Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 3751452, 15 pages
https://doi.org/10.1155/2022/3751452

Research Article

IntelliPortfolio: Intelligent Portfolio for Enhanced Index Tracking Using Clustering and LSTM

Kan Yi,[1] Jin Yang,[2] Shuangling Wang,[1] Zhengtong Zhang,[2] Jing Zhang,[2] Jinqiu Song,[2] and Xiao Ren[2]

[1] *Science and Technology on Information Systems Engineering Laboratory, Changsha, China*
[2] *School of Computer Science and Technology, Xidian University, Xi'an, China*

Correspondence should be addressed to Jin Yang; yangjin@stu.xidian.edu.cn

Received 2 November 2021; Revised 28 February 2022; Accepted 2 March 2022; Published 14 April 2022

Academic Editor: Gabriel Luque

Enhanced index tracking (EIT) is an active research area in portfolio management that focuses on adding reliable value relative to the index on the basis of mimicking the behavior of the benchmark index. To solve the EIT problem, many approaches have been proposed. However, it still remains a critical challenge to efficiently generate a portfolio with good quality. In this study, we propose a learning-based approach named IntelliPortfolio for the EIT problem. IntelliPortfolio uses PCA and clustering to select stock and estimates the investment weight for each constituent stock using a long short-term memory (LSTM) network. Two advantages of the proposed algorithm are as follows. (1) It considers both the fundamentals and the price information for stocks and can balance the trade-off between the performance and the diversity of the selected stocks. (2) It uses a LSTM model to estimate investment weights, which is more capable to handle long sequences of input and is more robust to predict the future trend of stock market. Experimental results on the five real-world datasets of the international stock market illustrate the significant performance superiority of the proposed approach in comparison with five state-of-the-art algorithms.

1. Introduction

As one of the most important strategies of passive investment, index tracking describes the process of investing in a portfolio that attempts to match the performance of some specified benchmark indexes. Today, indexed portfolios often function as the core holding in an investor's overall equity allocation. The idea of "enhanced index tracking (EIT)" has recently gained tremendous importance because more and more fund managers seek to outperform the index by appreciating the core of index investing. The reasons are pretty obvious. By design, index tracking can only produce a similar return to the index, which makes index funds always underperform the index by the amount of fund costs [1]. Enhanced return, therefore, could often bring fund managers competitive advantages after deducting expenses to track and reward new customers. In fact, enhanced index tracking is a dual-objective problem seeking the optimal decisions in the trade-off between maximizing expected performance and minimizing tracking risk.

Existing methods for enhanced index tracking include statistic-based, heuristic, and learning-based, as discussed in Section 2. Statistic-based approaches are the most mature method and have been studied for many years. However, such approach requires a significant amount of calculation and becomes instable when the covariance matrix of the dataset is ill-conditioned or nonpositive [2]. Heuristic methods have been shown to be effective in finding good portfolios [3], but are inefficient in solving high-dimensional EIT problems. More specifically, they are prone to fall into local minimums when searching for a large solution space and often result in suboptimal portfolios [4]. Learning-based approaches, although still in their infancy, are continuously popular. Methods belonging to this category are sensitive to the stability of the stock market, and their performance often fluctuates significantly.

To address the above issues, we propose an Intelligent Portfolio algorithm for enhanced index tracking problem, referred to as IntelliPortfolio, which aims to automatically select the constituent stocks for the portfolio from a benchmark index and determine the investment weight for each constituent stock. The key ideas of IntelliPortfolio are to select constituent stocks for the portfolio using principal component analysis (PCA) and k-means clustering algorithm and to estimate the investment weight for each constituent stock using a long short-term memory (LSTM) network. The motivation here is twofold. (1) The use of PCA and clustering algorithm to select stock is that it can consider both the fundamentals and the price information for stocks and can balance the trade-off between the performance and the diversity of the selected stocks. (2) The use of LSTM to estimate investment weights is that LSTM is more capable to handle long sequences of input when compared to other recurrent neural networks and is more robust to predict the future trends of stock [5]. This strategy is shown to produce better solutions in our practical experiments.

In summary, our work makes the following contributions to the field:

(i) We propose a novel, principal component analysis (PCA) and clustering-based stock selection algorithm to select constituent stocks for a portfolio from the benchmark index. Our algorithm considers both the fundamentals and the price information for each stock and can balance the trade-off between the performance and the diversity of the selected stocks. Our extensive experiments also show the effectiveness and the generality of proposed stock selection algorithm.

(ii) We propose IntelliPortfolio algorithm to solve *EIT* problem, which seamlessly integrates the stock selection algorithm with a long short-term memory (LSTM)-based investment weight estimation algorithm. Given a stock dataset and the number of stocks in a portfolio, IntelliPortfolio can both select the constituent stocks of the portfolio and determine the investment weight for each constituent stock.

(iii) We evaluate the performance of IntelliPortfolio through extensive experiments using five real-world datasets of international stock markets. We show that IntelliPortfolio outperforms five state-of-the-art enhanced index tracking algorithms by 8.78%–665.97% in terms of four well-known performance metrics.

2. Related Work

Enhanced index tracking is an active research area in portfolio management based on index tracking which aims to replicate the performance of the benchmark. We can classify the previous studies for index tracking on this problem into three categories: statistic-based, heuristic, and learning-based approaches, as discussed below.

2.1. Statistic-Based Approach. Traditionally, index tracking problem has been treated as linear or quadratic programming problems. Wu et al. [6] presented several constrained cluster-based linear mixed-integer optimization models for tracking broad market indices using the developed Lagrangian and semi-Lagrangian relaxation approaches for computing near optimal solutions. Chen and Kwon [2] developed a 0-1 integer program model considering initial portfolio selection and subsequent investment in assets. Fang and Wang [7] proposed a bi-objective programming model for the index tracking portfolio selection problem. Edirisinghe [8] developed a closed-form cost-free solution to the index tracking portfolio selection problem. Wu and Wu [9] designed a multifactor liner regression model as the basis of the tracking models and to enhance the capacity of the decision model, and a Lagrangian-based algorithm is applied to approximate optimal solutions. Oliver and Baumann [10] proposed a value-based mixed-integer linear programming (MILP) formulation for the index tracking problem that leads to a high similarity in terms of the normalized historical value developments between the tracking portfolio and the index and to low rebalancing costs.

Statistic-based approach is a category of classic solutions to the EIT problem and has been studied for many years. However, such approach requires high-precision dataset and a significant amount of calculation. Moreover, they suffer from the poor stability when the covariance matrix of the dataset is ill-conditioned or nonpositive [2].

2.2. Heuristic Approach. Because the enhanced index tracking problem has been proven to be NP hard [11–15], many previous studies apply heuristic algorithms to solve it. Mutunge and Haugland [16] proposed a greedy constructive heuristic algorithm that can extend the current portfolio by a single asset in each iteration. Sant'Anna et al. [17] applied a hybrid solution approach combining mathematical programming and genetic algorithm to deal with the volatility problem of stock markets in developing countries. Chen et al. [18] introduced an indexed portfolio optimization model using the mean-variance-skewness framework and proposed a hybrid algorithm combining the firefly algorithm (FA) and the genetic algorithm (GA). Strub and Oliver [19] developed an iterated greedy heuristic for replicating the 1/N portfolio by investing in a subset from a given investment universe. Salehpoor and Molla-Alizadeh-Zavardehi [20] presented a hybrid metaheuristic algorithm to solve the index tracking problem. Beasley et al. [21] presented an evolutionary heuristic algorithm for the index tracking problem. Orito et al. [22] proposed a two-step stock selection algorithm that uses a heuristic method to select stocks and then uses a genetic algorithm to construct a portfolio. Oh et al. [23] used GA to optimize index funds, aiming at tracking stock index and minimizing tracking error. Roland and Berg [24] proposed a hybrid algorithm combining GA with quadratic programming to search for the optimal tracking portfolio. Kumar and Mishra [11] proposed a multiobjective optimizer for portfolio optimization which is based on covariance-guided Artificial Bee Colony (ABC) algorithm. Chen et al. [3] proposed a grouping genetic algorithm for solving the group trading strategy portfolio (GTSP) problem. Saborido et al. [25] proposed the mean-downside risk skewness (MDRS) model and defined the new mutation, crossover, and reparation operators for an evolutionary multiobjective optimization to solve the index tracking problem. Benidis et al. [26] provided a unified framework for a large variety of sparse index tracking formulations and derived a mixed-integer programming (MIP)-based algorithm considering various tracking error functions and constraints. Canakgoz and Beasley [27] presented a mixed-integer linear programming formulations for index tracking problem. García et al. [28] proposed a genetic algorithm and Tabu search-based for index tracking optimization.

