Professor Hongjoong KIM, PhD
E-mail:hongjoong@korea.ac.kr
Mathematics, Korea University

MEAN-VARIANCE PORTFOLIO OPTIMIZATION WITH STOCK RETURN PREDICTION USING XGBOOST

Abstract. *Portfolio optimization is one of the most concerning issues in finance and its success relies on accurate prediction of future stock market, which is challenging due to its dynamic, non-stationary, chaotic and noisy nature. This paper studies the performance of a portfolio optimization model when combined with stock return prediction using a machine learning model. In this study, two portfolio optimization algorithms are proposed. The first algorithm performs the eXtreme Gradient Boosting (XGBoost) for stock return forecasting and the mean-variance (MV) model for portfolio selection. The second algorithm modifies the MV model by introducing an additional penalty term based on the prediction error of XGBoost. The empirical tests using the historical data from 2010 to 2016 of the component stocks of the Korea Composite Stock Price Index show that the proposed algorithms are superior to traditional methods.*

Keywords*: Portfolio optimization, Stock return prediction, XGBoost, Mean-variance model, Machine learning.*

JEL Classification: C53, C63, C18

1. Introduction

Portfolio optimization is the process of selecting the best asset distribution that meets the financial objectives such as maximizing the expected return or minimizing financial risk. The Mean-Variance (MV) model by Markowitz (1952) solves the portfolio optimization problem by forming an efficient frontier, which is a graph showing the set of optimal portfolios that give the highest return for a given level of risk or the lowest risk for a given level of return. The MV model has limitations for practical applications and some improvements are required to solve those issues. In particular, the success of MV model depends on accurate predictions of future stock markets. As (Moon and Kim, 2019) explains, forecasting stock market is a challenging task since it is influenced by many factors and thus it is essentially a nonlinear, non-stationary, dynamic and noisy system.

Recently there have been many machine learning or deep learning approaches for various financial topics such as the algorithmic trading, risk

DOI: 10.24818/18423264/55.4.21.01

assessment, asset pricing and portfolio management. Takeuchi (2013) considered stock selection using an autoencoder composed of stacked restricted Boltzmann machines and achieved high returns. Grace (2017) implemented stock selection using Deep Multilayer Perceptron (DMLP) and then performed portfolio allocation based on the prediction results. (Ince and Trafalis, 2017) combined the independent component analysis and kernel methods to predict the stock market movement. (Fu et al, 2018) constructed a DMLP-based machine learning framework to demonstrate how to apply machine learning algorithms to distinguish good stocks from the bad stocks. (Jujie and Danfeng, 2018) proposed two hybrid models based on the genetic algorithm (GA), grey model (GM), back-propagation neural network (BPNN) and support vector regression (SVR) and performed experimental investigation for stock index prediction. (Lin et al, 2006) used Elman network for optimal portfolio selection which outperformed the Vector Auto Regression (VAR) model and provided the accurate dynamic portfolio selection while Maknickiene (2014) used Evolino Recurrent Neural Network (RNN) for return prediction and portfolio selection. (Heaton et al, 2016) explored the use of deep learning hierarchical models for problems in financial prediction and classification. (Aggarwal and Aggarwal 2017) organized deep investment techniques in financial markets using deep learning models. It introduced deep learning hierarchical decision models for prediction analysis and better decision making for financial domain problem set such as pricing securities, risk factor analysis and portfolio selection.

Batres-Estrada (2015) constructed the deep neural network by combining Deep Belief Network (DBN) and Multilayer Perceptron (MLP), which was used to predict each stock's monthly log-return and to form portfolios. (Lee et al, 2018) first presented a comparative study of simple RNN, Long-Short Term Memory (LSTM) and Gated-Recurrent Unit (GRU), and then built predictive threshold-based portfolios selecting the stocks according to the predictions. (Iwasaki and Chen, 2018) developed a Deep Neural Network (DNN) supervised learning approach to extract insightful topic sentiments from analyst reports at the sentence level and incorporate this qualitative knowledge in asset pricing and portfolio construction. Zhou (2019) utilized deep learning and both the price and fundamental information to separate stocks' winners from losers. Through predicting the next month's return, the LSTM and LSTM-MLP combined neural network produced good monthly returns. (Chen et al, 2016) used a machine-learning approach to forecast hedge fund returns and perform individual hedge fund selection within major hedge fund style categories. Deep Learning (DL) and Random Forest (RF) models showed the best performance. (Jiang and Liang, 2017) presented a portfolio management model constructed by Convolutional Neural Network (CNN) and Deep Reinforcement Learning (DRL) on selected cryptocurrencies to produce portfolio weights for the financial assets. (Jiang et al, 2017) presented a financial-model-free RL framework to provide a deep machine

learning solution to the portfolio management problem and implemented cryptocurrency portfolio management based on RNN, LSTM and CNN. (Liang et al, 2018) implement three continuous Reinforcement Learning (RL) algorithms, Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO) and Policy Gradient (PG) in portfolio management and used RL for portfolio allocation by adjusting the stocks weights using various RL models.

The eXtreme Gradient Boosting (XGBoost) proposed by (Chen and Guestrin 2016) gains much attention from researchers in the field of financial problems in recent years. (Basak et al, 2019) used RF and XGBoost to predict the trend of stock prices. (Chen et al, 2021) developed a hybrid model based on machine learning and the mean-variance model for stock prediction and portfolio selection. An improved firefly algorithm (IFA) is proposed to optimize the hyperparameters. Zolotareva (2021) concentrated on recognizing stock market long-term upward and downward trends using XGBoost method. Table 1 summarizes recent studies about the portfolio management. See (Ozbayoglu et al, 2020) for deep learning studies for various financial applications.

The current study proposes two portfolio management algorithms. The first algorithm uses a hybrid model based on XGBoost and the mean-variance model. XGBoost is used to predict stock returns for the next rebalance date. Once potential returns of stocks are estimated, the weights of assets in the portfolio are computed by employing the MV model. The second model additionally introduces a penalty term in the MV model based on the prediction error. The empirical tests show that two proposed algorithms are superior to traditional approaches.

The remainder of this paper is presented as follows. Section 2 reviews the XGBoost method and the Mean-Variance model. Section 3 proposes two portfolio optimization algorithms and Section 4 reports experimental results. Section 5 draws conclusions.

