$. Then $\hat{p}_{t+1} = EMA_K$.

3.2 LSTM model

The LSTM network consists of a series of LSTM blocks, whose input includes the output (information) of the previous block. The number of blocks depends on the time step size of training data. The framework of one LSTM block is demonstrated in Figure 1.

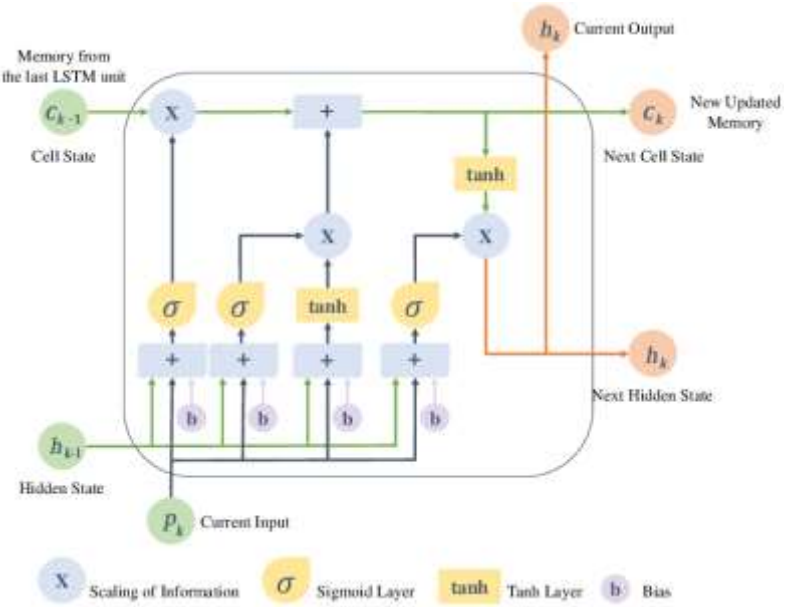

Figure 1: One LSTM block Framework (Wang et al., 2021)

Given the training dataset $\mathbf{p}_h$ and time step size ω, we first generate new training data $\mathbf{p}_{xLSTM} = \{\mathbf{p}_{t-K+\omega}, \mathbf{p}_{t-K+\omega+1}, \cdots, \mathbf{p}_{t-1}\}$ and $\mathbf{p}_{yLSTM} = \{\mathbf{p}_{t-K+\omega+1}, \mathbf{p}_{t-K+\omega+2}, \cdots, \mathbf{p}_t\}$, where $\mathbf{p}_k = [p_{k-\omega+1}, p_{k-\omega+2}, \cdots, p_k]$ for $k = t-K+\omega, t-K+\omega+1, \cdots, t$. Let batch size equal 1, the LSTM model is trained iteratively by $(\mathbf{p}_{xLSTM}, \mathbf{p}_{yLSTM})$. The LSTM block dealing with p_k comprises cell state c_k together with three gates: the forget gate f_k, the input gate i_k, and the output gate o_k, then the block output and hidden state represented by h_k can be obtained with o_k and c_k (Farzad et al., 2019). At time k, the gates and states are computed by the following equations:

$$\begin{cases} f_k &= \sigma(W_{f1}p_k + W_{f2}h_{k-1} + b_f), \\ i_k &= \sigma(W_{i1}p_k + W_{i2}h_{k-1} + b_i), \\ o_k &= \sigma(W_{o1}p_k + W_{o2}h_{k-1} + b_o), \\ \tilde{c}_k &= \tanh(W_{c1}p_k + W_{c2}h_{k-1} + b_c), \\ c_k &= f_k \odot c_{k-1} + i_k \odot \tilde{c}_k, \\ h_k &= o_k \odot \tanh(c_k), \end{cases}$$

where $\sigma(\cdot)$ and $\tanh(\cdot)$ represent the sigmoid and hyperbolic tangent functions, respectively, the operator $\odot$ is the element-wise product. Let $* = \{i, f, o, c\}$ and d be the hidden size, then $W_{*1} \in \mathbb{R}^{d\times 1}$ and $W_{*2} \in \mathbb{R}^{d\times d}$ are weight matrices, and $b_* \in \mathbb{R}^{d\times 1}$ are bias vectors. They can be trained by adopting the Mean Squared Error (MSE) loss function calculated by Eq. (3).