Heuristic methods have been proven to be effective in finding good portfolios [3], but they often suffer from the inefficiency in the EIT problem with a high-dimensional space [29]. They are prone to fall into the local minimum when searching for the large solution space [4] and often result in suboptimal portfolios.

2.3. Learning-Based Approach. Fu et al. [29] presented a stacking stock selection model based on supervised learning, used a genetic algorithm to select stock features, and labelled stocks according to the return-to-volatility ratio ordering. Dose and Cincotti [30] presented a stochastic-optimization technique based on time-series cluster analysis for index tracking problem. Ouyang et al. [4] used the deep autoencoder to select the portfolio and then used the deep neural network to dynamically determine the portfolio weight. Zhang and Tan [31] proposed a new stock selection model named DeepStockRanker to predict the future stock return ranking based on the historical data without handcrafted features. Chalvatzis and Hristu-Varsakelis [32] proposed a deep long short-term memory (LSTM) model to predict asset prices and a prediction-based trading strategy. Paiva et al. [33] proposed a method combining support vector machine (SVM) and mean-variance (MV) for portfolio selection. Jiang et al. [34] proposed a deep reinforcement learning framework aiming to maximize cumulative returns using deterministic policy gradient (DPG). Liang et al. [35] implemented three new continuous reinforcement learning algorithms, namely, deep deterministic policy gradient (DDPG), proximal policy optimization (PPO), and policy gradient (PG), in portfolio management. Moody et al. [36] trained two kinds of reinforcement learning methods, namely, real-time recurrent learning (RTRL) and Q-learning, to solve index tracking problem. Park et al. [37] proposed an approach for deriving a multiasset portfolio trading strategy using deep Q-learning. Lu [38] implemented a learning model using LSTM with reinforcement learning or evolution strategies as agents. Zhang and Maringer [39] developed a model that combines GA with recurrent reinforcement learning (RRL) for asset trading. García-Galicia et al. [40] provided a reinforcement learning model in continuous-time discrete-state portfolio management with time penalization for transaction cost. Vo et al. [41] introduced a deep responsible investment portfolio model containing a multivariate bidirectional LSTM neural network to predict stock returns.

Learning-based approach has received much attention recently. However, many learning-based algorithms construct the portfolio by predicting the future price of stocks, which has been proven to be inaccurate [41]. Reinforcement learning-based algorithms, on the contrary, can construct the portfolio adaptively. However, our experiment shows that they are very sensitive to the stability of the stock market, and their performance fluctuates significantly.

3. Problem Statement

This study focuses on the enhanced index tracking problem, referred to as EIT, that aims to produce a portfolio that attempts to earn a higher return than the benchmark index (excess return) while minimizing the risk of deviating from the benchmark (tracking risk). In this section, we introduce notations and terminology first and then define the EIT problem formally.

3.1. Notations and Terminology

(i) t: time point.

(ii) T: decision time point. $[1, T]$ is the in-sample time period to select tracking portfolio, and $[T, T+L]$ is the out-of-sample time period to evaluate it.

(iii) N: number of stocks in the benchmark index.

(iv) K: number of stocks in the portfolio.

(v) p_i^t: price of stock i $(i = 1, 2, \ldots, N)$ at time point t $(t = 1, 2, \ldots, T+L)$.

(vi) I^t: index value at time t $(t = 1, 2, \ldots, T+L)$.

(vii) x_i^t: stock selection indicator for a portfolio. If the ith stock is selected to form the portfolio at time point t, $x_i^t = 1$; otherwise $x_i^t = 0$.

(viii) w_i^t: investment weight for the ith stock in the tracking portfolio $(i = 1, 2, \ldots, N)$.

(ix) r_i^t: the single period continuous time return of the tracking portfolio at time point t and $t-1$ $(t = 2, 3, \ldots, T+L)$:

$$r^t = \ln \frac{\sum_{i=1}^N x_i^t \cdot p_i^t \cdot w_i^t}{\sum_{i=1}^N x_i^{t-1} \cdot p_i^{t-1} \cdot w_i^{t-1}}. \tag{1}$$

(x) R^t: the single period continuous time return of the benchmark index at time point t and $t-1$ $(t = 2, 3, \ldots, T+L)$:

$$R^t = \ln \frac{I^t}{I^{t-1}}. \tag{2}$$

(xi) TE: tracking error of the portfolio, which is defined as the distance between the returns of the tracking portfolio and its benchmark index [1]:

$$TE = \frac{1}{T}\left[\sum_{t=2}^{T}\left(r^t - R^t\right)^2\right]^{1/2}$$

$$= \frac{1}{T}\left[\sum_{t=2}^{T}\left(\ln\frac{\sum_{i=1}^{N} x_i^t \cdot p_i^t \cdot w_i^t}{\sum_{i=1}^{N} x_i^{t-1} \cdot p_i^{t-1} \cdot w_i^{t-1}} - \ln\frac{I^t}{I^{t-1}}\right)^2\right]^{1/2}. \tag{3}$$

(xii) *ER*: excess return of the portfolio, which is assessed by the average excess return per period achieved by the tracking portfolio [1]:

$$ER = \sum_{t=1}^{T}\frac{1}{T}\left(\ln\sum_{i=1}^{N}\frac{p_i^{t+1}}{p_i^t}\cdot w_i^t \cdot x_i^t - \ln\frac{I^{t+1}}{I^t}\right). \tag{4}$$

3.2. Problem Formulation. As shown in Figure 1, the EIT problem consists of two phases named in-sample and out-of-sample, respectively, and the decision time point is T. Any solutions to the EIT problem need to construct a tracking portfolio first according to the performance of stock market during the in-sample phase (from the time point 1 to T in Figure 1). The ultimate goal is to find an optimal tracking portfolio that can obtain the minimized tracking error and the maximized excess return during the out-of-sample phase (from the time point T to $T + L$ in Figure 1). However, it is impossible to optimize this problem directly because a solution has to finish the construction of a portfolio at the decision time point T, and the price of a stock during the out-of-sample period (i.e., $[T, T + L]$) is *unknown* at T. To deal with this issue, we follow the previous studies [1] and simplify the EIT problem by optimizing the portfolio *during the in-sample phase instead of the out-of-sample phase*, which assumes that the stock prices between $[1, T]$ and $[T, T + L]$ are independent and identical distributed (*i.i.d*).

In summary, our EIT can be defined as a two-objective optimization problem:

$$\min(TE) = \frac{1}{T}\left[\sum_{t=2}^{T}\left(r^t - R^t\right)^2\right]^{1/2}$$

$$= \frac{1}{T}\left[\sum_{t=2}^{T}\left(\ln\frac{\sum_{i=1}^{N} x_i^t \cdot p_i^t \cdot w_i^t}{\sum_{i=1}^{N} x_i^{t-1} \cdot p_i^{t-1} \cdot w_i^{t-1}} - \ln\frac{I^t}{I^{t-1}}\right)^2\right]^{1/2}, \tag{5}$$

$$\max(ER) = \sum_{t=1}^{T}\frac{1}{T}\left(\ln\sum_{i=1}^{N}\frac{p_i^{t+1}}{p_i^t}\cdot w_i^t \cdot x_i^t - \ln\frac{I^{t+1}}{I^t}\right), \tag{6}$$

$$\text{s.t.} \sum_{i=1}^{N} x_i^t = K, \quad \forall t \in [1, T], \tag{7}$$

$$w_i^t \geq 0; \quad \forall t \in [1, T], i \in [1, N], \tag{8}$$

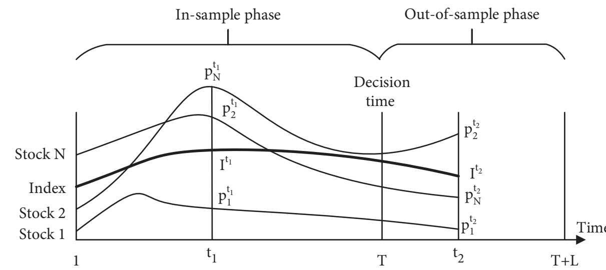


Figure 1: Overview of EIT problem.

$$\sum_{i=1}^{N} w_i^t = 1, \quad \forall t \in [1, T], \tag{9}$$

where T and K is given, and equations (5) and (7) state that the goal of *EIT* is to optimize the tracking error and excess return simultaneously.

