Engineering Applications of Artificial Intelligence 120 (2023) 105843

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Prediction based mean-value-at-risk portfolio optimization using machine learning regression algorithms for multi-national stock markets

Jyotirmayee Behera [a], Ajit Kumar Pasayat [b], Harekrushna Behera [a], Pankaj Kumar [a,*]

[a] Department of Mathematics, SRM Institute of Science and Technology, Kattankulathur 603203, India
[b] School of Computer Engineering, KIIT University, Bhubaneswar, Odisha 751024, India

ARTICLE INFO

Keywords:
Portfolio optimization
Value-at-risk
Machine learning
Regression technique
Stock prediction

ABSTRACT

The future performance of stock markets is the most crucial factor in portfolio creation. As machine learning technique is advancing, new possibilities have opened up for incorporating prediction concepts into portfolio selection. A hybrid approach that constitutes machine learning algorithms for stock return prediction and a mean–VaR (value-at-risk) model for portfolio selection is illustrated in this paper as a unique portfolio construction technique. Machine learning regression models such as Random Forest, Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), Support Vector Machine Regression (SVR), k-Nearest Neighbors (KNN), and Artificial Neural Network (ANN) are adopted to forecast stock values for the next period. The stocks with greater prospective returns are chosen in the first stage of this affection. Further, the mean–VaR portfolio optimization model is employed for portfolio selection in the second stage. The monthly datasets of the Bombay Stock Exchange (BSE), India, Tokyo Stock Exchange, Japan, and Shanghai Stock Exchange, China, are used as the research sample, and the findings show that the mean–VaR model with AdaBoost prediction outperforms other models.

1. Introduction

Optimizing a portfolio is the efficient allocation of wealth amongst different assets, in which two factors are crucial, namely average return and risk. Investors typically aim to maximize profit and reduce risk. The selection of portfolios as a research area started with the Markowitz model (Markowitz, 1952), where the return as mean and risk as variance are assessed. Since then, variance has been a popular risk indicator, and the mean–variance model has gotten a lot of attention among researchers such as Björk et al. (2014), Liagkouras and Metaxiotis (2018), Rosadi et al. (2020), Çela et al. (2021) and Katsikis et al. (2021). While the quadratic programming problem based mean–variance model is extensively researched, Konno and Yamazaki (1991) introduced a linear portfolio selection approach by defining investment risk as absolute deviation, which is also widely popular among researchers (Kenyon et al., 1999; Kamil and Ibrahim, 2007; Moon and Yao, 2011; Qin et al., 2011). The mean–variance and mean–absolute deviation models have been shown to be effective in many stock markets when portfolio returns are symmetrical. However, if portfolio returns are asymmetrical, they are insufficient because variance and absolute deviation regard high returns, which investors desire, as bad as low returns, which investors do not want. To address the issue, many downside risks (Expected shortfall, VaR, CVaR (conditional value-at-risk) have been proposed those solely evaluate the negative deviations from the expected return. However, compared to CVaR, expected shortfall, and Sharpe ratio, non-parametric VaR has superior robustness, according to Cont et al. (2010). Kou et al. (2013) further stated that utilizing CVaR as a risk metric is not robust due to the models and data used. They also demonstrated that VaR is a better risk indicator for calculating trading book capital requirements. So far, many researchers have investigated VaR risk measure as an objective function in the portfolio selection problem and its extensions (Ahmadi-Javid and Fallah-Tafti, 2019; Wang et al., 2020; Liu et al., 2021; Kumar et al., 2022).

Nonetheless, the above studies focus on improving and expanding the VaR model while overlooking the need for high-quality asset selection before constructing an optimum portfolio. It should be noted that different assets might have different characteristics and that not all assets in the market are suitable for every investor. Therefore, to construct an optimal portfolio, one must choose certain high-quality assets depending on investor preferences. In this framework, the improvement and optimization of the portfolio are now one of the most critical aspects of current financial research and investment policy. Since the effectiveness of portfolio selection is highly dependent on future stock market performance, according to Yang et al. (2019), reasonable and

[*] Corresponding author.
E-mail addresses: jyotirmayeebehera452@gmail.com (J. Behera), akpasayat@gmail.com (A.K. Pasayat), hkb.math@gmail.com (H. Behera), maths.pk@gmail.com (P. Kumar).

https://doi.org/10.1016/j.engappai.2023.105843
Received 10 August 2022; Received in revised form 4 December 2022; Accepted 8 January 2023
Available online 24 January 2023

accurate predictions are likely to lead to good returns on investment and also mitigate risk.

However, the forecast of the stock market is a difficult prediction, as the stock market is fundamentally a non-linear, dynamic, noisy, and chaotic system. Moreover, stock prices are impacted by numerous variables, including political events, the policies and announcements of companies, economical conditions, interest rates, and the feelings of investors (Wang et al., 2011). So far, for the prediction process, the majority of previous research has centered on statistical methods and machine learning methods. Statistical approaches attempt to forecast the future by examining previous pricing characteristics like auto-regressive integrated moving average (ARIMA) (Box et al., 2015; Katris, 2021), auto-regressive conditional heteroscedasticity (ARCH) (Engle, 1982; Shephard, 2020), generalized auto-regressive conditional heteroscedasticity (GARCH) (Bollerslev, 1986; Garcia et al., 2005) while common methods in machine learning are support vector machine (Villegas et al., 2018; Sartakhti et al., 2019), random forest (Uimonen et al., 2020; Kavzoglu and Teke, 2022), neural networks (Liu and Yeh, 2017; Korkmaz and Acikgoz, 2022), etc. Despite the fact that statistical methods can help in predicting stock prices, multiple comparison studies (Song et al., 2004; Boulesteix and Schmid, 2014; Zhang et al., 2018) have found that machine learning is better at dealing with non-stationary and non-linear difficulties than statistical models.