$$MSE = \frac{\sum_{j=k-\omega+1}^{k} (p_j - \hat{p}_j)^2}{\omega}. \tag{3}$$

3.3 XGBoost model

Besides closing price, using XGBoost for prediction needs more features. The widely used technical analysis indicators, Relative Strength Index (RSI) (Ţăran-Moroşan, 2011), EMA over

a period of 9 (EMA9), Moving Average Convergence and Divergence (MACD) (Appel, 2005), and MACDdiff (the difference between MACD and MACD signal (Appel, 2005)) are calculated as additional training features. Provided $\mathbf{p}_h$ and window size of RSI ω_{RSI}, we first generate $\boldsymbol{m}_k = (p_k, RSI_k, EMA9_k, MACD_k, MACDdiff_k)^\top$, for $k = t-l+1, t-l+2, \cdots, t$. Note that $l = K - \omega_{RSI} + 1$ since the first $\omega_{RSI} - 1$ terms of RSI are not applicable. Then we train XGBoost model with regenerated training dataset $(M, \mathbf{p}_{yXGBoost})$, where $M = \{\mathbf{m}_{t-l+1}, \mathbf{m}_{t-l+2}, \cdots, \mathbf{m}_{t-1}\}$, and $\mathbf{p}_{yXGBoost} = [p_{t-l+2}, p_{t-l+3}, \cdots, p_t]$. The model prediction of $\hat{p}_{k+1}$ can be given by Eq. (4):

$$\hat{p}_{k+1} = \sum_{d=1}^{D} g_d(\mathbf{m}_k), \tag{4}$$

where g is a tree in the space of regression trees and D is the number of trees.

During the training process, the objective function to be minimized is defined as Eq. (5):

$$\text{Obj}(\Theta) = \sum_{k=t-l+1}^{t-1} L(p_{k+1}, \hat{p}_{k+1}) + \sum_{d=1}^{D} \Omega(g_d), \tag{5}$$

where Θ denotes the model parameters, $L(\cdot)$ is the loss function assessing the model's prediction accuracy on training data, and $\Omega(\cdot)$ as expressed in Eq. (6) represents the regularization term that controls model complexity to prevent over-fitting.

$$\Omega(g) = \gamma N + \frac{1}{2}\lambda \sum_{j=1}^{N} u_j^2, \tag{6}$$

where N is the number of tree leaves, $\boldsymbol{u} \in \mathbb{R}^N$ is the vector of leaf weights, γ is the penalty on the number of leaves, and λ is the L_2 regularization coefficients.

Instead of learning parameters of all trees at once, it adds the newly learned tree to the already learned ones. The predicted value after generating *d-th* tree can be expressed as $\hat{p}_{k+1}^{(d)} = \hat{p}_{k+1}^{(d-1)} + g_d(\mathbf{m}_k)$, note that $\hat{p}_{k+1}^{(0)} = 0$. Choosing squared error as the loss function $L(\cdot)$, the objective function at step d can be expressed as follows:

$$\text{Obj}^{(d)} = \sum_{k=t-l+1}^{t-1} [p_{k+1} - (\hat{p}_{k+1}^{(d-1)} + g_d(\mathbf{m}_k))]^2 + \sum_{j=1}^{d} \Omega(g_j). \tag{7}$$

Applying the second-order Taylor expansion for the loss and removing all constants, the parameters at step d are updated by minimizing Eq. (8).

$$\sum_{k=t-l+1}^{t-1} [\Psi_k g_d(\mathbf{m}_k) + \frac{1}{2}\Phi_k g_d^2(\mathbf{m}_k)] + \Omega(g_d), \tag{8}$$

where $\Psi_k = \partial_{\hat{p}_{k+1}^{(d-1)}} L(p_{k+1}, \hat{p}_{k+1}^{(d-1)})$ and $\Phi_k = \partial^2_{\hat{p}_{k+1}^{(d-1)}} L(p_{k+1}, \hat{p}_{k+1}^{(d-1)})$.

4. Online portfolio selection frameworks

In this section, we first introduce the application of three optimization models Mean-Variance, Max-Return, and On-Line Moving Average Reversion for updating portfolios in a

specific period $t+1$ of OLPS. Then we demonstrate how to integrate the prediction models with the optimization models in OLPS.