Equations (9)–(11) state the constraints for the EIT problem. Equation (9) ensures that the number of stocks selected in the tracking portfolio is equal to K at any time point. Equation (9) ensures that no short positions are considered in a tracking portfolio. Equation (10) normalizes the investment weight for each stock in the tracking portfolio.

3.3. Complexity Analysis. This formulation shows that the goal of EIT problem is to search for an optimal portfolio consisting of K stocks, each of which is selected from a known benchmark index containing N $(N \gg K)$ stocks, to minimize tracking error and maximize excess return. According to previous studies [11–15], the enhanced index tracking problem is essentially a classic combinatorial optimization (CO) problem and has been proven to be NP hard. The NP-completeness proof is established in [15]. Therefore, a naive exhaustive search solutions would not be practical due to the high dimensionality of decision space and the combinatorial nature of brutal force search.

4. Algorithm Design and Implementation

In this section, we introduce IntelliPortfolio, an efficient clustering and LSTM-based portfolio construction algorithm to solve the EIT problem. Its key idea is twofold: firstly, it applies principal component analysis (PCA) and k-means clustering algorithm to automatically select K constituent stocks for the portfolio from the benchmark index; secondly, it uses a long short-term memory (LSTM) network to determine the investment weight for each constituent stock in the portfolio. In the following, we first present the overview of our IntelliPortfolio algorithm. We then present the constituent stock selection algorithm and the weight estimation algorithm. Finally, we discuss the details of the IntelliPortfolio algorithm.

4.1. Overview. Figure 2 illustrates the detailed steps of IntelliPortfolio. IntelliPortfolio contains two phases: constituent stock selection and investment weight estimation. At the constituent stock selection phase, after normalizing the original dataset, we use a PCA-based algorithm to reduce the dimensionality of the original dataset and apply a k-means clustering algorithm to generate K clusters from the benchmark index and then add K stocks that are closest to the center in each cluster to the portfolio. At the investment weight estimation phase, we adopt a novel windowed-random sampling strategy to generate random samples with fixed-window size and apply the LSTM model to estimate the investment weight for each constituent stock, which finally finishes the construction of the portfolio.

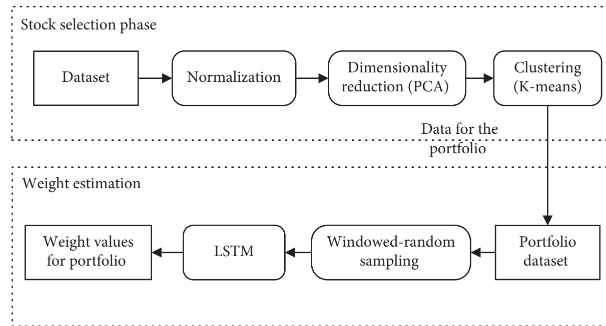


Figure 2: Overview of IntelliPortfolio approach.

4.2. Constituent Stock Selection. The first step of constructing a tracking portfolio is to select K constituent stocks from the benchmark index consisting of N stocks. Previous studies on this problem include industry-based method [34], trading volume-based method [23], and autoencoder-based method [4]. Industry-based method takes into account of the information on industry, market capitalization, and trading amount for each stock. Such approach is interpretable and easy to understand, but fail to reflect the trend in stock market [1]. Trading volume-based method, by contrast, can indicate the fluctuation of stock price effectively, but ignores the fundamentals of a stock. Autoencoder-based method trains a deep encoder-decoder network by using the stock price dataset and can adaptively select K stocks with the largest and smallest training error. However, it suffer from the low performance issue when the benchmark index contains a large number of stocks.

To deal with these issues, we propose a novel stock selection algorithm integrating PCA and clustering method. The key of the our algorithm is twofold. (1) It considers both the fundamentals and the price information for each stock by receiving all known features from a dataset and using a PCA algorithm to find out the major factors. (2) It uses a clustering algorithm to automatically divide the stocks in the benchmark index into K clusters and selects stocks that are closest to the center in each cluster as the constituent stocks of the portfolio. Our intensive experiment shows that this selection strategy has the ability to balance the trade-off between the performance and the diversity of the selected stocks.

Specifically, the detailed process of our stock selection algorithm is listed in Algorithm 1.

As shown in Algorithm 1, the stock selection algorithm starts by performing mean-variance normalization on every feature in the original dataset D (line 3). After that, it applies a PCA algorithm [42] to reduce the dimensionality of D to R, where R is a small, positive integer given by a human expert (line 4). The idea here is to retain the important features containing as much as the variance of the dataset, while removing the insignificant features whose variance is near to zero.

Given the time point τ and the time interval P that are used to select the stocks, we repeat P times of applying a k-means clustering algorithm with cluster number K to generate P different sets of clusters, each of which represent the clustering result during time period $[\tau, \tau + P]$ (line 6). For each time point $t \in [\tau, \tau + P]$, we find the stocks that are closest to the center of each cluster as the candidate constituent stocks of the portfolio (line 7), where the function $\text{Distance}(s, s_c^t)$ measures the distance between a stock s and the center s_c^t of a cluster. Once we get the candidate stocks in each time point, we put them together (line 8) and compute the total number of occurrences during $[\tau, \tau + P]$ (line 11) and then find the stocks with the K-maximum number of occurrences (i.e., S^*) as the constituent stocks of the portfolio. Finally, the algorithm extracts the information for stocks in S^* from the dataset D' (line 14) and returns the two-tuple (S^*, D^*). Note that the characteristic of this selection algorithm is that it can consider both the fundamentals and the price information for each stock over a period of time.

4.3. Investment Weight Estimation. After determining K constituent stocks for the portfolio, we need to assign the investment weight for each constituent stock. In this study, we propose a long short-term memory (LSTM) network to estimate the investment weight for each constituent stock, as shown in Figure 3.

As shown in Figure 3, our LSTM networks are composed of an input layer, a hidden layers, and an output layer. The number of neurons in the input layer is equal to the number of explanatory variables (feature space) reduced in stock selection. The number of neurons in the output layer reflects the output space, i.e., K neurons in our case indicating the weight for each constituent stock in the portfolio in $t + 1$. The main characteristic of LSTM networks is contained in the hidden layer consisting of so-called memory cells. Each of the memory cells has three gates maintaining and adjusting its cell state s_t: a forget gate (f_t), an input gate (i_t), and an output gate (o_t).

When processing an input sequence, its features are presented time point by time point to the LSTM network. Hereby, the input at each time point t (in our case, the information of constituent stocks) is processed by the network as denoted in the equations above. Once the last element of the sequence has been processed, the final output for the whole sequence is returned. The detailed process is described as follows.

Require: D: original dataset; K: the number of stocks in the portfolio; R: the desired number of features after dimensionality reduction; τ: the start time point for selection; P: the decision time interval for stock selection.
Ensure: S^*: constituent stocks in the portfolio; D^*: the dataset consisting of the information of K constituent stocks in the portfolio.

$S \longleftarrow \varnothing$;
$S^* \longleftarrow \varnothing$;
Perform mean-variance normalization on every feature of all stocks in D;
$D' \longleftarrow \mathrm{PCA}(D, R)$;
for $t = \tau$: $\tau + P$ **do**
    $S^t \longleftarrow \mathrm{K-means}[D'[t], K]$;
    $S^t \longleftarrow \underset{\forall s \in S^t}{\mathrm{argmin}}\, \mathrm{Distance}(s, s_c^t)$;
    $S \longleftarrow S \cup S^t$;
end for
for $i = 1$: K **do**
    $S^* \longleftarrow S^* \cup \underset{\forall s \in S}{\mathrm{argmax}}(\mathrm{Count}(S, \{s\}))$;
    $S \longleftarrow S - \{s\}$;
end for
$D^* \longleftarrow \mathrm{Extract}(D', S^*)$; **return** (S^*, D^*);

Algorithm 1: Stock selection algorithm: StockSelection (D, K, R, τ, P).