2.Methods

2.1 XGBoost method

The eXtreme Gradient Boosting (XGBoost) method proposed by (Chen and Guestrin 2016) is an optimized distributed gradient boosting library designed for speed and performance. The mathematical model to predict y_i from the input data x_i can be described by Eq. (1)

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{1}$$

Table 1. Portfolio management studies

Article	Data set	Method
(Lin et al, 2006)	Taiwans stock market	Elman RNN
Takeuchi (2013)	Stocks from NYSE, AMEX, NASDAQ from 1965 to 2009	Autoencoder , RBM
Maknickiene (2014)	FOREX (EUR/USD, etc.), Gold in 2013	Evolino RNN
Batres-Estrada (2015)	S&P500 from 1985 to 2006	DBN, MLP
(Chen et al, 2016)	Hedge fund monthly return data from 1996 to 2015	DMLP
(Heaton et al, 2016)	IBB biotechnology index, stocks from 2012 to 2016	Auto-encoding, Calibrating, Validating, Verifying
(Aggarwal and Aggarwal 2017)	Top 5 companies in S&P500	LSTM, Auto-encoding, Smart indexing
Grace (2017)	20 stocks from S&P500 from 2012 to 2015	DMLP
(Ince and Trafalis, 2017)	Dow-Jones, Nasdaq, S&P500 from 2007 to 2015	ICA, Kernel method, SVM
(Jiang and Liang, 2017)	12 most-volumed cryptocurrency from 2015 to 2016	CNN, RL
(Jiang et al, 2017)	Cryptocurrencies, Bitcoin from 2014 to 2017	CNN, RNN, LSTM
(Fu et al, 2018)	Chinese stock data from 2012 to 2013	LR, RF, DMLP
(Iwasaki and Chen, 2018)	Analyst reports on the TSE and Osaka Exchange from 2016 to 2018	LSTM, CNN, Bi-LSTM
(Jujie and Danfeng, 2018)	SHSE and SZSE from 2012 to 2016	GA, GM, BPNN, SVR
(Lee and Yoo, 2018)	10 stocks in S&P500 from 1997 to 2016	RNN, LSTM, GRU
(Liang et al, 2018)	Stocks from Chinese/American stock market from 2015 to 2018	DDPG, PPO
(Basak et al, 2019)	10 companies till 2017	RF, XGBoost
Zhou (2019)	Stocks in NYSE, AMEX, NASDAQ, TAQ intraday trade from 1993 to 2017	LSTM, DMLP
(Chen et al, 2021)	24 stocks in Shanghai Stock Exchange 50 index from 2009 to 2019	IFA
Zolotareva (2021)	The datasets from 2005 to 2017	XGBoost
Proposed approach	Top 15 stocks in Korea KOSPI 200 index from 2010 to 2016	XGBoost, MV

where f is a tree in the regression tree space F and K is the number of trees. The objective function to be optimized is given by Eq. (2)

$$obj^{(t)} = \sum_i L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_i \Omega(f_i) \ , \tag{2}$$

where $L\left(y_i, \hat{y}_i^{(t)}\right)$ is the training loss function and $\Omega(f_i)$ is the regularization. Since it is difficult to learn the parameters of all the trees at once, an additive strategy is used to fix what we have learned and add one new tree at a time. We then have

$$\begin{aligned} &\hat{y}_i^{(0)} = 0 \\ &\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ &\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ &\vdots \\ &\hat{y}_i^{(t)} = \sum_{i=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \end{aligned} \tag{3}$$

If the tree that optimizes our objective is added at each step and if the Taylor expansion of the loss function up to the second order is used, the specific objective at step t can be written as Eq. (4)

$$obj^{(t)} = \sum_i \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t), \tag{4}$$

where $g_i = \partial_{\hat{y}_i^{(t-1)}} L\left(y_i, \hat{y}_i^{(t-1)}\right)$ and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} L\left(y_i, \hat{y}_i^{(t-1)}\right)$, respectively. The regularization $\Omega(f)$ is defined as Eq. (5)

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \ , \tag{5}$$

where $w \in R^T$ is the vector of scores on leaves and T is the number of leaves. γ and λ are L1 and L2 regularization coefficients, respectively.

2.2 Modern-Variance model

Markowitz is the pioneer of Modern portfolio theory (MPT). MPT is a theory about how rational investors construct portfolios to maximise the expected returns for given levels of risk or to minimize the risks for given levels of return. The Mean-Variance (MV) model in Markowitz (1952) presents a mathematical optimization problem (6) between return maximization and risk minimization:

$$\begin{aligned} &\min \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j \sigma_{ij} \\ &\max \sum_{i=1}^{n} x_i \mu_i \\ &s.t. \sum_{i=1}^{n} x_i = 1, 0 \le x_i \le 1, i = 1, \dots, n \end{aligned} \tag{6}$$

where x_i is the proportion of asset i in portfolio, σ_{ij} is the covariance between asset i and asset j, and μ_i is the expected return on asset i.

3. Portfolio optimization algorithms

Suppose that one performs portfolio optimization at day T. If the expected return and the risk up to T are all known, the Modern Portfolio Theory can be applied to find the optimal portfolio which gives the lowest risk for a given level of expected return. There are, however, some situations at which one wants to find the optimal portfolio for day $T + h$ in the future at day T. Forming an efficient frontier requires the expected return and the risk at day $T + h$, but the values for $(T, T + h]$

are the values in the future and thus *unknown* as in Figure 1. Hence the MPT by Markowitz requires modification in that case.

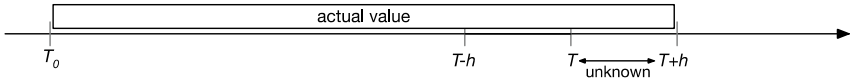

Figure 1. Time interval for MPT including *unknown* values

One way to resolve the situation is to combine the known values up to T with the *predicted* values for $(T, T+h]$. In this study, the XGBoost method is used for the prediction. First, the partition $t, t+dt, t+2dt, \dots, t+ndt$ in $(T, T+h]$ of stepsize dt are introduced. Then, XGBoost is trained and validated with the data in $[t-T_0, t-h]$ for a sufficiently large T_0 to predict the expected return and the covariance at t as in Figure 2 (Top). Once the prediction is completed, XGBoost is trained and validated with the data in $[t-T_0+dt, t-h+dt]$ to predict the expected return and the covariance at $t+dt$ as in Figure 2 (Middle). Such procedure is repeated until the expected returns and the covariances at all points in the partition $t, t+dt, t+2dt, \dots, t+ndt \in (T, T+h]$ are obtained.