4.1 Mean-Variance model

The Mean-Variance (MV) model generates an efficient frontier to consider the trade-off between maximizing returns and minimizing risks (Hongjoong, 2021). Here, to decide the portfolio allocation of time $t+1$, we apply Model (9) to maximize a quadratic function of expected return with a penalty on the stock variance (risk).

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \boldsymbol{x}^T \hat{\boldsymbol{y}}_{t+1} - \frac{\delta}{2} \boldsymbol{x}^T \nu_{t+1} \boldsymbol{x}, \\ \text{s.t.} \quad & \boldsymbol{x}^T \mathbf{1} = 1, \\ & \boldsymbol{x} \geq \mathbf{0}, \end{aligned} \tag{9}$$

where $\nu_{t+1} \in \mathbb{R}^{n \times n}$ is the asset return covariance matrix at time $t+1$, calculated based on the past 251 return data, δ is the risk-aversion parameter, and $\mathbf{1} \in \mathbb{R}^{n \times 1}$ is the column vector of all ones.

4.2 Max-Return model

The Max-Return model (MaxRet) with the risk-aversion parameter of Model (9) set to 0 is available for investors considering only the maximum expected return. It can be obtained by solving Model (10).

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \boldsymbol{x}^T \hat{\boldsymbol{y}}_{t+1}, \\ \text{s.t.} \quad & \boldsymbol{x}^T \mathbf{1} = 1, \\ & \boldsymbol{x} \geq \mathbf{0}. \end{aligned} \tag{10}$$

4.3 OLMAR

Empirical studies indicate that the mean reversion trading principle, arguing that the stocks' performance will reverse in the future, is suitable for the markets (Li et al., 2013). As a multiple-period mean reversion, On-Line Moving Average Reversion (OLMAR) approach propounded by Li and Hoi (2012), is designed for online portfolio selection with the sequential nature. The fundamental idea for its optimization part, as formulated in Model (11), is maximizing expected return while maintaining or making minimal adjustments to the original asset allocation.

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{x}_t\|^2, \\ \text{s.t.} \quad & \boldsymbol{x}^T \hat{\boldsymbol{r}}_{t+1} \geq \epsilon, \\ & \boldsymbol{x}^T \mathbf{1} = 1, \\ & \boldsymbol{x} \geq \mathbf{0}, \end{aligned} \tag{11}$$

where ϵ is a threshold. The algorithm of the portfolio updating process refers to (Li and Hoi, 2012).

4.4 Integration of prediction and optimization models

Once the latest information (price) $\boldsymbol{p}_t$ is obtained, $\widehat{\boldsymbol{p}}_{t+1}$can be predicted with the prediction models mentioned in Section 3. The predicted price relative vector $\hat{\boldsymbol{r}}_{t+1}$ and predicted return vector $\widehat{\boldsymbol{y}}_{t+1}$can be calculated, correspondingly. Then the portfolio allocation strategy $\boldsymbol{x}_{t+1}$ is determined by optimization models. The whole procedure for updating the portfolio of OLPS is shown in Algorithm 1.

Input: Historical prices of all n assets; Parameters of specific prediction and optimization models; Training data size, K.
Output: Final cumulative wealth, W_T.
Initialize: Investment strategy, $\boldsymbol{x}_K$; Original wealth, W_K.
For $t = K, K+1, \ldots, T-1$ **do**
  Provide the latest K historical price of each asset i, $\boldsymbol{p}_{h,i} = \left[p_{t-K+1,i}, p_{t-K+2,i}, \ldots, p_{t,i}\right]$.
  Train a prediction model and then predict the price of next period, $\hat{p}_{t+1,i}$ for each asset i (EMA model skips the training process).
  Generate the price vector of the next period, $\widehat{\boldsymbol{p}}_{t+1}$.
  Calculate predicted price relative vector, $\hat{\boldsymbol{r}}_{t+1} = \frac{\widehat{\boldsymbol{p}}_{t+1}}{\boldsymbol{p}_t}$.
  Calculate predicted return vector, $\widehat{\boldsymbol{y}}_{t+1} = \frac{\widehat{\boldsymbol{p}}_{t+1}}{\boldsymbol{p}_t} - 1$.
  Update portfolio allocation vector by the particular optimization model, $\boldsymbol{x}_{t+1}$.
  Calculate price relative vector, $\boldsymbol{r}_{t+1} = \frac{\boldsymbol{p}_{t+1}}{\boldsymbol{p}_t}$.
  Update cumulative wealth, $W_{t+1} = W_t \boldsymbol{r}_{t+1}^{\top} \boldsymbol{x}_{t+1}$
end

Algorithm 1: Integration of prediction and optimization models for OLPS.