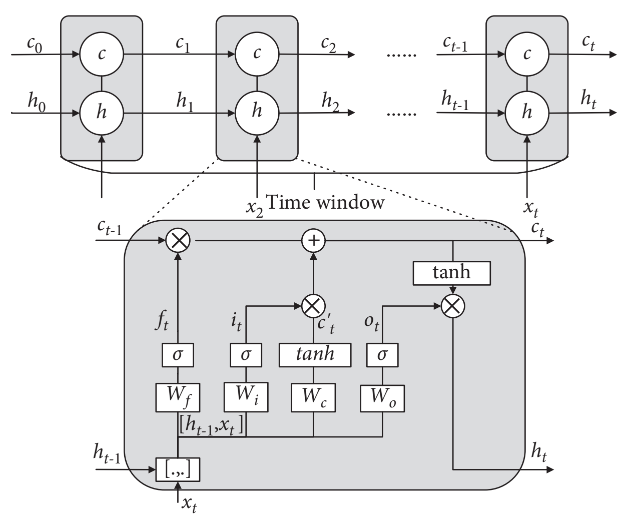


Figure 3: The architecture of our LSTM model.

Require: D: the dataset consisting of information of K constituent stocks; Q: the length of a data sequence for training; iter: training iterations.
Ensure: M^*: the LSTM model for weight estimation.

$M^* \longleftarrow \mathrm{RandomInit}$;
$T\ D \longleftarrow \mathrm{DataSplit}(D, Q)$;
for $i = 1$: iter **do**
    $Seq \longleftarrow \mathrm{RandomPick}(T\ D)$;
    $M \longleftarrow \mathrm{LSTM}(Seq, M^*)$;
    $M^* \longleftarrow \underset{\forall}{\mathrm{argmin}}\, M\mathrm{Loss}(M, f)$;
end for return M^*;

Algorithm 2: Weight estimation algorithm: WeightEstimation (D, Q, iter).

The weight estimation algorithm first initializes the LSTM model with random parameters (line 1) and splits the dataset D into training dataset $T\ D$ containing $|D| - Q + 1$ continuous data sequences (line 2). During each iteration, it randomly picks a data sequence Seq from $T\ D$ (line 3), generates a newer LSTM model M (line 4), and finally optimizes the model according to the loss function (lines 5 and 6). More specifically, the loss function f is defined as the weighted sum of TE and ER: $f = \alpha \cdot TE + (1 - \alpha) \cdot ER$, and the ADAM optimizer [43] is used in the training process.

4.4. IntelliPortfolio Algorithm. We are now ready to describe our IntelliPortfolio algorithm in Algorithm 3. IntelliPortfolio first cleans the original dataset by removing stocks with missing data (line 1) and selects K stocks to form the initial portfolio (line 2). After that, it obtains a LSTM-based weight estimation model using stock information on portfolio (line 3). To finish the final optimization, IntelliPortfolio constructs the test dataset $D_{\Delta T}$ by extracting the data with ΔT length of time interval from $T - \Delta T$ to T in D (line 4) and produces the final investment weights for constituent stocks according to M^* and $D_{\Delta T}$ (line 5).

Note that, according to the definition of the EIT problem, D, T, and K are given. Our IntelliPortfolio algorithm contains seven hyperparameters, namely, R, τ, P, Q, α, $iter$, and ΔT, respectively, where

(i) R: the desired number of features after dimensionality reduction on D
(ii) τ: the start time point for stock selection
(iii) P: the time period for stock selection
(iv) Q: the length of a data sequence used for training a LSTM model
(v) α: the weight value for two-objectives TE and ER
(vi) iter: the training iterations of the LSTM model
(vii) ΔT: the time interval for final optimization

We will report the values of these hyperparameters in Section 5 and explain the basic principle for determining the values for some important hyperparameters.

5. Experiments

We have implemented our approach and conducted extensive experiments on five real-world datasets of international stock market. The source code and the data can be found in https://github.com/anon4review/IntelliPortfolio. In this section, we first describe our experiment setup and then present the experimental results to prove the efficiency and effectiveness of the proposed approach.

5.1. Experimental Settings

5.1.1. Datasets and Running Environment. We choose five well-known stock indices in the international stock markets as our datasets to evaluate our IntelliPortfolio algorithm, namely, SSE constituent index (SSE180) [44], Dow Jones Industrial Average (DJIA) [45], Financial Time Stock Exchange 100 Index (FTSE100) [46], Hang Seng Index (HSI) [47], and Nikkei stock average (Nikkei225) [48]:

(i) SSE180 chooses 180 sample stocks from all 1000 A-share stocks registered Shanghai, China. It reflects the profile and operation of the Shanghai stock market.
(ii) DJIA consists of stocks of 30 representative large industrial and commercial companies, which can roughly reflect the price level of the entire industrial and commercial stocks in the United States.
(iii) FTSE100 contains 100 representative stocks of influential companies in European. It is one of the most important indicators for global investors to observe the trend of European stocks [21].
(iv) HSI is an important indicator of Hong Kong stock market prices and represents 63% of the 12-month average market capitalization rate of all listed companies on the Hong Kong Stock Exchange; it contains 50 representative stocks.
(v) Nikkei225 contains 225 stocks with good continuity and comparability. It is the most common and reliable indicator for examining and analyzing the long-term evolution and dynamics of the Japanese stock market.

The available features of each dataset are shown in Table 1.

For each index, we extract ten-year (2431 trading days) data from 2009 to 2018 and remove stocks with missing data. As a result, we have 111 stocks in SSE180, 28 in DJIA, 58 in FTSE100, 39 in HSI, and 201 in Nikkei225. For each dataset, we choose 2371 trading days as the in-sample period (i.e., training dataset) and use the last 60 trading days as out-of-sample period (i.e., test dataset).

Table 2 lists the number of constituent stocks (i.e., K) in the result portfolio.

All experiments run on a computer equipped with four processors, 8GiB RAM, 512 GB disk, and running windows 10. To ensure consistency, we run each algorithm five times and calculate the average of these five runs.

5.1.2. Performance Metrics. We use four metrics to evaluate the performance of the algorithms, namely, tracking error (TE) [4, 16, 23], excess return (ER) [21], information ratio (IR) [2, 49], and the Sharp ratio (SR) [34, 36, 38], defined as follows:

(i) Tracking error ($TE\downarrow$): the performance difference between the portfolio and the benchmark index, which measures the tracking accuracy of the portfolio. We have defined TE in equation (3) and want to minimize the value of TE.
(ii) Excess return ($ER\uparrow$): the average excess return per period achieved by the tracking portfolio compared

Require: D: the original dataset; K: the number of stocks in the portfolio; T the decision time point.
Ensure: W: the set of weight values of constituent stocks in the portfolio.
(1) $D_c \longleftarrow \text{DataClean}(D)$;
(2) $D_p \longleftarrow \text{StockSelection}(D_c, K, R, \tau, P)$;
(3) $M^* \longleftarrow \text{WeightEstimation}(D_p, Q, iter)$;
(4) $D_{\Delta T} \longleftarrow \text{Extract}(D_c, [T - \Delta T, T])$;
(5) $W \longleftarrow \text{LSTM}(M^*, D_{\Delta T})$; **return** W;

ALGORITHM 3: The proposed IntelliPortfolio algorithm: IntelliPortfolio (D, K, T).

TABLE 1: Features of each dataset.

SSE180	Other four indices
Opening price	
Closing price	
Highest price	
Lowest price	
Average price	
Volume	
Turnover	Opening price
Ups and downs	Closing price
Quote change	Highest price
Hand turnover rate	Lowest price
A-share market capitalization	Volume
The total market capitalization	Adjusted closing price
A-share tradable share capital	
Total share capital	
Price-to-earnings ratio (PE)	
Price-to-book ratio (PB)	
Price-to-sales ratio (PS)	
Price cash flow ratio (PCF)	

TABLE 2: Number of constituent stocks in the portfolio.