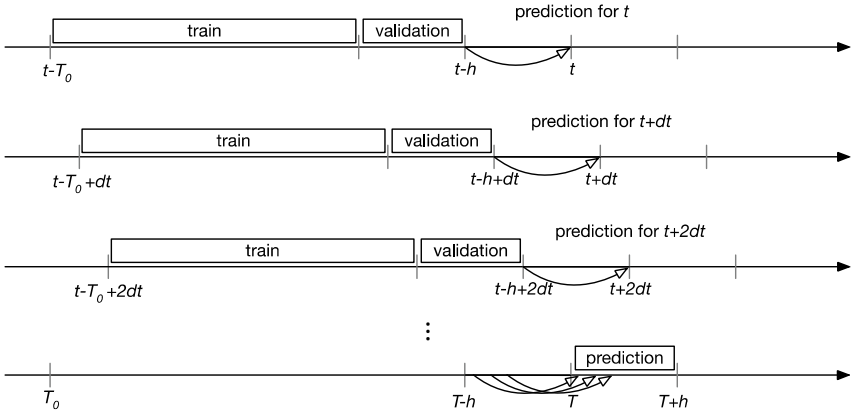

Figure 2. Predictions of expected returns and covariances for $(T, T+h]$

Once those predictions are completed, one can perform MV analysis with the combined values of the actual value for $[T_0, T]$ and the predicted values for $(T, T+h]$ as in Figure 3 and estimate appropriate weights of the assets in the portfolio. For the time during which the portfolio is managed, above procedure is repeated at each rebalancing day. Such a method will be called MPT-XGBoost in this study and Figure 4 shows the outline of the algorithm.

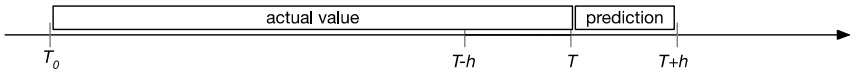

Figure 3. Time interval for XGBoost -based MPT

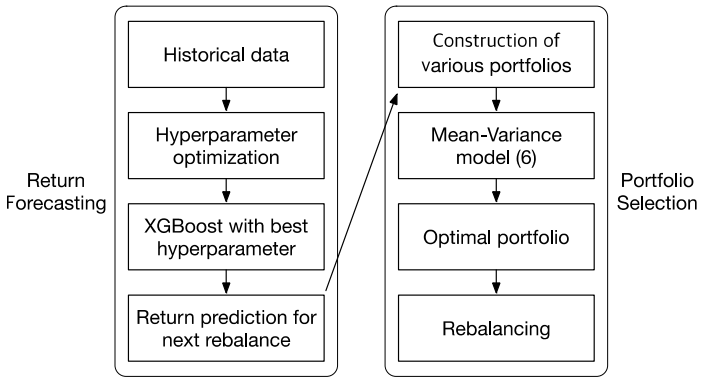

Figure 4. Outline of the MPT-XGBoost algorithm

When XGBoost is used for the prediction, the prediction error can be estimated from the validation procedure. Since the inclusion of assets of large prediction errors may increase the risk of the portfolio, following modification of the MV model can be considered:

$$\begin{aligned} &\min \sum_{i=1}^{n}\sum_{j=1}^{n}(1-\epsilon_i)x_i(1-\epsilon_j)x_j\sigma_{ij} \\ &\max \sum_{i=1}^{n}(1-\epsilon_i)x_i\mu_i \\ &s.t.\sum_{i=1}^{n}x_i = 1, 0 \le x_i \le 1, i = 1, \dots, n \end{aligned} \tag{7}$$

where ϵ_i is the prediction error for the i^{th} stock during the validation. In this study, the prediction error measured by the Root Mean Square Error (RMSE) defined in Eq. (8) is used to define ϵ_i. One can perform the modified MV model (7) instead of the MV model (6) and then estimate appropriate weights of the assets in the portfolio. Such a method will be called MPT-XGBoost-RMSE in this study. Since the computational reliability is additionally considered, the MPT-XGBoost-RMSE algorithm is expected to improve the portfolio optimization, which is numerically supported in Section 4.3.

4. Empirical tests

In this study, the algorithms explained above are validated with the component stocks in Korea Composite Stock Price Index. Top 15 stocks in market capitalization in Table 2 are used. Table 3 shows their statistical characteristics.

Table 2. Stocks used in the empirical tests

Samsung Electronics Co., Ltd.	SK hynix, Inc.	LG Chem, Ltd.	NAVER Corporation	Hyundai Motor Company
Samsung SDI Co., Ltd.	Kia Corporation	Hyundai Mobis Co., Ltd.	LG Electronics Inc.	LG Household & Health Care Ltd.
SK Holdings Co., Ltd.	POSCO	Ncsoft Corporation	SK Telecom Co., Ltd.	KB Financial Group Inc.

Table 3. Statistics of the financial data

Stock	count	mean	std	min	25%	50%	75%	max
Samsung Electronics Co., Ltd.	1666	23661.18	4920.92	13600.00	18705.00	25260.00	27295.00	33740.00
SK hynix, Inc.	1666	31286.91	8323.70	15600.00	24712.50	28850.00	36400.00	51900.00
LG Chem, Ltd.	1666	302712.79	67914.58	164500.00	261000.00	291000.00	327500.00	567000.00
NAVER Corporation	1666	100745.74	35957.39	50538.00	66861.00	89933.00	134190.00	175849.00
Hyundai Motor Company	1666	190665.07	42182.18	101500.00	151500.00	197000.00	228000.00	268500.00
Samsung SDI Co., Ltd.	1666	141581.09	24533.80	76800.00	124000.00	142000.00	159875.00	201000.00
Kia Corporation	1666	54798.77	14027.03	18550.00	46700.00	55000.00	64600.00	83800.00
Hyundai Mobis Co.,Ltd	1666	267990.70	47568.46	141500.00	243625.00	271500.00	297000.00	414500.00
LG Electronics Inc.	1666	74443.47	18929.83	39800.00	60825.00	71000.00	84525.50	125091.00
LG Household & Health Care Ltd.	1666	594238.30	205562.30	270000.00	454125.00	551500.00	660000.00	1181000.00
SK Holdings Co., Ltd	1666	150324.01	65002.37	47250.00	98025.00	128500.00	215000.00	320500.00
POSCO	1666	345253.30	104385.56	156000.00	275750.00	330500.00	416000.00	625000.00
Ncsoft Corporation	1666	216921.07	53488.32	120500.00	175625.00	211500.00	248500.00	380500.00
SK Telecom Co.,Ltd	1666	197788.42	45066.86	120500.00	160000.00	196750.00	228375.00	301000.00
KB Financial Group Inc.	1666	41152.37	7398.61	28300.00	36100.00	38500.00	43637.50	62100.00