Recently, certain research (Freitas et al., 2009; Day and Lin, 2019; Jiang et al., 2020; Kaczmarek and Perez, 2021; Padhi et al., 2022) have been performed by incorporating the pre-selection of assets into portfolio selection models using machine learning techniques. For instance, a generic framework for integrating stock predictions in portfolio optimization models is provided by Ustun and Kasimbeyli (2012), in which they created a mean–variance-skewness portfolio optimization model with eleven goal functions. Deng and Min (2013) used a linear regression model with ten factors for selecting stocks from US and worldwide markets and put up a mean variance-based portfolio with practical risk tolerance, error monitoring, and turnover limitations. They observed that the risk-adjusted return on the global equities model provided was superior compared to the domestic equity universe, and the return from the portfolio rose with systemic monitoring errors and risk tolerance. Moreover, Gupta et al. (2012) presented a hybrid method to assist investors in making investment decisions. They took into account three asset classes: risk, return, and liquidity, and used a support vector machine kernel-based radial function to classify a chosen sample of financial assets into those three classes for portfolio optimization. Ma et al. (2021) used two machine learning models, random forest (RF) and support vector regression, as well as three deep learning models, convolutional neural network, long short-term memory (LSTM) neural network, and deep multi-layer perceptron (DMLP) to predict the return in portfolio construction. In the next step, they used the predicted result to advance the omega portfolio optimization and mean–variance (MV) models. As a result, it is found that the prediction of return in MV and omega models with the RF algorithm outperforms the other models.

The above-mentioned studies revealed that combining stock prediction with portfolio selection might offer a fresh viewpoint on financial analysis. One can also note from the aforementioned pieces of literature that statistical techniques have been used in specific manners by own consideration for the prediction of stocks' value, which involve some computational drawbacks. Furthermore, there are no studies on hybrid techniques that combine machine learning regression algorithms for stock prediction with the mean–VaR portfolio optimization for efficient investment strategies. The present study, therefore, aimed to overcome the mentioned issues by examining the suitability of different forecasting methods for the world stock market and finding a single procedure that can create an efficient investment strategy for all markets. Also, it can help investors save time and resources with better decisions. This study is helpful for investors who wish to minimize risk and diversify their portfolios, aiming to maximize profit. Based on all these facts, the objectives of this work are depicted as follows:

- A portfolio optimization algorithm is designed by integrating the machine learning regression models for stock prediction with the mean–VaR model for portfolio selection.
- To examine the performance of six machine learning models, RF, AdaBoost, XGBoost, SVR, KNN, and ANN, in the stock preselection process before portfolio creation. The results of these models ensure that only high-quality stocks are selected before the portfolio optimization model is built.
- The mean–VaR model is incorporated over the predicted set of stocks to obtain a more appropriate and efficient investment strategy for the investor with the market condition.
- For the efficiency and applicability of the suggested methodology, datasets are taken from the Bombay Stock Exchange, Tokyo Stock Exchange, and Shanghai Stock Exchange.

The remaining paper proceeds as follows: Section 2 describes the forecasting methods used. The mean–VaR model is provided in Section 3. Section 4 presents the details of the experiment performed in the paper. The results of the empirical studies are discussed in Section 5. Section 6 concludes with a summary of our major findings and a plan for the future.

2. Prediction models

2.1. Random forest

Random forest is a widely used machine learning technique introduced by Ho (1995), aggregating several decision trees' outputs to produce a single solution. Every tree in the ensemble is constructed using a bootstrap sample, a random sample produced with replacement from a training set. One-third of the training sample is taken out as test data; this portion of the sample is known as the out-of-bag (OOB) sample (Chelgani et al., 2021). Additional feature bagging is then used to introduce randomization into the dataset, boosting its diversity and lowering decision tree correlation. The determination of the prediction will vary according to the kind of problem. In the case of classification, the decision trees in the RF-induced prediction model are arranged in a group, and the outcome is determined by the predictions made by each tree, such as by using a majority vote (Tang et al., 2021). Meanwhile, the average of the trees' predictions makes up the final prediction in regression. That is, when RF gets an input vector (x) containing the values of the many evident features that have been examined for a certain training area, it generates m regression trees and then averages the outcomes. The RF prediction equation after m such trees $\{T(x)\}_1^M$ is

$$f_{rf}^{M}(x) = \frac{1}{M}\sum_{m=1}^{M} T(x). \tag{1}$$

An RF model has many parameters similar to other tree-based algorithms. However, this requires more hyperparameters which are designed to reduce the risk of overfitting and thus increase accuracy. The tuning parameters presented in Table 1 are as follows: the number of decision trees (n-estimator), the maximum depth of the tree (max-depth), the least number of samples needed to split an internal node (min-sample-split), the minimum number of samples required to be at a leaf node (min-sample-leaf), and the number of features to take into account while determining the best split (max-feature). This work sets the optimal value of the hyperparameters after many experiments, which are listed in Table 1.

2.2. AdaBoost

Freund and Schapire (1997) introduced AdaBoost as the first boosting approach. AdaBoost builds a stronger and more stable classifier by using weak classifiers. The first stage is for the base learner or weak learner to give each observation equal attention or weight. Once each

Table 1
Parameters for prediction models.