Datasets	No. of constituent stocks	K
SSE180	111	10
DJIA	28	5
FTSE100	58	5
HSI	39	5
Nikkei225	201	20

to its benchmark index, which is defined in equation (4). We want to maximize the value of ER.

(iii) Information ratio $(IR\uparrow)$: a measurement of portfolio returns beyond the returns of a benchmark index, compared to the volatility of those returns. IR during the time interval $[1, T]$ can be defined as

$$IR = \frac{ER}{TE}, \tag{10}$$

where ER is the excess return and TE is the tracking error. We want to maximize the value of SR.

(iv) Sharp ratio $(SR\uparrow)$: a well-known measurement that indicates the performance of an investment (i.e. the portfolio in our case) compared to a risk-free asset, after adjusting for its risk. SR during the time interval $[1, T]$ is defined as

$$SR = \frac{E(r^t)}{s(r^t)}, \quad t \in [1, T], \tag{11}$$

where E is expectation and s is the standard variance. We want to maximize the value of SR.

The performance improvement of an algorithm over a baseline algorithm in comparison is defined as

$$Imp(\text{baseline}) = \frac{P - P_{\text{baseline}}}{P_{\text{baseline}}} \cdot 100\%, \tag{12}$$

where P_{baseline} is the performance of the baseline algorithm and P is that of the algorithm being evaluated.

5.1.3. Baseline Algorithms and Hyperparameters. Because IntelliPortfolio is a two-step optimization approach consisting of stock selection and weight estimation algorithms, we need to evaluate the performance of our stock selection algorithm first. Specifically, we compare it with four state-of-the-art methods, namely, random method (Random), industry-based method (Industry) [34], trading volume-based method (Volume) [23], and autoencoder-based method (Autoencoder) [4]. We provide a brief description for each method as follows and report its hyperparameters (including IntelliPortfolio) in Table 3.

Random method (Random) randomly selects K constituent stocks from the benchmark index

Industry-based method (Industry) selects K constituent stocks from the benchmark index considering the industry, the market capitalization, and the trading amount information for each stock in the market

Trading volume-based method (Volume) chooses the top K stocks from the benchmark index by sorting the total trading volume from large to small during a fixed period

Autoencoder-based method (Autoencoder) trains a encoder-decoder network using the historical dataset and selects K stocks with the largest and smallest training error

Finally, to evaluate the overall performance of our IntelliPortfolio method, we compare it with five state-of-the-art algorithms, namely, genetic algorithm and recurrent reinforcement learning (GA-RRL) [39], deep deterministic policy gradient (DDPG) [35], recurrent reinforcement learning (RRL) [38], DPG [34], and heuristic genetic algorithm (HGA) [21]. We provide a brief description for each algorithm as follows and report its hyperparameters (including IntelliPortfolio) in Table 4.

GA-RRL uses a genetic algorithm (GA) to improve the trading results of a RRL-type equity trading system. It takes the advantage of GA's capability to select an optimal combination of technical indicators, fundamental indicators, and volatility indicators for improving out-of-sample trading performance.

DDPG adopts a deep deterministic policy gradient algorithm to implement portfolio management, in which the agent takes the stock data during a fixed time interval and the current stock weights as the observed environment, and derives the weights of portfolio for the next day.

RRL uses trading volume-based method to select constituent stocks of a portfolio and adopts the recurrent reinforcement learning and LSTM model to implement portfolio management.

DPG is a financial-model-free reinforcement learning framework to provide a deep machine learning solution to the portfolio management problem. It consists of the Ensemble of Identical Independent Evaluators (EIIE) topology, a Portfolio-Vector Memory (PVM), an Online Stochastic Batch Learning (OSBL) scheme, and a fully exploiting and explicit reward function.

HGA selects stocks and determines portfolio weights based on the genetic algorithm (GA) and uses a heuristic method to update the population.

5.2. Experiment Results

5.2.1. Hyperparameter Estimation Results for Stock Selection Algorithm. Two important hyperparameters, namely, τ and P, are involved in our stock selection algorithm. τ represents the start time point for stock selection process, and P denotes the length of the time interval for stock selection. Specially, we divide each dataset into three subsets: a validation set containing 60 days of data, a test set consisting of 60 days of data, and a training set containing all of the remaining data. In our experiment, we choose time intervals of 60, 90, and 120 days for stock selection (i.e., $P = 60/90/120$), as suggested in [1, 21, 38, 39]. For each time interval, we try the first, the middle, and the last time point as our start time point for stock selection process.

Based on the above experiment design, we have performed $45 = 5 * 3 * 3$ groups of experiments, and only report the results on SSE180 dataset in Table 5 due to length limitations. Detailed results can be found in our online repository.

We can see from Table 5 that the results during the last stages containing 120 days (i.e., $\tau = 2191$ and $P = 120$, denoted as 2191-120) outperform other eight combinations by obtaining three out of four best values of our performance metrics, with the only exception of the TE. Considering the TE values obtained by these combinations are not significantly different and all others are, we can safely take TE as an unimportant metrics. Moreover, in terms of SR metric, which reflects both the risks and the benefits of a portfolio, the results of 2191-120 combination outperform others significantly.

The above results are reasonable because of the following. (1) The latest time point can reflect the recent trend of stock market. This is consistent with the findings of many previous studies in [1, 23, 35]. (2) The longest period of time interval can fully reflect the trends of stock, which are in turn utilized by our LSTM model. In summary, we make a conclusion that we should use the latest 120 days of data for stock selection, which is adopted in the following experiments.

5.2.2. Comparison Results with Different Stock Selection Algorithms. To prove the effectiveness and the robustness of our selection algorithm, we mix five stock selection algorithms (i.e., Random, Industry, Volume, Autoencoder, and IntelliPortfolio), five *EIT* algorithms requiring stock selection process (i.e., GA-RRL, DDPG, RRL, DPG, and IntelliPortfolio), and five dataset (i.e., SSE180, DJIA, FTSE100, HSI, and Nikkei225) together and conduct 225 groups of tests. Due to limited space, Table 6 shows the 25 groups of testing results on the SSE180 dataset and highlights the best values. For the complete experimental results, please check our online repository.

Table 3: Hyperparameters for stock selection methods.

Algorithm	Hyperparameters
Random	N/A
Industry	N/A
Volume	N/A
Autoencoder	HiddenUnitsNum: 640; CodeUnitsNum: 300; LearningRate: 0.001; TrainNum: 2000; TrainEpoch: 1
IntelliPortfolio	R: 3; τ: $T-120$; P: 120

Table 4: Hyperparameters for *EIT* algorithms.

Algorithms	Hyperparameters
GA-RRL	SequenceStep:30; WindowLen:20; LearningRate: 0.0001; stddev: 0.05; ElitistsNum: 10; ChoosenEliNum: 4; TrainEpoch: 15; TradeEpoch: 10; PopulationSize: 50; GenerationNum: 20; CrossoverProb: 0.5; MutationProb: 0.2
DDPG	λ: 0.3; MaxEpisodes: 1000; MaxEpSteps: 200; LrA: 0.001; LrC: 0.002; γ: 0.9; τ: 0.1; MemoryCapacity: 1000; BatchSize: 32
RRL	LearningRate: 0.0001
DPG	TrainingSteps: 8000; LearningRate: 0.0001; BatchSize: 40; FeatureNum: 3
HGA	Epsilon: 0.01; delta: 0.5; WinLen: 300; gamma: 0.1; Lambda: 0.3, α: 2.0; PopuationSize: 100; LoopNum: 10000; CrossoverProb: 0.5; MutationProb: 0.5

Table 5: Hyperparameter estimation results for stock selection algorithm.

τ	P	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
1	60	0.00198	−0.00026	−0.13240	0.95168
1	90	0.00221	0.00086	0.38720	0.87584
1	120	0.00293	0.00142	0.48443	1.37334
1156	60	0.00222	0.00006	0.02733	0.86875
1156	90	0.00144	−0.00045	−0.30875	0.73049
1156	120	0.00428	0.00018	0.04302	1.05809
2251	60	0.00541	0.00101	0.18715	1.56098
2221	90	0.00495	0.00066	0.13293	1.45561
2191	120	0.00465	0.00115	0.24705	1.57182

Table 6: Results of different selection algorithms on the SSE180 dataset.