The experimental results are presented in three aspects:

1. the accuracy of XGBoost for the prediction of stock returns
2. returns and risks of optimal portfolios from several optimization methods
3. the results from long-term portfolio managements

4.1 The accuracy of stock return prediction

XGBoost is implemented using Scikit-learn by (Pedregosa *et al*. 2011). 1260 daily returns from 1/5/2010 to 2/2/2015 are used for training of the XGBoost and 252 returns from 2/3/2015 to 2/11/2016 are used for the validation. Then the test is performed using 154 daily returns from 2/12/2016 to 9/27/2016. On each day, 10 previous values are given as the feature. Table 4 shows the hyperparameters for the XGBoost algorithm considered in the experiments.

Table 4. Hyperparameters for the XGBoost

Hyperparameter	Usage	Value
n_estimators	Number of gradient boosted trees	$1, 3, 5, \dots, 59$
max_depth	Maximum tree depth for base learners	$2, 3, 4, \dots, 9$
learning_rate	Boosting learning rate	$0.0001, 0.001, 0.005, 0.01, 0.05,$ $0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$
gamma	Minimum loss reduction required to make a further partition on a leaf node of the tree	$0, 0.1, 0.2, \dots, 1.0$
min_child_weight	Minimum sum of instance weight(hessian) needed in a child	$3, 4, 5, \dots, 14$
subsample	Subsample ratio of the training instance	$0.1, 0.2, 0.3, \dots, 1.0$
colsample_bytree	Subsample ratio of columns when constructing each tree	$0.1, 0.2, 0.3, \dots, 1.0$
colsample_bylevel	Subsample ratio of columns for each level	$0.1, 0.2, 0.3, \dots, 1.0$

The prediction horizon h is set to 14 days, i.e. the stock return at day $t + 14$ is predicted at day t. The prediction errors at day t are estimated in terms of MAE

and RMSE in Eq. (8):

$$\text{MAE} = \frac{1}{h}\sum_{i=0}^{h-1}|y_{t+i} - \hat{y}_{t+i}| \quad \text{and} \quad \text{RMSE} = \sqrt{\frac{1}{h}\sum_{i=0}^{h-1}(y_{t+i} - \hat{y}_{t+i})^2} \tag{8}$$

Portfolio is rebalanced every 14 days in this study and Table 5 and Table 6 show the MAE and RMSE errors, respectively, in forecasting stock returns at those rebalancing days. Note that both errors are small and alike for all stocks.

Table 5. MAE errors in return prediction

Stock	2016-03-03	2016-03-23	2016-04-12	2016-05-03	2016-05-25	2016-06-15
Samsung Electronics Co., Ltd.	0.92	1.30	0.92	0.91	1.33	1.10
SK hynix, Inc.	1.32	1.51	1.23	1.52	1.68	2.07
LG Chem, Ltd.	0.94	1.68	1.66	1.57	1.16	1.95
NAVER Corporation	0.84	1.58	1.16	1.03	1.31	1.42
Hyundai Motor Company	1.14	1.17	1.56	1.34	0.90	1.16
Samsung SDI Co., Ltd.	1.20	1.70	1.70	1.25	2.90	1.95
Kia Corporation	1.52	1.32	1.18	1.04	0.84	0.90
Hyundai Mobis Co.,Ltd	0.96	1.09	0.94	1.28	0.87	1.65
LG Electronics Inc.	1.10	1.23	1.35	1.65	1.20	1.64
LG Household & Health Care Ltd.	1.93	1.92	1.39	0.95	1.29	1.46
SK Holdings Co., Ltd	1.82	1.23	1.67	1.50	1.28	1.04
POSCO	1.91	1.55	1.88	1.79	1.75	1.66
Ncsoft Corporation	2.40	1.70	1.69	1.65	1.28	1.62
SK Telecom Co.,Ltd	1.03	0.82	1.54	0.76	1.27	0.77
KB Financial Group Inc.	1.12	0.88	1.40	0.86	1.29	1.44

Stock	2016-07-05	2016-07-25	2016-08-12	2016-09-02	2016-09-27	mean
Samsung Electronics Co., Ltd.	1.18	1.36	1.64	2.03	1.59	1.30
SK hynix, Inc.	1.33	1.77	1.24	1.41	1.43	1.50
LG Chem, Ltd.	1.67	1.77	1.00	1.85	1.29	1.50
NAVER Corporation	1.32	1.05	1.01	1.40	1.67	1.25
Hyundai Motor Company	1.30	1.61	1.06	0.98	1.38	1.24
Samsung SDI Co., Ltd.	1.46	1.46	1.97	1.75	1.26	1.69
Kia Corporation	0.91	1.05	0.90	1.03	0.83	1.05
Hyundai Mobis Co.,Ltd	1.54	1.51	0.97	1.61	1.39	1.25
LG Electronics Inc.	1.31	1.44	1.05	1.32	1.95	1.39
LG Household & Health Care Ltd.	1.53	2.40	1.89	0.79	1.73	1.57
SK Holdings Co., Ltd	0.87	1.23	1.07	1.19	1.03	1.27
POSCO	2.12	1.49	1.07	1.15	1.26	1.60
Ncsoft Corporation	0.70	2.23	1.20	1.16	1.24	1.53
SK Telecom Co.,Ltd	0.82	0.83	0.55	0.80	0.83	0.91
KB Financial Group Inc.	1.32	1.37	1.05	1.07	1.17	1.18