Model	RF	AdaBoost	XGBoost	SVR	KNN	ANN
Parameter	n_estimator	base_estimator	n_round	C	n_neighbors	number of hidden layer
(Value)	(500)	(100)	(100)	(10)	(3)	(4)
	max-depth	n_estimator	max_depth	gamma		learning_rate
	(20)	(50)	(7)	(0.1)		(0.1)
	min-sample-split	learning_rate	learning_rate			batch_size
	(10)	(1)	(0.01)			(120)
	min-sample-leaf		gamma			epochs
	(10)		(2)			(100)
	max-feature					
	(40)					

observation has been given a weight, the weak learner can be used to make predictions (Shang et al., 2022). The weak learner's misclassified observations are given higher weights, and the next weak learner is utilized to make predictions. This procedure will be repeated until it approaches the T_i, which is the limit of the basic learning algorithm. The outputs of the base learner will be merged in the last stage to create a stronger learner that will improve prediction accuracy.

The AdaBoost method based on Schapire (1999) runs on a training set consisting of vectors (x_1, y_1), (x_2, y_2), …, (x_k, y_k), where x_k is an element from the domain X and y_k is an element from the set Y. The weights of the distribution corresponding to case i, denoted as $D_t(i)$, $i = 1, 2, \ldots ., k$, are determined at each iteration t. The weight is set evenly as $D_1(i) = 1/k$ in the first iteration for $t = 1$. After that, $h_t(x) \rightarrow X(0, 1)$ is obtained as a weak classifier. The weak learner's error $h_t(x)$, is then determined using $\epsilon_t = \sum_{i=1}^{k} D_t(i)\ \delta_t$, where

$$\delta_t = \begin{cases} 0 & \text{if } h_t(x_i) = y_i, \\ 1 & \text{otherwise.} \end{cases}$$

$D_{t+1}(i) = D_t(i)\frac{F_i}{Z_t}$ updates the weight at each iteration, where

$$F_i = \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i, \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i, \end{cases}$$

$$\alpha_t = \frac{1}{2}\ ln\ (\frac{1 - \epsilon_t}{\epsilon_t}),$$

Z_t = normalization factor such as D_t sums one.

The considered hyperparameters for AdaBoost are base-estimator, n-estimator, and learning rate. Moreover, the investigated values for these parameters are listed in Table 1.

2.3. XGBoost

XGBoost is a distributed gradient boosting toolkit that has been tuned for performance (Yan et al., 2022). It uses a recursive binary partitioning strategy to obtain the optimal model by choosing the best partition at each step. XGBoost's tree-based nature makes it insensitive to outliers, and like many boosting methods, it is resistant to overfitting, which makes model selection much easier (Alabdullah et al., 2022; Pasayat et al., 2022). Eq. (2) depicts the XGBoost model's regularized objective at the tth training step, where $l(y_{pred}^{(t)}, y_{truth})$ denotes the loss, which refers to calculating the difference between the simulated value $y_{pred}^{(t)}$ and the associated ground truth y_{truth}.

$$L^{(t)} = \sum_{i} l(y_{pred}^{(t)}, y_{truth}) + \sum_{k} \Omega(f_k), \tag{2}$$

where $\Omega(f_k) = \gamma T + \frac{1}{2}\ \lambda\ \|w\|^2$ represents the complexity of the kth tree, in which T signifies the number of leaves and $\|w\|^2$ specifies the ℓ_2 norm of all leaf scores for training examples. When searching the tree, the parameters γ and λ regulate the degree of conservatism.

The tuned parameters for XGBoost are n-rounds (to determine a maximum number of iterations), max-depth (to control the depth of the tree), learning rate, and gamma (to control regularization to prevent overfitting). After tuning, the values of the hyperparameters are listed in Table 1.

2.4. Support vector machine

Support vector machine (SVM) was first introduced in 1995 by Cortes and Vapnik (1995). It is a supervised machine learning algorithm used in various industries to solve regression and classification problems (Pant and Kumar, 2022). For classification problems, SVM is popular as an approach that creates the largest margin hyperplane to offer the maximum separation between different classes. The instances that are closest to the maximum margin hyperplane for a given learning problem are referred to as support vectors. The distinct function might also be either linear or nonlinear. If the case is linearly separable, a linear hyperplane can be used to divide the instances; if not, the situation is nonlinearly separable (Xia et al., 2022). SVM models have also been expanded to include general estimation and prediction issues. One such extension is SVR introduced by Drucker et al. (1996). SVR uses convex optimization techniques to transform complex nonlinear regression problems into linear regression problems. Through this conversion, an adjustment has been developed between model complexity and empirical error that provides high prediction accuracy. In SVR, the non-linear relationship between random variables X and Y is expressed by the linear function f:

$$f(x) = w\phi(x) + b,$$

where $f(x)$ represents the simulated value, and the parameters $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ can be changed (Chelgani et al., 2021). SVR is formulated based on minimizing structural and empirical risks as presented in Eq. (3) (Zhou et al., 2022).

$$\min\ R(w, \xi, \xi^*) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi^* + \xi) \tag{3}$$

$$\text{subject to:}\ \ y_i - w\phi(x_i) - b \leq \epsilon + \xi, \quad i = 1, 2, \ldots, n, \tag{4}$$

$$-\ y_i + w\phi(x_i) + b \leq \epsilon + \xi^*, \quad i = 1, 2, \ldots, n, \tag{5}$$

$$\xi_i^* \geq 0, \quad i = 1, 2, \ldots, n, \tag{6}$$

$$\xi_i \geq 0, \quad i = 1, 2, \ldots, n. \tag{7}$$

This work uses the radial basis function as the kernel function, which is given as follows

$$k(x_i, x_j) = exp(-\gamma\|x_i - x_j\|^2).$$

The considered SVR parameters for tuning are the regularization parameter C and γ. The optimal values of these parameters are listed in Table 1.