IntelliPortfolio				
Selection algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.00554	0.00116	1.76507	0.24298
Random	0.00200	0.00002	0.44814	0.01141
Industry	0.00215	−0.00120	0.87021	−0.55862
Volume	0.00171	−0.00030	0.62009	−0.17336
Autoencoder	0.00475	0.00055	1.26564	0.09968
GA-RRL				
Selection algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.01103	0.00094	0.58642	0.11496
Random	0.00510	0.00050	0.24446	0.09727
Industry	0.00546	−0.00144	0.35697	−0.26416
Volume	0.00441	0.00011	0.35448	0.02391
Autoencoder	0.00818	0.00084	0.52252	0.07605
DDPG				
Selection algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.02431	0.00047	0.16269	0.01950
Random	0.00365	0.00012	0.07826	0.03227
Industry	0.00581	−0.00017	0.05993	−0.02978
Volume	0.00296	−0.00006	−0.23251	−0.02252
Autoencoder	0.01785	0.00021	−0.17050	0.01179

Table 6: Continued.

Selection algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
		RRL		
Selection algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.04002	0.00117	0.02298	0.01953
Random	0.01092	−0.00011	−0.01936	−0.01017
Industry	0.01738	−0.00008	−0.03844	−0.00483
Volume	0.01029	0.00019	−0.04926	0.01857
Autoencoder	0.04593	0.00090	0.02032	0.00753
		DPG		
Selection algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.00266	0.00118	−0.03730	0.36666
Random	0.00179	−0.00024	−0.14548	−0.13338
Industry	0.00085	−0.00018	−0.36048	−0.21024
Volume	0.00141	0.00008	−0.17127	0.05529
Autoencoder	0.00225	0.00060	−0.05151	0.26669

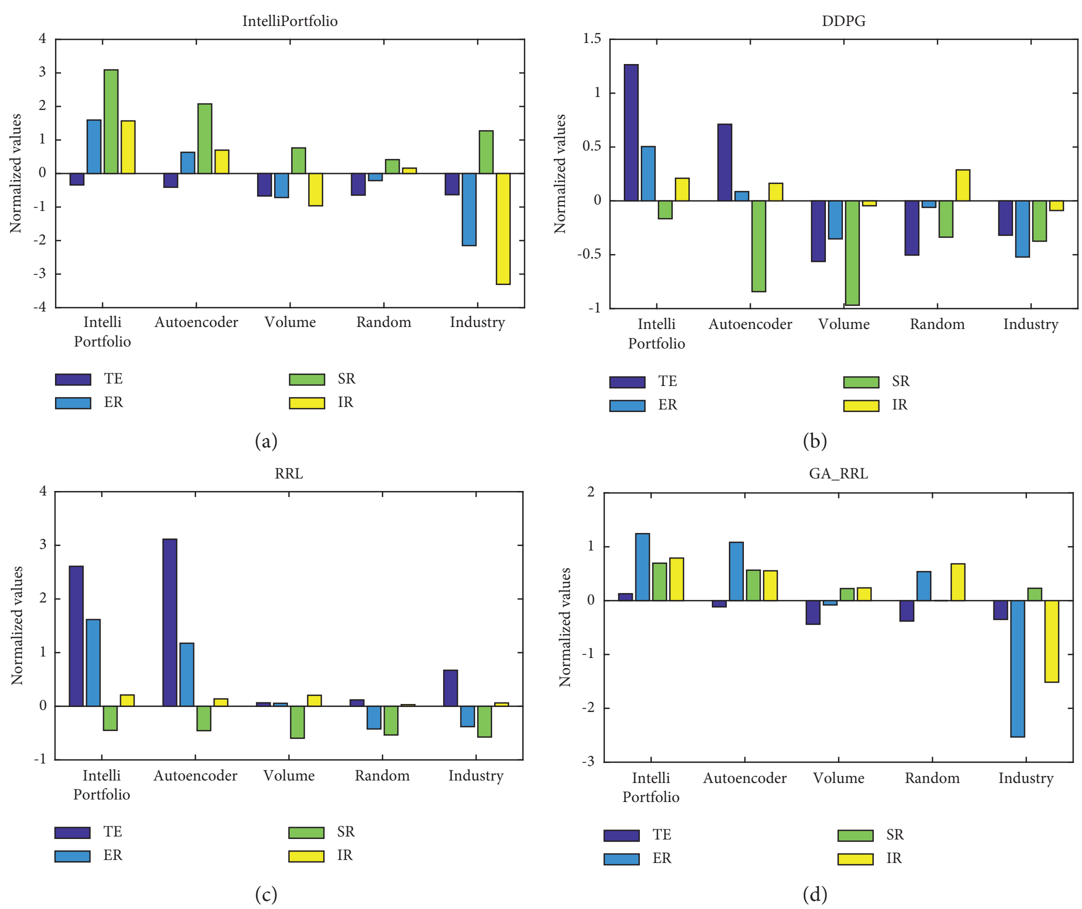
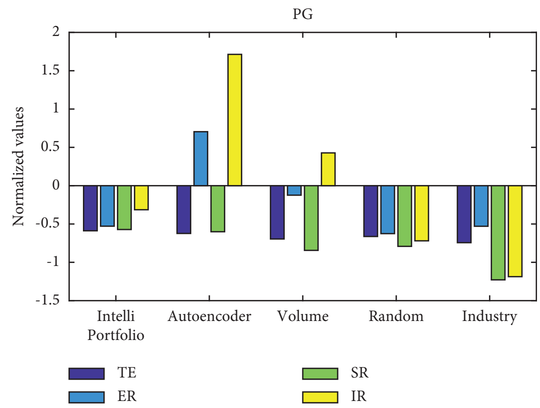


Figure 4: Continued.



(e)

FIGURE 4: Results for stock selection algorithms.

TABLE 7: Final results on five datasets.

		SSE180		
Algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.00554	0.00055	1.76506	0.19968
DPG	0.00559	$5.68095e-05$	−0.21228	0.09699
RRL	0.01029	0.00019	−0.04926	0.01857
DDPG	0.02431	0.00047	0.16268	0.01950
GA-RRL	0.00441	0.00010	0.35448	0.02391
HGA	0.04434	−0.01076	0.07013	−0.24273
		Nikkei225		
Algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.00114	0.00116	1.12680	0.05055
DPG	0.00096	−0.00142	−0.20583	−1.48279
RRL	0.01669	−0.00150	−0.02886	−0.08960
DDPG	0.01055	0.00033	0.13347	0.03162
GA-RRL	0.00483	−0.00098	0.43392	−0.20310
HGA	0.01521	−0.01194	−0.10225	−0.78510
		HSI		
Algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.00287	0.00072	0.66467	0.25105
DPG	0.00109	−0.00155	−0.10316	−1.42760
RRL	0.00694	−0.00147	−0.02697	−0.21127
DDPG	0.00280	0.00006	0.08823	0.02068
GA-RRL	0.00185	−0.00184	0.08630	−0.99876
HGA	0.01133	−0.01046	0.03701	−0.92311
		DJIA		
Algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.00375	0.00063	0.41209	0.16682
DPG	0.00167	−0.00014	−0.18196	−0.08172
RRL	0.01434	0.00008	−0.06911	−0.00553
DDPG	0.01403	−0.00048	0.13292	−0.03438
GA-RRL	0.00289	−0.00223	0.22810	−0.77050
HGA	0.01046	−0.00951	−0.20269	−0.90926
		FTSE100		
Algorithms	$TE\downarrow$	$ER\uparrow$	$SR\uparrow$	$IR\uparrow$
IntelliPortfolio	0.00238	−0.00002	0.43635	−0.00683
DPG	0.00212	−0.00080	−0.07507	−0.37783
RRL	0.02067	−0.00028	0.00667	−0.01370
DDPG	0.00156	−0.00049	−0.05788	−0.31460
GA-RRL	0.00516	−0.00060	0.40223	−0.11589
HGA	0.01884	−0.00855	0.07314	−0.45402