Table 6. RMSE errors in return prediction

Stock	2016-03-03	2016-03-23	2016-04-12	2016-05-03	2016-05-25	2016-06-15
Samsung Electronics Co., Ltd.	1.24	1.66	1.23	1.12	1.56	1.36
SK hynix, Inc.	1.81	1.78	2.10	1.98	2.30	2.28
LG Chem, Ltd.	1.33	1.99	1.89	2.04	1.35	2.84
NAVER Corporation	1.42	2.07	1.44	1.40	1.79	1.97
Hyundai Motor Company	1.50	1.53	1.81	1.59	1.22	1.38
Samsung SDI Co., Ltd.	1.50	1.82	2.30	1.69	3.74	2.30
Kia Corporation	2.00	1.59	1.51	1.22	1.10	1.14
Hyundai Mobis Co.,Ltd	1.30	1.62	1.27	1.64	1.25	2.07
LG Electronics Inc.	1.37	1.52	1.60	2.26	1.82	2.14
LG Household & Health Care Ltd.	2.36	2.43	1.64	1.21	1.57	1.70
SK Holdings Co., Ltd	2.20	1.55	1.95	1.95	1.54	1.41
POSCO	2.40	2.19	2.55	2.44	2.53	2.26
Ncsoft Corporation	3.40	2.06	2.05	2.04	1.60	1.88
SK Telecom Co.,Ltd	1.22	0.96	1.88	1.02	1.41	1.11
KB Financial Group Inc.	1.35	1.04	1.79	1.09	1.66	1.88

Stock	2016-07-05	2016-07-25	2016-08-12	2016-09-02	2016-09-27	mean
Samsung Electronics Co., Ltd.	1.51	1.56	1.93	2.83	2.63	1.69
SK hynix, Inc.	1.78	2.22	1.45	1.99	1.84	1.96
LG Chem, Ltd.	2.21	2.24	1.25	2.35	1.56	1.92
NAVER Corporation	1.76	1.45	1.23	1.71	2.11	1.67
Hyundai Motor Company	1.68	2.07	1.25	1.44	1.61	1.55
Samsung SDI Co., Ltd.	1.84	1.98	2.58	2.24	1.54	2.14
Kia Corporation	1.21	1.39	1.05	1.42	1.11	1.34
Hyundai Mobis Co.,Ltd	1.78	1.93	1.29	2.07	1.52	1.61
LG Electronics Inc.	1.53	1.71	1.19	1.56	2.50	1.75
LG Household & Health Care Ltd.	2.00	3.47	2.48	0.94	2.45	2.02
SK Holdings Co., Ltd	1.15	1.64	1.55	1.48	1.58	1.64
POSCO	2.69	1.85	1.32	1.56	1.57	2.12
Ncsoft Corporation	0.96	2.82	1.52	1.40	1.41	1.92
SK Telecom Co.,Ltd	0.97	1.11	0.66	0.95	1.06	1.12
KB Financial Group Inc.	1.57	1.74	1.45	1.23	1.28	1.46

4.2 Returns and risks of the optimal portfolios

Four types of MV models are evaluated and compared. First, the MV model by Markowitz is used, which utilizes the actual but *unknown* values of the return and risk for $(t, t+h]$ at day t, and the result is denoted by *actual*. The result from equal weight that gives the same weight to each stock in the portfolio is denoted by $1/N$. *XG* represents the result from the MPT-XGBoost model using the known values of the return and risk up to t and predicted values for $(t, t+h]$ at day t. The result from the MPT-XGBoost-RMSE model is denoted by *RMSE*.

At each rebalancing day, 10000 portfolios are generated in the MV model by Markowitz to form an efficient frontier, which gives the highest expected return for a defined level of risk. Each yellow circle in Figure 5 represents the result from each portfolio. The blue circle in Figure 5 (Left) represents the optimal portfolio with the highest Sharpe ratio when risk-free rate is 0.01 and the blue circle in Figure 5 (Right) represents the optimal portfolio with risk-free rate 0.02.

Remark. Note that those yellow and blue circles are the results from the *actual* MV model, which utilizes unknown values in the future. Thus, the blue circles are the optimal portfolios but not available in practice. They are given for the comparison purpose only and they are the best values that MPT-XGBoost and MPT-XGBoost-RMSE models can take.

The black triangle represents the portfolio from the $1/N$ model. The green square and the red star are the results from the XG and RMSE models, respectively. Note that XG and RMSE are quite close to the optimal but unknown *actual*, while $1/N$ is not close enough. The results for different dates are similar and omitted.

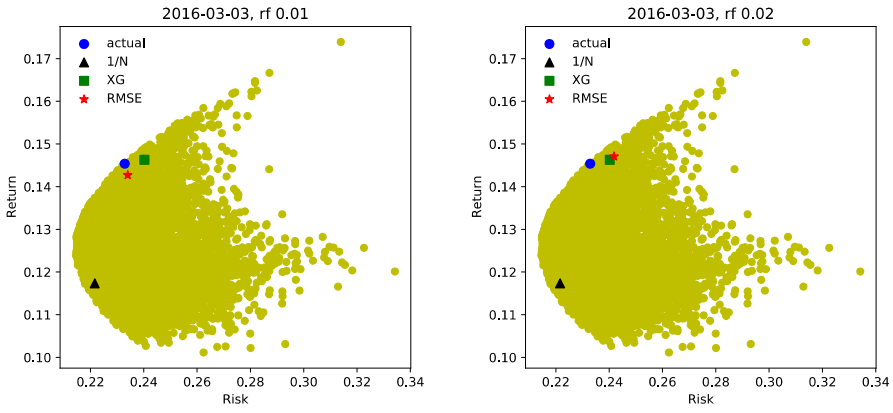

Figure 5. Optimal portfolios for the risk-free rate 0.01 (Left) and 0.02 (Right)

Table 7. Parameters used in the experiments

Parameter	Value
Types of MV model	actual, 1/N, XG, RMSE
Risk-free rate	$0.01, 0.02$
Number of stocks in the portfolio	$5, 10, 15$

Table 8 shows the returns and volatilities at each rebalancing day for the minimum variance portfolio, which provides the lowest variance among all possible portfolios of risky assets. Table 8 (Top) shows the results when the portfolio consists of 5 stocks. Table 8 (Middle) and Table 8 (Bottom) represent the results when the portfolios are constructed with 10 and 15 stocks, respectively. Note that the values from XG and RMSE are quite close to those of *actual* in all cases while the volatilities from $1/N$ are relatively large as observed in Figure 5.