2.5. k-Nearest neighbors

The k-nearest neighbors algorithm, also referred to as KNN or k-NN, is a non-parametric, supervised learning technique that is used to solve both classification and regression problems. During determining a class label for a classification problem, "the majority vote" is used, i.e., the label associated with the data point that is used most often. The term "majority vote" is more frequently used in literature even though this is technically referred to as "plurality voting" (Gawri et al., 2022). Furthermore, KNN regression is based on locating a group of points within a dataset that is closest to the "query" point as measured by some distance metric (such as Euclidean, Minkowski, or Manhattan) (Lin et al., 2022). For a specific value of K,

$$f(x) = \frac{1}{K} \sum_{x_i \in N(x)} y_i. \tag{8}$$

The best K value depends on the trade-off between bias and variance, with a smaller K value allowing lower bias and higher variance and a larger K value allowing higher bias and lower variance. A hierarchical tree-based data structure is created using the KNN method to organize the dataset (Qiu et al., 2021; Mishra et al., 2022a). KNN regression does not require fitting any models as it is memory-based (Aslan et al., 2022). If $p = (p_1, p_2, \dots .p_n)$ and $q = (q_1, q_2, q_3 \dots \dots q_n)$ are two vectors, then the distance metrics are determined as follows

$$\text{Euclidean distance} = \left(\sum_{i=1}^{n} |p_i - q_i|^2 \right)^{\frac{1}{2}}, \tag{9}$$

$$\text{Minkowski distance} = \left(\sum_{i=1}^{n} |p_i - q_i|^\alpha \right)^{\frac{1}{\alpha}}, \tag{10}$$

$$\text{Manhattan distance} = \sum_{i=1}^{n} |p_i - q_i|. \tag{11}$$

The main parameter for KNN is n-neighbors (number of neighbors), and the optimal value for this parameter is presented in Table 1.

2.6. Artificial neural network

ANN is a fusion of various machine learning concepts, including regression, ensemble, perceptrons, and gradient descent. It takes a linear combination of the input variables and transforms them into non-linear functions of the dependent variable's derived characteristics (Mishra et al., 2022b). They are the most basic type of deep networks, made up of numerous layers of hidden neurons that are each completely connected to both the layer above (from which they influence) and below (from which they receive input) (Sahoo et al., 2022). By taking into consideration networks with increasing layers of interconnected units, any general functions can be built. The weighted linear combination of inputs is processed by each layer and transformed using an activation function. The output from jth unit when w^1_{ji} is the weight that connects the jth unit from ith unit in layer 1 is

$$Z_j = g\left(\sum_{i=0}^{d} w^1_{ji} x_i \right). \tag{12}$$

Among several activation function options, Rectified Linear Unit (ReLU) is used in the current work. ANNs are capable of learning the feature on their own, and they are trained using the backpropagation approach. The backpropagation approach is a highly well-liked neural network learning algorithm despite the fact that the solution is not a global one (Qiu et al., 2021). However, ANN has some drawbacks. ANN requires a lot of computational power. Also, neural network models are hard to explain. Neural network training requires lots of data. Moreover, data preparation for neural network models needs careful attention.

The number of hidden layers, learning rate, batch size, and epochs are considered hyperparameters for ANN. Moreover, the optimal hyperparameters obtained after tuning are presented in Table 1.

3. Mean-VaR portfolio optimization model

Suppose there are $n \geq 2$ number of risky assets that the investor decides to invest for a fixed time period T. Suppose r_i represents the asset's return rate i. Then the expected rate of return of the asset i is given by $\mu_i = \frac{1}{T} \sum_{t=1}^{T} r_{it}$. Let x_i be the proportion of the wealth invested in asset i in such a way that $\sum_{i=1}^{n} x_i = 1$. Then the rate of return of the portfolio is $r_p = \sum_{i=1}^{n} r_i x_i$, the expected return of the portfolio is $\mu_p = \sum_{i=1}^{n} \mu_i x_i$, and the variance of the portfolio is $\sigma_p^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$. Further, the definition for VaR of the portfolio is provided as follows:

Definition 3.1 (Alexander and Baptista, 2002, 2004)**.** Let α^* be the α-quantile of standard normal distribution, $\alpha \in (0.5, 1]$. Then, for a fixed period of time, the portfolio's VaR may therefore be expressed as follows:

$$\begin{aligned} VaR &= \alpha^* \sigma_p - \mu_p \\ &= \alpha^* \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j} - \sum_{i=1}^{n} \mu_i x_i. \end{aligned}$$

This paper has considered the mean–VaR portfolio optimization model, which is adapted from (Sheng et al., 2012). Following is the mean–VaR portfolio optimization problem.

$$\min \ VaR = \alpha^* \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j} - \sum_{i=1}^{n} \mu_i x_i \tag{13}$$

$$\text{subject to:} \quad \sum_{i=1}^{n} \mu_i x_i \ = \mu_{fix}, \tag{14}$$

$$\sum_{i=1}^{n} x_i = 1, 0 \leq x_i < 1, \forall \, i = 1, 2, \dots, n. \tag{15}$$

This mean–VaR model is constructed by allying VaR as a risk measure to the mean–variance portfolio optimization model. In the model, the objective is to minimize the VaR of the portfolio. Meanwhile, as presented in constraints (14) and (15), the investor wants that the expected return of the portfolio must achieve the preset minimum expected return value μ_{fix}, and the investment proportion (x_i) must sum to one.