5. Numerical experiments

This section details numerical experiments of 9 combination models on 3 datasets. Each prediction model (EMA, LSTM, and XGBoost) is combined in series with one of the optimization models (MV, MaxRet, and OLMAR) to build a combination model for addressing OLPS. Subsection 5.1 introduces datasets and settings of models' parameters. The following three subsections present the results of the final cumulative wealth, Sharpe ratio, and Calmar ratio, respectively.

5.1 Experimental setup

The numerical experiments are conducted on subsets of NYSE-O, NYSE-N, and TSE datasets in (Li and Hoi, 2015), consisting of consecutive 504 daily trading data. To better distinguish from the original datasets, we mark the subsets as NYSE-O', NYSE-N', and TSE', respectively. The NYSE-O' dataset comprises 36 American stocks starting from Jun. 3, 1962, the NYSE-N' dataset includes 23 American stocks beginning on Jan. 1, 1985, and the TSE' dataset contains 88 Canadian stocks since Jan. 4, 1994. The latest 252 (K) consecutive trading data serves as training data to anticipate the stock price in the next period.

The time step size ω of LSTM is set to 30. For XGBoost, the window size of RSI ω_{RSI} equals 14, and the number of trees D is 500 with the maximum depth 5 for each tree. The risk-aversion parameter of MV optimization model δ is fixed on 1. According to Li and Hoi (2012), the threshold of OLMAR ϵ is selected as 10, where the model achieves relatively stable

performance across various datasets. The initial value of wealth W_K is given 1 and portfolio allocation $\mathbf{x}_K$ equals $\left(\frac{1}{n}, \ldots, \frac{1}{n}\right)^\top$.

5.2 Final cumulative wealth

Final cumulative wealth acquired by Eq. (1) represents the total wealth accumulated from the start to the conclusion of all the investment periods. It is of significant interest to investors due to its ability to directly reflect the profit-generating performance of OLPS algorithms.

Table 1: Final cumulative wealth

Model	NYSE-O'	NYSE-N'	TSE'
EMA+MV	2.0880	1.1569	1.6779
LSTM+MV	**2.6489**	**1.2976**	**4.9267**
XGBoost+MV	1.3605	1.2856	1.5152
EMA+MaxRet	2.4152	1.0434	2.7642
LSTM+MaxRet	2.0577	**3.4811**	**2.8809**
XGBoost+MaxRet	**2.5799**	1.3751	1.3970
OLMAR (EMA)	1.4635	1.2782	1.3142
OLMAR (LSTM)	**3.2477**	**2.5776**	**2.2079**
OLMAR (XGBoost)	2.0827	1.2535	1.3743

Table 1 demonstrates the final cumulative wealth over 252 investment periods of all combination models. The bold figures highlight the best-performance integration models with the same optimization model over three datasets. The combination models with the LSTM model exhibit remarkable performance in almost all cases. It indicates that introducing LSTM as a prediction model can achieve stable and good performance compared to exploiting EMA and XGBoost models. The outstanding performance of the LSTM+MV model on the TSE' dataset, increasing the initial wealth to nearly 5 times, further verifies the effectiveness of the LSTM model for processing sequence data. However, the XGBoost+MaxRet model stands out on the NYSE-O' dataset, suggesting that while LSTM leads in most situations, XGBoost may better capture market nuance hidden in extra features. The comparison of the performance of EMA-based and XGBoost-based models varies across different optimization models and datasets. It indicates that XGBoost model could be an option for risk-appetite investors.

The observations highlight the significant benefits of incorporating the machine learning technique LSTM with portfolio optimization strategies in achieving higher cumulative wealth than the traditional EMA model. Conversely, integrating the other kind of machine learning model XGBoost may not have superior performance over the EMA method in some scenarios.