Table 8. Returns and volatilities for minimum variance portfolio

5 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.124	0.215	0.117	0.222	0.117	0.213	0.117	0.213
2016-03-23	0.132	0.213	0.125	0.221	0.123	0.213	0.124	0.213
2016-04-12	0.142	0.212	0.147	0.220	0.147	0.212	0.149	0.212
2016-05-03	0.133	0.212	0.135	0.220	0.136	0.211	0.137	0.211
2016-05-25	0.122	0.211	0.124	0.219	0.127	0.211	0.129	0.211
2016-06-15	0.123	0.210	0.109	0.218	0.116	0.210	0.118	0.210
2016-07-05	0.125	0.210	0.112	0.218	0.122	0.210	0.125	0.209
2016-07-25	0.131	0.209	0.130	0.217	0.137	0.209	0.140	0.209
2016-08-12	0.126	0.209	0.110	0.217	0.121	0.209	0.126	0.209
2016-09-02	0.137	0.209	0.128	0.217	0.137	0.209	0.139	0.209
2016-09-27	0.133	0.209	0.127	0.216	0.134	0.208	0.138	0.208

10 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.124	0.185	0.108	0.196	0.134	0.184	0.129	0.184
2016-03-23	0.134	0.184	0.121	0.195	0.132	0.184	0.140	0.185
2016-04-12	0.140	0.183	0.135	0.194	0.145	0.183	0.149	0.184
2016-05-03	0.139	0.183	0.125	0.194	0.138	0.183	0.138	0.184
2016-05-25	0.121	0.183	0.118	0.193	0.129	0.182	0.130	0.184
2016-06-15	0.121	0.182	0.103	0.192	0.117	0.181	0.113	0.183
2016-07-05	0.124	0.182	0.103	0.192	0.121	0.181	0.138	0.183
2016-07-25	0.129	0.181	0.114	0.192	0.135	0.181	0.149	0.183
2016-08-12	0.118	0.181	0.096	0.191	0.120	0.180	0.135	0.183
2016-09-02	0.121	0.180	0.107	0.191	0.124	0.180	0.135	0.182
2016-09-27	0.121	0.180	0.102	0.190	0.121	0.179	0.130	0.181

15 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.114	0.159	0.095	0.170	0.108	0.158	0.108	0.163
2016-03-23	0.114	0.158	0.104	0.170	0.114	0.158	0.129	0.163
2016-04-12	0.124	0.157	0.122	0.169	0.128	0.157	0.147	0.162
2016-05-03	0.120	0.157	0.111	0.168	0.119	0.156	0.136	0.161
2016-05-25	0.107	0.156	0.103	0.168	0.113	0.156	0.126	0.161
2016-06-15	0.108	0.156	0.093	0.168	0.107	0.156	0.119	0.161
2016-07-05	0.107	0.157	0.091	0.167	0.107	0.156	0.116	0.161
2016-07-25	0.112	0.156	0.100	0.167	0.114	0.155	0.123	0.161
2016-08-12	0.102	0.155	0.088	0.167	0.102	0.155	0.111	0.161
2016-09-02	0.102	0.155	0.096	0.167	0.106	0.155	0.111	0.161
2016-09-27	0.096	0.155	0.090	0.166	0.097	0.154	0.106	0.160

Table 9 shows the returns and volatilities for the optimal portfolio, at which the average return earned in excess of the risk-free rate per unit of volatility is maximized, when the risk-free rate is 0.01 and Table 10 shows the results when the risk-free rate is 0.02. It is observed again that the expected returns and volatilities from XG and RMSE are close to those of *actual*.

Table 9. Returns and volatilities for optimal portfolios with $rf = 0.01$

5 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.145	0.233	0.117	0.222	0.146	0.240	0.143	0.234
2016-03-23	0.152	0.229	0.125	0.221	0.145	0.230	0.149	0.237
2016-04-12	0.177	0.237	0.147	0.220	0.173	0.230	0.175	0.232
2016-05-03	0.156	0.227	0.135	0.220	0.162	0.232	0.160	0.228
2016-05-25	0.155	0.236	0.124	0.219	0.156	0.235	0.154	0.232
2016-06-15	0.155	0.236	0.109	0.218	0.152	0.235	0.161	0.250
2016-07-05	0.170	0.238	0.112	0.218	0.166	0.237	0.168	0.240
2016-07-25	0.182	0.236	0.130	0.217	0.185	0.236	0.185	0.240
2016-08-12	0.178	0.236	0.110	0.217	0.173	0.236	0.173	0.241
2016-09-02	0.188	0.236	0.128	0.217	0.186	0.236	0.183	0.238
2016-09-27	0.193	0.241	0.127	0.216	0.193	0.239	0.197	0.248

5 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.172	0.200	0.108	0.196	0.173	0.199	0.176	0.198
2016-03-23	0.185	0.199	0.121	0.195	0.181	0.199	0.184	0.198
2016-04-12	0.191	0.199	0.135	0.194	0.196	0.199	0.197	0.198
2016-05-03	0.189	0.199	0.125	0.194	0.183	0.198	0.187	0.197
2016-05-25	0.171	0.198	0.118	0.193	0.179	0.198	0.180	0.196
2016-06-15	0.172	0.197	0.103	0.192	0.171	0.197	0.170	0.195
2016-07-05	0.172	0.197	0.103	0.192	0.171	0.197	0.170	0.194
2016-07-25	0.166	0.189	0.114	0.192	0.171	0.189	0.180	0.198
2016-08-12	0.154	0.189	0.096	0.191	0.156	0.188	0.160	0.194
2016-09-02	0.157	0.189	0.107	0.191	0.160	0.188	0.163	0.197
2016-09-27	0.163	0.194	0.102	0.190	0.156	0.187	0.166	0.197

5 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.173	0.175	0.095	0.170	0.170	0.174	0.189	0.183
2016-03-23	0.172	0.174	0.104	0.170	0.172	0.174	0.188	0.182
2016-04-12	0.180	0.173	0.122	0.169	0.187	0.173	0.198	0.175
2016-05-03	0.167	0.169	0.111	0.168	0.173	0.173	0.185	0.175
2016-05-25	0.155	0.169	0.103	0.168	0.160	0.169	0.175	0.174
2016-06-15	0.158	0.169	0.093	0.168	0.159	0.172	0.171	0.174
2016-07-05	0.156	0.169	0.091	0.167	0.154	0.168	0.166	0.174
2016-07-25	0.159	0.168	0.100	0.167	0.162	0.168	0.171	0.173
2016-08-12	0.148	0.168	0.088	0.167	0.149	0.167	0.158	0.174
2016-09-02	0.151	0.167	0.096	0.167	0.153	0.167	0.159	0.173
2016-09-27	0.147	0.167	0.090	0.166	0.146	0.166	0.153	0.173