4. Experimental process

4.1. Data

The stock return predictability is related to volatility over time. Stocks that are reasonably steady are naturally easier to predict than relatively noisy ones. This research uses monthly historical data of randomly selected stocks from the BSE as an experimental dataset to test the proposed methodology. Experimental data contains 50 randomly selected stocks ranging from January 2015 to December 2021, which are listed in Table 2. After data collection, data preprocessing is done. In this part, data cleaning is considered for handling missing values. Also, data integration, attribute selection, and creation of new variables are performed in preprocessing part for better prediction of the stock. Moreover, the capping method is used to remove the outliers from the dataset. For each stock, monthly returns are used as an input feature. The dataset is divided into a training set and a testing set with a ratio of 7:3. The model is trained using the training set, whereas the testing dataset is used to evaluate the performance of the applied methods. In the experiment, RF, AdaBoost, XGBoost, SVR, and KNN are prepared based on the Scikit-learn machine learning package, whereas ANN is prepared based on Keras deep learning package using Python software. Moreover, the visual representation of the monthly rate of return of the stocks is provided in Fig. 1.

4.2. Methodology

Individual investors in the financial market typically want to know which strategy should be implemented to assist them in assembling the optimal investment portfolio, one that offers the most significant return potential with the least amount of risk. As a result, the primary emphasis of this paper is placed on picking stocks that have high returns in order to construct a portfolio from the prediction viewpoint. This strategy has the following two stages:

- **Stock return prediction:** For stock return prediction, machine learning regression techniques like RF, AdaBoost, XGBoost, SVR, KNN, and ANN are taken into account. To check the accuracy of the models, three metrics, namely mean absolute error, mean

Table 2
List of stock number and stock code.

Stock No.	S-1	S-2	S-3	S-4	S-5	S-6	S-7	S-8	S-9	S-10
Code	TATASTEEL	RELIANCE	KOTAKBANK	ASHOKLEY	ADANIENT	TCS	HDFCBANK	SBIN	APOLLOHOSP	TATAPOWER
Stock No.	S-11	S-12	S-13	S-14	S-15	S-16	S-17	S-18	S-19	S-20
Code	ICICIBANK	INFY	ONGC	TATAMOTORS	MARUTI	ADANIPORTS	JSWSTEEL	AXISBANK	HCLTECH	INDUSINDBK
Stock No.	S-21	S-22	S-23	S-24	S-25	S-26	S-27	S-28	S-29	S-30
Code	IGL	BHARTIARTL	LT	ITC	BAJFINANCE	HINDALCO	DRREDDY	HINDUNILVR	HDFC	BAJAJFINSV
Stock No.	S-31	S-32	S-33	S-34	S-35	S-36	S-37	S-38	S-39	S-40
Code	DLF	AUROPHARMA	VEDL	BPCL	WIPRO	ASIANPAINT	PEL	EICHERMOT	POWERGRID	M&M
Stock No.	S-41	S-42	S-43	S-44	S-45	S-46	S-47	S-48	S-49	S-50
Code	GAIL	CIPLA	ZEEL	COALINDIA	HEROMOTOCO	NTPC	TECHM	TITAN	UPL	SUNPHARMA

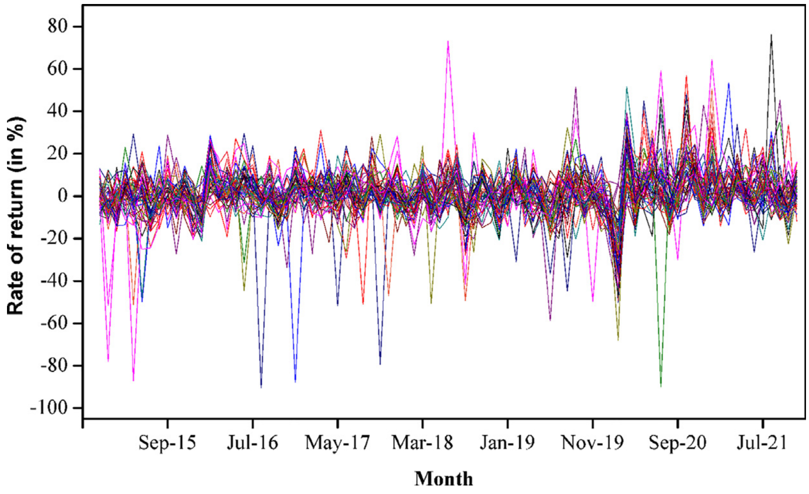


Fig. 1. Monthly rate of return of the stocks.

squared error, and root-mean-squared error, are used in this work. Stocks with better performance obtained from different models are taken into consideration for the next stage. For a better explanation, the pseudocode for the stock return prediction process is presented in Algorithm 1.

- **Portfolio optimization:** The aim of this stage is to determine the proportion of the wealth allocated to each stock. This work uses the mean–VaR model with selected top-performing stocks from different models to conduct the portfolio's asset allocation.

The detailed procedure of the proposed approach is illustrated in Fig. 2.

Algorithm 1: Pseudocode for stock prediction

```
Input: Determine the hyperparameters and training & testing dataset
Output: Determine the RMSE, MSE, and MAE
Select the optimal hyperparameters for all the regression models.
while (Stopping criteria is not met) do
    Implement regression models to train each data point
    Implement models for testing of the data points
Return: RMSE, MSE, and MAE
```

5. Result analysis

5.1. Result for stock return prediction

In this study, three metrics, namely mean absolute error (MAE), mean squared error (MSE), and root mean square error (RMSE), are used to measure the efficiency of different models in the stock return prediction process.

Table 3
Prediction performance of different models in different training–testing ratios.