5.3 Sharpe ratio

When comparing portfolios that yield same expected returns, investors often prefer to choose the one exhibiting lower volatility. Besides final cumulative wealth, risk-adjusted return is also a significant indicator for investors to make decisions. It allows for a standardized comparison of portfolio performance under varying risk conditions. The Sharpe ratio is a commonly used risk-adjusted return metric. Initially introduced by Nobel laureate Sharpe in 1966 (Sharpe, 1994), this ratio takes into account the investment risk characterized by the volatility of returns. With daily returns, we firstly obtain the daily excess returns of portfolios over a daily risk-free rate by Eq. (12). Then the formula of Sharpe ratio as given in Eq. (13) is $\sqrt{252}$ multiplying the mean value of daily excess return divided by its standard deviation.

$$dailyExcessReturn_t = \boldsymbol{y}_t^\top \boldsymbol{x}_t - \left[\left(1 + r_f\right)^{\frac{1}{252}} - 1\right],$$
$$\text{for } t = K+1, K+2, \cdots, T, \tag{12}$$

where r_f represents the annualized risk-free rate.

$$\text{Sharpe ratio} = \frac{mean(dailyExcessReturn)}{Std(dailyExcessReturn)} \times \sqrt{252}. \tag{13}$$

Table 2: Sharpe ratio

Model	NYSE-O'	NYSE-N'	TSE'
EMA+MV	1.9704	0.5471	1.3990
LSTM+MV	**3.1519**	0.7747	**3.9084**
XGBoost+MV	1.7359	**1.1011**	2.2154
EMA+MaxRet	1.6955	0.2483	1.7902
LSTM+MaxRet	1.3806	**2.9757**	**1.9907**
XGBoost+MaxRet	**2.0271**	1.0861	0.9062
OLMAR (EMA)	3.3521	1.4881	2.5703
OLMAR (LSTM)	**3.9779**	**3.3265**	**4.2933**
OLMAR (XGBoost)	3.3714	1.0908	2.0686

The Sharpe ratio with r_f equals 0.02 of all combination models on the three datasets are displayed in Table 2. The comparative analysis demonstrates a pronounced ability of the LSTM-based models to achieve superior risk-adjusted returns. The performance of the OLMAR (LSTM) model surpasses its counterparts over all three datasets, showcasing LSTM's capability to simulate complex patterns and volatility is well reflected by the OLMAR optimization model. The excellent performance of the LSTM-based models on the TSE' dataset demonstrates LSTM's promising application in the Canadian stock market. The XGBoost+MaxRet model has outstanding performance on the NYSE-O' dataset while the XGBoost+MV model performs best on the NYSE-N' dataset, reflecting XGBoost may have a better adaptation to market characteristics. The XGBoost+MV model improves the Sharpe ratio significantly compared with the EMA+MV model on NYSE-N' and TSE' datasets and slightly decreases on NYSE-O'. Combined with corresponding cumulative wealth depicted in Table 1, we observe that the excess return of XGBoost+MV shows less volatility, which is preferred for risk-averse investors. Though XGBoost-based models beat EMA-based models in some circumstances, it still depends on market conditions and integrated optimization models.

5.4 Calmar ratio

The Calmar ratio, devised by Young (1991), serves as another essential risk-adjusted measurement of investment portfolios within a specific timeframe. Contrary to the Sharpe ratio, which assesses overall volatility of return, the Calmar ratio quantifies the return per unit of potential downside loss assessed by the worst peak-to-trough performance during the investment period. This attribute is particularly pertinent for scrutinizing high-stakes trading strategies where significant drawdowns pose a critical risk factor. Referring to Magdon-Ismail & Atiya (2004), the Calmar ratio is the annualized return over a certain period divided by the maximum drawdown (MDD) of the same period. Following Pospisil & Vecer (2010), we first define the running maximum at time t as Eq. (14), and then MDD can be calculated by Eq. (15). The Calmar ratio can be expressed by Eq. (16).