Table 10. Returns and volatilities for optimal portfolios with $rf = 0.02$

5 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.145	0.233	0.117	0.222	0.146	0.240	0.147	0.242
2016-03-23	0.155	0.234	0.125	0.221	0.150	0.239	0.149	0.237
2016-04-12	0.177	0.237	0.147	0.220	0.173	0.230	0.175	0.232
2016-05-03	0.164	0.238	0.135	0.220	0.162	0.232	0.160	0.228
2016-05-25	0.159	0.243	0.124	0.219	0.156	0.235	0.159	0.239
2016-06-15	0.155	0.236	0.109	0.218	0.152	0.235	0.161	0.250
2016-07-05	0.170	0.238	0.112	0.218	0.166	0.237	0.168	0.240
2016-07-25	0.182	0.236	0.130	0.217	0.185	0.236	0.185	0.240
2016-08-12	0.178	0.236	0.110	0.217	0.173	0.236	0.173	0.241
2016-09-02	0.188	0.236	0.128	0.217	0.186	0.236	0.183	0.238
2016-09-27	0.193	0.241	0.127	0.216	0.193	0.239	0.197	0.248

5 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.172	0.200	0.108	0.196	0.173	0.199	0.176	0.198
2016-03-23	0.185	0.199	0.121	0.195	0.181	0.199	0.184	0.198
2016-04-12	0.191	0.199	0.135	0.194	0.196	0.199	0.197	0.198
2016-05-03	0.189	0.199	0.125	0.194	0.183	0.198	0.187	0.197
2016-05-25	0.171	0.198	0.118	0.193	0.179	0.198	0.180	0.196
2016-06-15	0.172	0.197	0.103	0.192	0.171	0.197	0.170	0.195
2016-07-05	0.172	0.197	0.103	0.192	0.171	0.197	0.170	0.194
2016-07-25	0.166	0.189	0.114	0.192	0.171	0.189	0.180	0.198
2016-08-12	0.154	0.189	0.096	0.191	0.156	0.188	0.160	0.194
2016-09-02	0.162	0.195	0.107	0.191	0.160	0.188	0.163	0.197
2016-09-27	0.163	0.194	0.102	0.190	0.161	0.194	0.166	0.197

5 assets	actual		1/N		XG		RMSE	
	return	volatility	return	volatility	return	volatility	return	volatility
2016-03-03	0.173	0.175	0.095	0.170	0.170	0.174	0.189	0.183
2016-03-23	0.172	0.174	0.104	0.170	0.172	0.174	0.188	0.182
2016-04-12	0.180	0.173	0.122	0.169	0.187	0.173	0.198	0.175
2016-05-03	0.171	0.173	0.111	0.168	0.173	0.173	0.185	0.175
2016-05-25	0.158	0.173	0.103	0.168	0.160	0.169	0.175	0.174
2016-06-15	0.158	0.169	0.093	0.168	0.159	0.172	0.171	0.174
2016-07-05	0.156	0.169	0.091	0.167	0.154	0.168	0.166	0.174
2016-07-25	0.159	0.168	0.100	0.167	0.162	0.168	0.171	0.173
2016-08-12	0.148	0.168	0.088	0.167	0.149	0.167	0.158	0.174
2016-09-02	0.151	0.167	0.096	0.167	0.153	0.167	0.159	0.173
2016-09-27	0.147	0.167	0.090	0.166	0.146	0.166	0.153	0.173

4.3 The results from long-term portfolio managements

The long-term portfolio management is performed from 2/12/2016 to 9/27/2016 based on the results in Section 4.1 and 4.2. Figure 6 (Left) shows the result of the portfolio management when the portfolio is constructed with 5 stocks and $rf = 0.01$. The blue line represents the change of the portfolio value in time based on the *actual* MV model, which is optimal but uses *unknown* values in the future. Both XG and RMSE are quite close to *actual* while the orange line for $1/N$ is not. Figure 6 (Center) shows the results when the portfolio consists of 10 stocks and the results are quite similar to those in Figure 6 (Left). The results when the number of stocks in the portfolio is 15 are shown in Figure 6 (Right). Even though both XG and RMSE still show similar trends to *actual*, XG slightly deviates from *actual* and RMSE in Figure 6 (Right) and the final value of XG-based long-term management is lower than those of *actual* and RMSE. It is observed that $1/N$ model shows inconsistent pattern as the size of the portfolio changes.

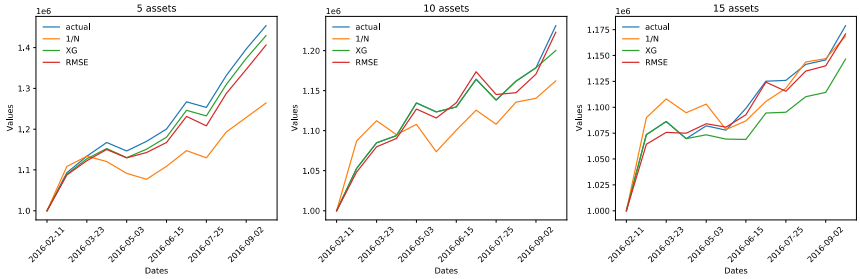

Figure 6. The values of the portfolios in time when $rf = 0.01$ and the number of stocks in the portfolio is (Left) 5, (Center) 10 and (Right) 15

The results when the risk-free rate is $rf = 0.02$ are shown in Figure 7. Similarly to Figure 6, $1/N$ seems to be an inappropriate choice. Both XG and RMSE are close to *actual* but XG is inferior to RMSE when the size of the portfolio is large. RMSE produces accurate results for all three sizes of the portfolio.