Model	Training–testing ratio	MAE	MSE	RMSE
RF	60:40	0.095025	0.018299	0.128680
	70:30	0.089915	0.016306	0.120430
	80:20	0.092755	0.017117	0.122681
	90:10	0.091220	0.018971	0.122111
AdaBoost	60:40	0.090904	0.019318	0.131720
	70:30	0.082587	0.015485	0.116202
	80:20	0.089631	0.017629	0.125166
	90:10	0.093105	0.021685	0.131764
XGBoost	60:40	0.109125	0.024783	0.149152
	70:30	0.103672	0.021403	0.138668
	80:20	0.107367	0.021377	0.139910
	90:10	0.110329	0.025700	0.145028
SVR	60:40	0.097336	0.019064	0.132171
	70:30	0.089996	0.016969	0.129265
	80:20	0.093154	0.018524	0.134103
	90:10	0.099596	0.019262	0.136788
KNN	60:40	0.108434	0.024798	0.150407
	70:30	0.103673	0.022034	0.140439
	80:20	0.107105	0.022346	0.144677
	90:10	0.010984	0.025887	0.158885
ANN	60:40	0.173421	0.028793	0.163685
	70:30	0.163581	0.025015	0.158161
	80:20	0.189253	0.028876	0.162929
	90:10	0.203520	0.031501	0.171485

- **Mean Absolute Error (MAE)**: (Elavarasan et al., 2018) The mean absolute error is estimated by averaging the absolute difference between predicted and actual values over the whole dataset. To put it mathematically, it is the arithmetic average of absolute error. MAE simply measures the magnitude of the errors and does not care about their direction. The lower the MAE, the more

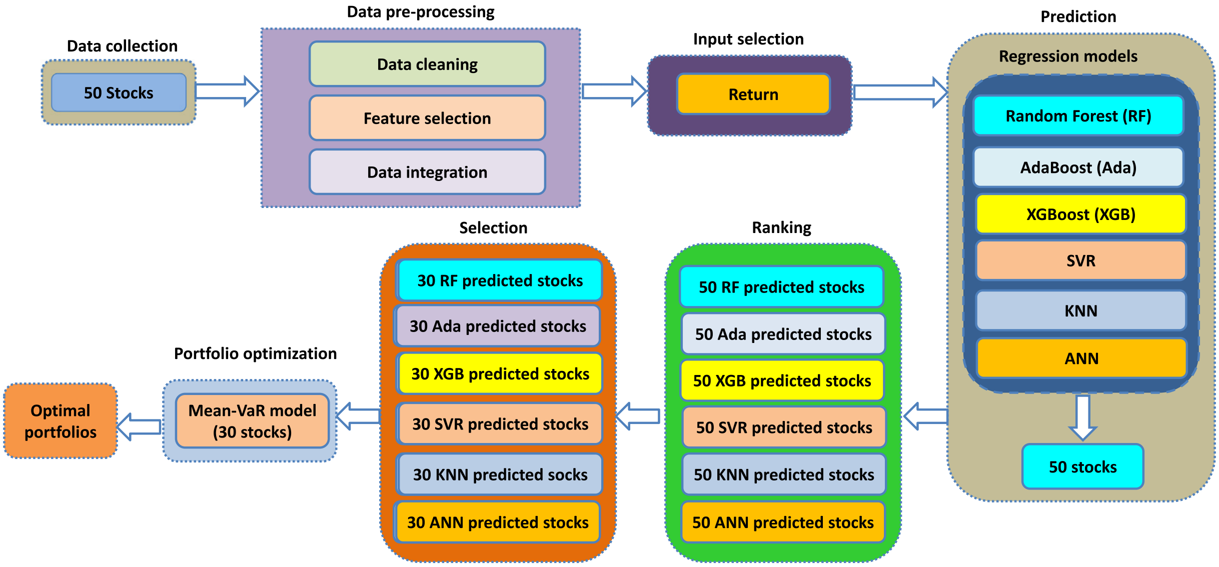


Fig. 2. The scheme of regression approach to portfolio optimization.

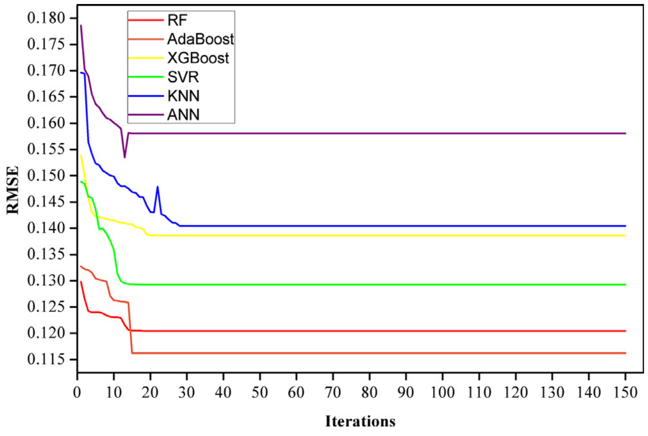


Fig. 3. RMSE convergence graph for different prediction models.

Table 4
Prediction performance of different models.

Model		MAE	MSE	RMSE
RF	Mean	0.089915	0.016306	0.120430
	σ	0.025573	0.013410	0.042883
	σ^2	0.000654	0.000180	0.001839
AdaBoost	Mean	0.082587	0.015485	0.116202
	σ	0.027270	0.012641	0.044976
	σ^2	0.000744	0.000160	0.002023
XGBoost	Mean	0.103672	0.021403	0.138668
	σ	0.030260	0.015379	0.047105
	σ^2	0.000916	0.000237	0.002219
SVR	Mean	0.089996	0.016969	0.129265
	σ	0.026260	0.013704	0.045771
	σ^2	0.000690	0.000188	0.002095
KNN	Mean	0.103673	0.022034	0.140439
	σ	0.031102	0.016044	0.047115
	σ^2	0.000967	0.000257	0.002220
ANN	Mean	0.163581	0.025015	0.158161
	σ	0.038324	0.019930	0.050090
	σ^2	0.001469	0.000397	0.002509

accurate a model is. The formula in the below equation is used to compute the average of the dataset's residuals.