$$RM_t = \max_{u \in [K+1, \cdots, t]} \prod_{j=K+1}^{u} \boldsymbol{r}_j^\top \boldsymbol{x}_j. \tag{14}$$

$$\text{MDD} = \max_{t\in[K+1,K+2,\cdots,T]} \frac{RM_t - \prod_{j=K+1}^{t} \boldsymbol{r}_j^{\top} \boldsymbol{x}_j}{RM_t}. \tag{15}$$

$$\text{Calmar ratio} = \frac{\text{Annualized Return}}{\text{MDD}}, \tag{16}$$

where $\text{Annualized Return} = \left(\prod_{t=K+1}^{T} \mathbf{r}_t^{\cdot} \mathbf{x}_t\right)^{\frac{252}{T-K}} - 1.$

Table 3: Calmar ratio

Model	NYSE-O'	NYSE-N'	TSE'
EMA+MV	6.0397	0.7563	2.2066
LSTM+MV	**17.5087**	0.5038	**25.4270**
XGBoost+MV	3.0727	**2.2069**	3.8410
EMA+MaxRet	4.2457	0.1475	4.1302
LSTM+MaxRet	2.1442	**10.6356**	**4.7848**
XGBoost+MaxRet	**5.3837**	1.5415	1.0606
OLMAR (EMA)	7.8724	2.1280	4.5672
OLMAR (LSTM)	**20.2930**	**13.5305**	**8.3300**
OLMAR (XGBoost)	12.5295	1.9046	4.2480

Table 3 evaluates the Calmar ratio of the 9 models over 3 datasets. Similar to the results of the Sharpe ratio in Table 2, LSTM-based models fulfill the best performance in most situations, showcasing their stable and superior risk-adjusted returns, regardless of the risk of variance in returns or downside risk. The highest Calmar ratio reaches 25.4270 of the LSTM+MV model on TSE' dataset, far exceeding the outcome of the other two models combined with MV, which highlights the effectiveness of LSTM in capturing the characteristics of time series data and making reasonable predictions. When using OLMAR as the optimization model, LSTM consistently stands out on the three datasets in accordance with the final cumulative return, the Sharpe ratio, and the Calmar ratio. It indicates that the valuable speculations of LSTM are fully utilized by OLMAR. Although XGBoost-based models outperform EMA-based models in more than half of all the cases, their effectiveness varies with the optimization model and dataset. It is worth noting that, the same phenomenon XGBoost is striking whenever LSTM fails, is observed under all three evaluation criteria. This indicates that the information obtained by LSTM and XGBoost is complementary to a certain extent and models employing machine learning techniques LSTM and XGBoost constantly deliver superb cumulative return and risk-adjusted return. These insights highlight the significant advantages of machine learning models over conventional financial modeling techniques, suggesting their promising applications in determining OLPS strategies, especially within highly volatile and unpredictable market environments.

6. Conclusions

In this work, we explore the integration of machine learning models LSTM and XGBoost with optimization models MV, MaxRet, and OLMAR for OLPS, intending to improve the forecasting of asset price and thereby enhancing the performance of combined models from the conventional prediction method EMA based models. The numerical experiments indicate that sophisticated machine learning models provide substantial improvements concerning both cumulative return and risk-adjusted return measured by Sharpe ratio and Calmar ratio. However, the ameliorated phenomenons depend on the selection of the optimization model and the behavior of the dataset, notably for XGBoost. The results of LSTM-based models showcase the utility of LSTM in coping with the sequentially updated financial time series data, whose

patterns are challenging to acquire by the traditional EMA model. This enhanced modeling capability is crucial for developing a more efficient OLPS model. The overall performance of LSTM-based models has the potential to be improved by integrating with other optimization models. For XGBoost-based models, the performance may be strengthened by alternating or adding the features used for training and prediction, which are closing price, RSI, EMA9, MACD, and the difference between MACD and MACD signal (MACDdiff) in our models.

We recognize the limitation of our study that does not consider transaction costs and price impact, which are factors not ignorable in real-world trading scenarios. Our future study may assess the performance of integrated OLPS models with machine learning techniques under more realistic market conditions, such as including transaction costs. Additionally, while this work focuses on the effects of machine learning models on the forecasting stage, their impacts on the optimization process of OLPS are worthy to be investigated. With the development of machine learning models, financial analysis and portfolio management are expected to be continuously innovated, leading to more sophisticated and adaptive financial tools.

Acknowledgements

This research work was supported by Hong Kong Research Grants Council under Grant Number 17309522 and Hung Hing Ying Physical Science Research, the University of Hong Kong.