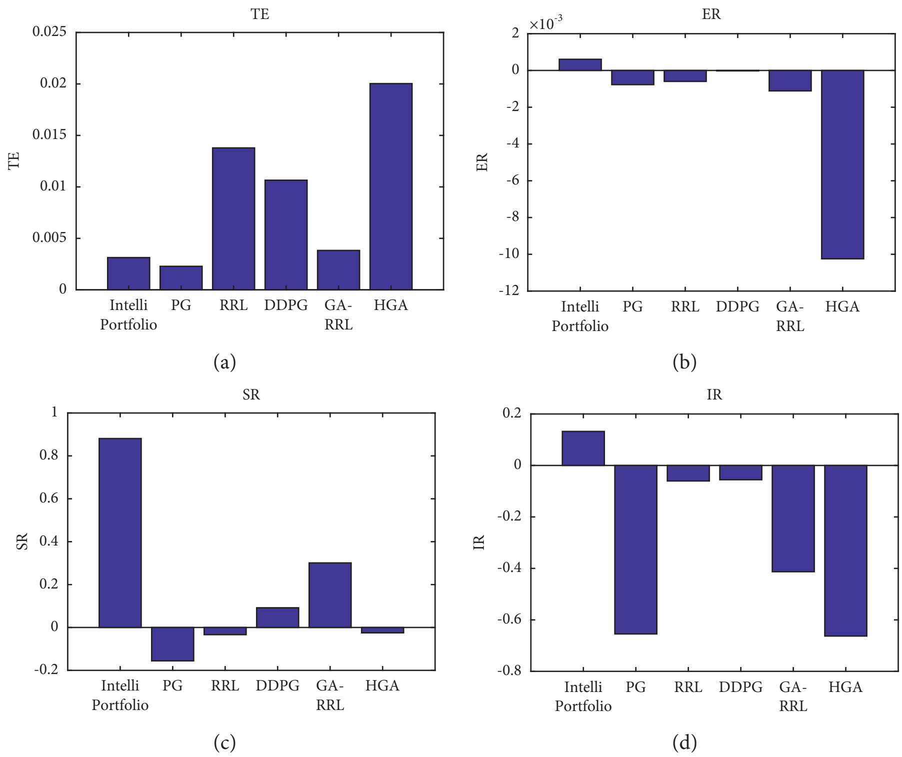


Figure 5: Average measurements of different algorithms on five databases.

We can see from Table 6 that our stock selection algorithm outperforms other four algorithms on five *EIT* algorithms over four performance metrics, i.e., obtaining 14 out of 20 best values in all. Note that the volume algorithm outperforms other algorithms on TE metric; this is because it only considers the trading volume while choosing stocks, and stocks with large trading volume are influencing factors for index pricing. Although volume algorithm performs well on TE, it is underperforming in other three metrics. Another important observation from Table 6 is that our stock selection algorithm achieves more significant improvements over the other algorithms on the *SR* metric, which means our algorithm has better performance when considering the TE and ER metrics simultaneously. In summary, the above results indicate that our selection algorithm not only has better performance than others but also has strong generality and is suitable for many *EIT* algorithms.

For a better illustration, we plot the four normalized performance metrics of five stock selection algorithms on SSE180 dataset in Figures 4(a)–4(e), respectively. In each figure, the x-axis lists the five algorithms and the y-axis represents the measurements of the four performance metrics. We conclude from Figure 4 that our stock selection algorithm achieves stable improvements compared with the other five algorithms, and the improvements are more significant in *SR* and IR metrics.

5.2.3. Overall Algorithm Performance Results. Finally, we compare our IntelliPortfolio algorithm with five state-of-the-art *EIT* algorithms on five international stock markets, and the results are listed in Table 7.

We can see from Table 7 that IntelliPortfolio outperforms other five algorithms on five datasets over four performance metrics, i.e., obtaining 16 out of 20 best values in all. Specifically, in terms of ER, our algorithm achieves an average performance improvement of 178.90% over DPG, 29.3% over RRL, 260.90% over DDPG, 98.01% over GA-RRL, and 89.16% over HGA; in terms of *SR*, our algorithm achieves an average performance improvement of 665.97% over DPG, 364.57% over RRL, 136.47% over DDPG, 69.47% over GA-RRL, and 130.73% over HGA; in terms of IR, our algorithm achieves an average performance improvement of 120.20% over DPG, 98.54% over RRL, 8.78% over DDPG, 86.57% over GA-RRL, and 37.71% over HGA. Note that although DPG and DDPG algorithms outperform other algorithms on TE metric, they are still underperforming in other three metrics. Such results illustrates the significant and consistent performance superiority of the IntelliPortfolio algorithm in comparison with five state-of-the-art algorithms.

Figure 5 illustrates the average performance measurements of different algorithms. We observe from Figure 5 that IntelliPortfolio outperforms all other five algorithms, followed by GA-RRL, DDPG, PG, RRL, and HGA. The difference between IntelliPortfolio and other algorithms is significant, and all others are not. Such results indicate that IntelliPortfolio is stable and robust on different datasets in comparison with other algorithms.

6. Conclusion and Future Work

In this study, we propose an learning-based approach named IntelliPortfolio for the *EIT* problem. It applies principal component analysis (PCA) and k-means clustering algorithm to automatically select constituent stocks for the portfolio from the benchmark index and uses long short term memory (LSTM) network to determine the investment weight for each constituent stock in the portfolio. We conducted extensive experiments on five real-world datasets of international stock market. The superiority of IntelliPortfolio was illustrated with four performance metrics in comparison with five state-of-the-art *EIT* algorithms.

It is of our future interest to make IntelliPortfolio more reactive by learning previous patterns of stock market and supporting automatic adjustments according to future market prediction. We will also explore the possibility of proposing practical reinforcement learning-based algorithms to make more intelligent and adaptive portfolio decisions.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declares no potential conflicts of interest.

References

[1] L. Qian, L. Sun, and B. Liang, "Enhanced index tracking based on multi-objective immune algorithm," *Expert Systems with Applications*, vol. 38, no. 5, pp. 6101–6106, 2011.

[2] C. Chen and Kwon, "Robust portfolio selection for index tracking," *Computers & Operations Research*, vol. 39, no. 4, pp. 829–837, 2012.

[3] C.-H. Chen and Y. H. Chen, M.-E. Wu, An effective approach for obtaining a group trading strategy portfolio using grouping genetic algorithm," *IEEE Access*, vol. 7, pp. 7313–7325, 2019.

[4] H. Ouyang, X. Zhang, and H. Yan, "Index tracking based on deep neural network," *Cognitive Systems Research*, vol. 57, pp. 107–114, 2019.

[5] D. M. Q. Nelson, A. C. M. Pereira, and R. A. d. Oliveira, "Stock market's price movement prediction with lstm neural networks," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1419–1426, IEEE, Anchorage, AK, USA, May 2017.

[6] D. Wu, Kwon, and G. Costa, "A constrained cluster-based approach for tracking the S&P 500 index," *International Journal of Production Economics*, vol. 193, pp. 222–243, 2017.

[7] Y. Fang and S.-Y. Wang, "A fuzzy index tracking portfolio selection model, Lecture Notes in Computer Science," in *Proceedings of the International Conference on Computational Science*, pp. 554–561, Springer, Georgia, USA, May 2005.

[8] N. C. P. Edirisinghe, "Index-tracking optimal portfolio selection," *Quantitative Finance Letters*, vol. 1, no. 1, pp. 16–20, 2013.

[9] D. Wu and D. D. Wu, "An enhanced decision support approach for learning and tracking derivative index," *Omega*, vol. 88, pp. 63–76, 2019.

[10] S. Oliver and P. Baumann, "Optimal construction and rebalancing of index-tracking portfolios," *European Journal of Operational Research*, vol. 264, no. 1, pp. 370–387, 2018.

[11] D. Kumar and K. K. Mishra, "Portfolio optimization using novel co-variance guided artificial bee colony algorithm," *Swarm and Evolutionary Computation*, vol. 33, pp. 119–130, 2017.

[12] T. F. Coleman, Y. Li, and J. Henniger, "Minimizing tracking error while restricting the number of assets," *Journal of Risk*, vol. 8, no. 4, p. 33, 2006.