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j|, \tag{16}$$

where y_j = actual values, $\hat{y}_j$ = predicted values.

- **Mean Squared Error (MSE)**: (Behera et al., 2022) The mean squared error (MSE) is a measurement of how close a regression line is to a group of points. This is determined by taking the square of distances between the regression line and the points. To eliminate any negative signs, squaring is essential. Since the average of a set of errors is determined by this, it is termed a mean squared error. The lower MSE, the more accurate prediction. Following is the formula to calculate MSE.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2. \tag{17}$$

- **Root mean squared error (RMSE)**: (Ho et al., 2021) The standard deviation of the prediction errors (residuals) is known as the root mean squared error (RMSE). It provides the measurement of how dispersed the residuals are, where the residual measures how far the data points deviate from the regression line. In other words, it informs the concentration of data around the best fit line. In climatology, regression analysis, and forecasting, RMSE is frequently used to verify the outcomes of experiments. It can be defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}. \tag{18}$$

The performances of the machine learning regression models at different training–testing ratios are compared in Table 3. According to the findings of the comparison, the models are showing better prediction at the 70:30 training–testing ratio. Meanwhile, the 90:10 training–testing ratio shows less prediction accuracy for all six models. Moreover, the metrics' mean, standard deviation, and variance for different models are provided in Table 4. As per the findings, AdaBoost possesses the lowest values for MAE, MSE, and RMSE when compared to the other five models. In addition to this, the variance and standard deviation of the MSE and RMSE are both lower when using AdaBoost. In the meantime, RF has the smallest standard deviation as well as the

Table 5
Top 30 stocks predicted from each regression model.

RF predicted stocks	Ada predicted stocks	XGB predicted stocks	SVR predicted stocks	KNN predicted stocks	ANN predicted stocks
HINDUNILVR	HCLTECH	POWERGRID	HCLTECH	POWERGRID	TATAPOWER
DRREDDY	HINDUNILVR	NTPC	HINDUNILVR	HDFC	UPL
BHARTIARTL	ASIANPAINT	HINDUNILVR	HDFC	NTPC	CIPLA
HEROMOTOCO	TITAN	ASIANPAINT	POWERGRID	HINDUNILVR	ADANIENT
ASIANPAINT	HDFC	INFY	CIPLA	ASIANPAINT	HINDALCO
HDFC	KOTAKBANK	HDFCBANK	ITC	ITC	ONGC
POWERGRID	ITC	HEROMOTOCO	UPL	COALINDIA	HINDUNILVR
NTPC	AXISBANK	ICICIBANK	NTPC	AXISBANK	DLF
ICICIBANK	IGL	ITC	TITAN	DRREDDY	ICICIBANK
HCLTECH	NTPC	HDFC	SUNPHARMA	TCS	ASHOKLEY
UPL	RELIANCE	TCS	DRREDDY	GAIL	NTPC
ASHOKLEY	WIPRO	COALINDIA	GAIL	EICHERMOT	ZEEL
ITC	HEROMOTOCO	WIPRO	BHARTIARTL	HDFCBANK	COALINDIA
BPCL	SUNPHARMA	SUNPHARMA	COALINDIA	ICICIBANK	SBIN
IGL	POWERGRID	DRREDDY	BAJFINANCE	HEROMOTOCO	POWERGRID
APOLLOHOSP	ONGC	AXISBANK	TECHM	TECHM	TATAMOTORS
TCS	COALINDIA	SBIN	KOTAKBANK	ZEEL	ADANIPORTS
AUROPHARMA	LT	ASHOKLEY	AUROPHARMA	UPL	TATASTEEL
LT	DRREDDY	ADANIPORTS	ASIANPAINT	IGL	GAIL
SBIN	ASHOKLEY	IGL	HDFCBANK	TITAN	ASIANPAINT
CIPLA	MARUTI	RELIANCE	AXISBANK	AUROPHARMA	ITC
MARUTI	BHARTIARTL	ONGC	MARUTI	SBIN	AUROPHARMA
GAIL	UPL	UPL	APOLLOHOSP	MARUTI	WIPRO
TITAN	TATAPOWER	MARUTI	INFY	RELIANCE	TCS
TATAPOWER	APOLLOHOSP	CIPLA	ADANIPORTS	BHARTIARTL	VEDL
HDFCBANK	TCS	BHARTIARTL	WIPRO	LT	LT
AXISBANK	GAIL	TECHM	PEL	SUNPHARMA	SUNPHARMA
SUNPHARMA	TATASTEEL	ZEEL	DLF	M&M	RELIANCE
ADANIPORTS	CIPLA	TITAN	ICICIBANK	ASHOKLEY	AXISBANK
INDUSINDBK	ICICIBANK	KOTAKBANK	JSWSTEEL	ADANIPORTS	BHARTIARTL

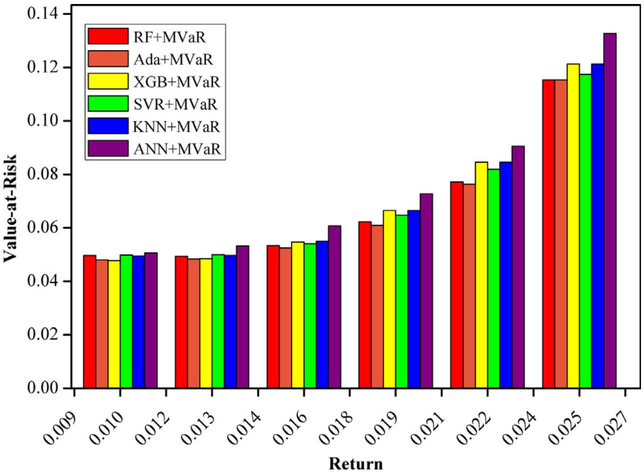


Fig. 4. Return-VaR barplot for different models.