References

Appel, G. (2005). *Technical analysis: power tools for active investors.* FT Press.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Chen, A. (2023). *High accuracy stock trend prediction using XGBoost model*. https://doi.org/10.17605/OSF.IO/HMJ53

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). ACM.

Dai, H. L., Lai, F. T., Huang, C. Y., Lv, X. T., & Zaidi, F. S. (2024). Novel online portfolio selection algorithm using deep sequence features and reversal information. *Expert Systems with Applications*, *255*, 124565. https://doi.org/10.1016/j.eswa.2024.124565

Farzad, A., Mashayekhi, H., & Hassanpour, H. (2019). A comparative performance analysis of different activation functions in LSTM networks for classification. *Neural Computing and Applications*, *31*(7), 2507–2521. https://doi.org/10.1007/s00521-017-3210-6

Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*. Springer.

Hongjoong, K. I. M. (2021). Mean-variance portfolio optimization with stock return prediction using xgboost. *Economic Computation and Economic Cybernetics Studies and Research*, *55*(4), 5–20. https://doi.org/10.24818/18423264155.4.21.01

Li, B., & Hoi, S. C. (2012). On-line portfolio selection with moving average reversion. In *Proceedings of the 29th International Coference on International Conference on Machine Learning* (pp. 563–570). ACM.

Li, B., & Hoi, S. C. (2014). Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, *46*(3), 1–36. https://doi.org/10.1145/2512962

Li, B., & Hoi, S. C. H. (2015). *Online portfolio selection: principles and algorithms.* CRC Press.

Li, B., Hoi, S. C. H., Zhao, P., & Gopalkrishnan, V. (2013). Confidence weighted mean reversion strategy for online portfolio selection. *ACM Transactions on Knowledge Discovery from Data*, *7*(1), 1–38. https://doi.org/10.1145/2435209.2435213

Li, B., Hoi, S. C., Sahoo, D., & Liu, Z. Y. (2015). Moving average reversion strategy for on-line portfolio selection. *Artificial Intelligence*, *222*, 104–123. https://doi.org/10.1016/j.artint.2015.01.006

Magdon-Ismail, M., & Atiya, A. F. (2004). Maximum drawdown. *Risk Magazine*, *17*(10), 99–102.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91.

Martelo, S., León, D., & Hernandez, G. (2022). Multivariate financial time series forecasting with deep learning. In J. C. Figueroa-García, C. Franco, Y. Díaz-Gutierrez, & G. Hernández-Pérez (Eds.), *Applied Computer Sciences in Engineering* (pp. 160–169). Springer.

Miao, Y., Gowayyed, M., & Metze, F. (2015, December). EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)* (pp. 167–174). IEEE.

Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science*, *170*, 1168–1173. https://doi.org/10.1016/j.procs.2020.03.049

Pospisil, L., & Vecer, J. (2010). Portfolio sensitivity to changes in the maximum and the maximum drawdown. *Quantitative Finance*, *10*(6), 617–627. https://doi.org/10.1080/14697680903008751

Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. *Neural Comput*, *9*(8), 1735–1780.

Sharpe, W. F. (1994). The Sharpe ratio. *Journal of Portfolio Management, 21*(1), 49–58. https://doi.org/10.1515/9781400829408-022

Singla, R., & Malik, N. S. (2016). Role of EMA in technical analysis: A study of leading stock markets worldwide. *Finance India*, *30*(3), 919.

Ţăran-Moroşan, A. (2011). The relative strength index revisited. *African Journal of Business Management*, *5*(14), 5855–5862. http://www.academicjournals.org/ajbm

Wang, C., Liu, X., Pei, J., Huang, Y., Zhang, Y., & Yang, J. (2021). Multiview attention CNN-LSTM network for SAR automatic target recognition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 12504–12513. https://doi.org/10.1109/JSTARS.2021.3130582

Xi, W., Li, Z., Song, X., & Ning, H. (2023). Online portfolio selection with predictive instantaneous risk assessment. *Pattern Recognition*, *144*, 109872. https://doi.org/10.1016/j.patcog.2023.109872

Young, T. W. (1991). Calmar ratio: A smoother tool. *Futures*, *20*(1), 40.