[13] R. Moral-Escudero, R. Ruiz-Torrubiano, and S. Alberto, "Selection of optimal investment portfolios with cardinality constraints," in *Proceedings of the 2006 IEEE International Conference on Evolutionary Computation*, pp. 2382–2388, IEEE, Vancouver, BC, Canada, July, 2006.

[14] R. Mansini and M. G. Speranza, "Heuristic algorithms for the portfolio selection problem with minimum transaction lots," *European Journal of Operational Research*, vol. 114, no. 2, pp. 219–233, 1999.

[15] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is np-hard," *Journal of Global Optimization*, vol. 1, no. 1, pp. 15–22, 1991.

[16] P. Mutunge and D. Haugland, "Minimizing the tracking error of cardinality constrained portfolios," *Computers & Operations Research*, vol. 90, pp. 33–41, 2018.

[17] L. R. SantAnna, T. P. Filomena, P. C. Guedes, and D. Borenstein, "Index tracking with controlled number of assets using a hybrid heuristic combining genetic algorithm and non-linear programming," *Annals of Operations Research*, vol. 258, no. 2, pp. 849–867, 2017.

[18] W. Chen, Y. Wang, P. Gupta, and M. K. Mehlawat, "A novel hybrid heuristic algorithm for a new uncertain mean-variance-skewness portfolio selection model with real constraints," *Applied Intelligence*, vol. 48, no. 9, pp. 2996–3018, 2018.

[19] S. Oliver, N. Trautmann, B. Vitoriano, G. Parlier, and D. deWerra, "An iterated greedy heuristic for the 1/n portfolio tracking problem," in *Proceedings of the 5th International Conference on Operations Research and Enterprise Systems*, pp. 424–431, Lisbon, Portugal, February 2016.

[20] I. B. Salehpoor and S. Molla-Alizadeh-Zavardehi, "A constrained portfolio selection model at considering risk-adjusted measure by using hybrid meta-heuristic algorithms," *Applied Soft Computing*, vol. 75, pp. 233–253, 2019.

[21] J. E. Beasley, N. Meade, and T.-J. Chang, "An evolutionary heuristic for the index tracking problem," *European Journal of Operational Research*, vol. 148, no. 3, pp. 621–643, 2003.

[22] Y. Orito, H. Yamamoto, and G. Yamazaki, "Index fund selections with genetic algorithms and heuristic classifications," *Computers & Industrial Engineering*, vol. 45, no. 1, pp. 97–109, 2003.

[23] K. J. Oh, T. Y. Kim, and S. Min, "Using genetic algorithm to support portfolio optimization for index fund management," *Expert Systems with Applications*, vol. 28, no. 2, pp. 371–379, 2005.

[24] J. Roland and J. V. D. Berg, "Optimized index tracking using a hybrid genetic algorithm," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 2327–2334, IEEE, Hong Kong, China, June 2008.

[25] R. Saborido, A. B. Ruiz, Bermúdez, E. Vercher, and M. Luque, "Evolutionary multi-objective optimization algorithms for fuzzy portfolio selection," *Applied Soft Computing*, vol. 39, pp. 48–63, 2016.

[26] K. Benidis, Y. Feng, and D. P. Palomar, "Optimization methods for financial index tracking: from theory to practice,"

Foundations and Trends in Optimization, vol. 3, no. 3, pp. 171–279, 2018.
[27] N. A. Canakgoz and J. E. Beasley, "Mixed-integer programming approaches for index tracking and enhanced indexation," *European Journal of Operational Research*, vol. 196, no. 1, pp. 384–399, 2009.
[28] F. García, F. Guijarro, and J. Oliver, "Index tracking optimization with cardinality constraint: a performance comparison of genetic algorithms and tabu search heuristics," *Neural Computing & Applications*, vol. 30, no. 8, pp. 2625–2641, October 2018, http://link.springer.com/10.1007/s00521-017-2882-2.
[29] X. Fu, J. Du, Y. Guo, M. Liu, T. Dong, and X. Duan, "A Machine Learning Framework for Stock Selection," 2018, https://arxiv.org/abs/1806.01743.
[30] C. Dose and S. Cincotti, "Clustering of financial time series with application to index and enhanced index tracking portfolio," *Physica A: Statistical Mechanics and Its Applications*, vol. 355, no. 1, pp. 145–151, 2005.
[31] X. Zhang and Y. Tan, "Deep stock ranker: a lstm neural network model for stock selection," in *Proceedings of the International Conference on Data Mining and Big Data*, pp. 614–623, Springer, Shanghai, China, July 2018.
[32] C. Chalvatzis and D. Hristu-Varsakelis, "High-performance Stock index Trading: Making Effective Use of a Deep Lstm Neural Network," 2019, https://arxiv.org/abs/1902.03125.
[33] F. D. Paiva, R. T. N. Cardoso, G. P. Hanaoka, and W. M. Duarte, "Decision-making for financial trading: a fusion approach of machine learning and portfolio selection," *Expert Systems with Applications*, vol. 115, pp. 635–655, 2019.
[34] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," 2017, https://arxiv.org/abs/1706.10059.
[35] Z. Liang, H. Chen, J. Zhu, K. Jiang, and Y. Li, "Adversarial Deep Reinforcement Learning in Portfolio Management," 2018, https://arxiv.org/abs/1808.09940.
[36] J. Moody, M. Saffell, Y. Liao, and L. Wu, *Reinforcement learning for trading systems and portfolios: immediate vs future rewards, Decision Technologies for Computational Finance*, Springer, Boston, MA, pp. 129–140, 1998.
[37] H. Park, M. K. Sim, and D. G. Choi, "An Intelligent Financial Portfolio Trading Strategy Using Deep Q-Learning," 2019, https://arxiv.org/abs/1907.03665.
[38] D. W. Lu, "Agent Inspired Trading Using Recurrent Reinforcement Learning and Lstm Neural Networks," 2017, https://arxiv.org/abs/1707.07338.
[39] J. Zhang and D. Maringer, "Using a genetic algorithm to improve recurrent reinforcement learning for equity trading," *Computational Economics*, vol. 47, no. 4, pp. 551–567, 2016.
[40] M. García-Galicia, A. A. Carsteanu, and Clempner, "Continuous-time reinforcement learning approach for portfolio management with time penalization," *Expert Systems with Applications*, vol. 129, pp. 27–36, 2019.
[41] N. N. Y. Vo, X. He, S. Liu, and G. Xu, "Deep learning for decision making and the optimization of socially responsible investments and portfolio," *Decision Support Systems*, vol. 124, Article ID 113097, 2019.
[42] S. Mika, B. Schölkopf, A. J. Smola, K. R. Müller, M. Scholz, and G. Rätsch, "Kernel pca and de-noising in feature spaces," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 536–542, Cambridge, MA, USA, July 1999.
[43] D. P. Kingma and B. Jimmy, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.
[44] X.-shi Tian and H.-yan Guo, "The application of extreme value theory to risk measurement based on sse-180 index [j]," *Operations Research and Management Science*, vol. 13, no. 1, pp. 106–111, 2004.
[45] M. D. Beneish and J. C. Gardner, "Information costs and liquidity effects from changes in the dow jones industrial average list," *Journal of Financial and Quantitative Analysis*, vol. 30, no. 1, pp. 135–157, 1995.
[46] M. Bryan, "The impact of changes in the ftse 100 index," *Financial Review*, vol. 42, no. 3, pp. 461–484, 2007.
[47] So and Y. Tse, "Price discovery in the hang seng index markets: index, futures, and the tracker fund," *Journal of Futures Markets*, vol. 24, no. 9, pp. 887–907, 2004.
[48] R. Kumar, A. Sarin, and K. Shastri, "The impact of index options on the underlying stocks: the evidence from the listing of nikkei stock average options," *Pacific-Basin Finance Journal*, vol. 3, no. 2-3, pp. 303–317, 1995.
[49] A. Goel, A. Sharma, and A. Mehra, "Index tracking and enhanced indexing using mixed conditional value-at-risk," *Journal of Computational and Applied Mathematics*, vol. 335, pp. 361–380, 2018.