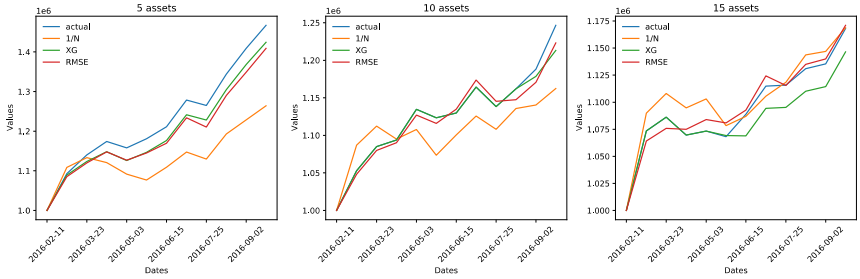

Figure 7. The values of the portfolios in time when $rf = 0.02$ and the number of stocks in the portfolio is (Left) 5, (Center) 10 and (Right) 15

5. Conclusions

This study proposes novel approaches to find optimal portfolios using XGBoost-based prediction and modern portfolio theory, which can capture the future characteristics of stock markets and form an efficient frontier properly. When the prediction error is additionally implemented as a penalty term in the mean-variance model, the long-term portfolio management becomes improved and the value of the portfolio approaches the theoretically optimal value.

The Modern Portfolio Theory emphasizes diversification to improve returns and reduce risks and thus the current study can be improved in several directions. First, the economic cycle or short-term opportunities in the market can be considered to adopt strategic asset allocation and tactical asset allocation in portfolio management. Secondly, the current study uses the MV model for portfolio selection and other portfolio selection schemes such as VaR or CVar can be considered. Various types of financial variables and the corresponding scale-up problems need to be also included.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(2017R1D1A1B03035543).

REFERENCES

[1] **Aggarwal, S., Aggarwal, S. (2017),** ***Deep Investment in Financial Markets using Deep Learning Models***. *International Journal of Computer Applications*, 162, 40-43;

[2] **Basak, S., Kar, S., Saha, S., Khaidem, L., Dey, S.R. (2019),** ***Predicting the Direction of Stock Market Prices Using Tree-Based Classifiers***. *The North American Journal of Economics and Finance, 47, 552-567*;

[3] **Batres-Estrada, B. (2015),** ***Deep Learning for Multivariate Financial Time Series***; (Master's thesis), *KTH, Mathematical Statistics;*

[4] **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*, 785-794;

[5] **Chen, J., Tindall, M.L., Wu, W. (2016),** ***Hedge Fund Return Prediction and Fund Selection: A Machine-Learning Approach***; *Occasional Papers 16-4, Federal Reserve Bank of Dallas;*

[6] **Chen, W., Zhang, H., Mehlawat, M.K., Jia, L. (2021),** ***Mean-variance Portfolio Optimization Using Machine Learning-based Stock Price Prediction***. *Applied Soft Computing Journal*, 100:106943;

[7] **Fu, X., Du, J., Guo, Y., Liu, M., Dong, T., Duan, X. (2018),** ***A Machine Learning Framework for Stock Selection***. ArXiv, abs/1806.01743;

[8] **Grace, A. (2017),** ***Can Deep Learning Techniques Improve the Risk Adjusted Returns from Enhanced Indexing Investment Strategies***;

[9] **Heaton, J.B., Polson, N.G., Witte, J. (2016),** ***Deep Learning for Finance: Deep Portfolios.*** *Econometric Modeling: Capital Markets - Portfolio Theory eJournal*;

[10] **Ince, H., Trafalis, T. (2017),** ***A Hybrid Forecasting Model for Stock Market Prediction***. *Economic Computation and Economic Cybernetics Studies and Research*, 51, 263-280; *ASE Publishing;*

[11] **Iwasaki, H., Chen, Y. (2018),** ***Topic Sentiment Asset Pricing with DNN Supervised Learning***. *Econometric Modeling: Capital Markets - Asset Pricing eJournal;*

[12] **Jiang, Z., Liang, J. (2017),** ***Cryptocurrency Portfolio Management with Deep Reinforcement Learning***. *2017 Intelligent Systems Conference (IntelliSys)*, 905-913;

[13] **Jiang, Z., Xu, D., Liang, J. (2017),** ***A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem***; *arXiv preprint arXiv*:1706.10059;

[14] **Jujie, W., Danfeng, Q. (2018),** ***An Experimental Investigation of Two Hybrid Frameworks for Stock Index Prediction Using Neural Network and Support Vector Regression***. *Economic Computation and Economic Cybernetics Studies and Research*, 52, 193-210; *ASE Publishing;*

[15] **Lee, S., Yoo, S.J. (2018),** ***Threshold-based Portfolio: The Role of the Threshold and its Applications***. *The Journal of Supercomputing*, 1-18;

[16] **Liang, Z., Jiang, K., Chen, H., Zhu, J., Li, Y. (2018),** ***Adversarial Deep Reinforcement Learning in Portfolio Management***. *arXiv: Portfolio Management;*

[17] **Lin, C., Huang, J., Gen, M., Tzeng, G. (2006),** ***Recurrent Neural Network for Dynamic Portfolio Selection***. *Appl. Math. Comput*., 175, 1139-1146;

[18] **Maknickiene, N. (2014),** ***Selection of Orthogonal Investment Portfolio Using Evolino RNN Trading Model***. *Procedia - Social and Behavioral Sciences*, 110, 1158-1165;

[19] **Markowitz, H. (1952),** ***Portfolio Selection***. *Journal of finance*, 7:77–91;
[20] **Moon, K., Kim, H. (2019),** ***Performance of Deep Learning in Prediction of Stock Market Volatility****, Economic Computation & Economic Cybernetics Studies & Research*, 53(2):77-92; *ASE Publishing:*
[21] **Ozbayoglu, A.M., Gudelek, M.U., Sezer, O.B. (2020),** ***Deep Learning for Financial Applications : A Survey***. *Appl. Soft Comput*., 93, 106384;
[22] **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011),** ***Scikit-learn: Machine Learning in Python****; Journal of Machine Learning Research*, 12:2825-2830;
[23] **Takeuchi, L. (2013),** ***Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks***;
[24] **Zhou, B. (2019);** ***Deep Learning and the Cross-Section of Stock Returns: Neural Networks Combining Price and Fundamental Information***. *Neuroeconomics eJournal*;
[25] **Zolotareva, E. (2021),** ***Aiding Long-Term Investment Decisions with XGBoost Machine Learning Model***. *ArXiv, abs*/2104.09341.