smallest variance for the MAE. In general, this section demonstrates that AdaBoost performs better than other models when it comes to the process of predicting stock returns. In addition, following the prediction process, only 30 out of the 50 stocks are taken into consideration for the subsequent stage. This selection is based on the performance of the stock. Stocks with higher predicted errors have been eliminated from the list. The 30 stocks that are chosen from the six different models are presented in Table 5. Moreover, in Fig. 3, RMSE convergence graphs for different prediction models are illustrated. It can be noticed that convergence of the RMSE curve is achieved for each model. It also states that the overfitting problem does not happen in our study.

5.2. Result for portfolio optimization

The outcomes of various mean-Value-at-Risk (MVaR) models' experimental results are presented in this section. Additionally, this study utilizes RF+MVaR, Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR to depict MVaR models with RF predicted stock, AdaBoost predicted stock, XGBoost predicted stock, SVR predicted stocks, KNN predicted stocks, and ANN predicted stocks respectively. To obtain the optimal investment proportion, the above nonlinear programming problem-based MVaR model (13)–(15) has been evaluated based on successive linear programming (SLP) algorithm with the help of the LINGO 11 optimization solver. LINGO is a well-known comprehensive optimization tool that is designed to solve linear, nonlinear (convex or nonconvex), and integer optimization models.

This paper begins the result analysis by obtaining the efficient portfolios for all of the values of $\mu_{fix} = 0.010, 0.013, 0.016, 0.019, 0.022, 0.025$ when α^* is equal to 1.75 for the models RF+MVaR, Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR. In addition, VaR values are calculated for each portfolio for the RF+MVaR, Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR models. Tables 6, 7, 8, 9, 10, and 11 present the comprehensive list of efficient portfolios that can be generated from different models at all values of μ_{fix}. In Table 6, the efficient portfolios obtained by varying μ_{fix} at $\alpha^* = 1.75$ for the model RF+MVaR are depicted. Table 6 also provides the details of the portfolio's VaR, standard deviation, and expected return values for a fixed $\alpha^* = 1.75$ at different μ_{fix} values. From the table, it can be observed that the value of VaR increases as the μ_{fix} value goes up. Also, one can see that different groups of stocks are obtained with different rates of proportion by varying the μ_{fix} value. As the value of μ_{fix} increases, the number of stocks selected for the efficient portfolio decreases due to an increase in VaR. Similarly, in Tables 7, 8, 9, 10, and 11, the same type of results for the Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR models, respectively, are illustrated. Overall, from Tables 6, 7, 8, 9, 10, and 11, it can be seen that with the increase of μ_{fix}, VaR and standard deviation are also increasing in all six models. However, Ada+MVaR shows a little lower VaR values among the six models at all μ_{fix} values. Thus, it concludes that Ada+MVaR outperforms other models by minimizing VaR. Furthermore, for better visualization of the obtained VaR value from the models RF+MVaR, Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR, a comparison barplot is provided in Fig. 4.

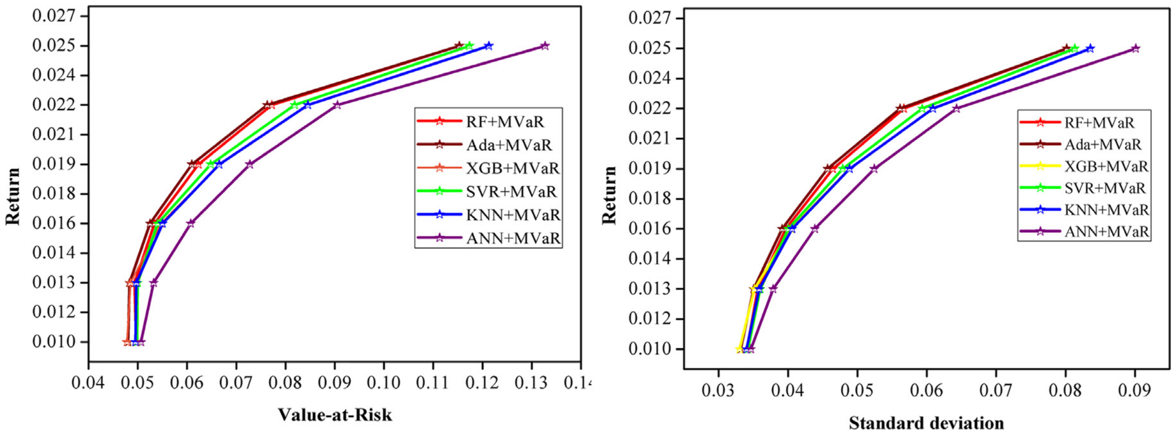


Fig. 5. Comparison risk-return efficient frontiers for different models.

Table 6
Proportion of investment on stocks obtained from RF+MVaR model for different μ_{fix} values.

μ_{fix}	**0.01**	**0.013**	**0.016**	**0.019**	**0.022**	**0.025**
VaR	**0.049650**	**0.049358**	**0.053393**	**0.062310**	**0.077214**	**0.115289**
σ	**0.034086**	**0.035633**	**0.039653**	**0.046463**	**0.056694**	**0.080165**
S-28	0.309726	0.291548	0.256569	0.163270	0	0
S-27	0.158674	0.148590	0.122039	0.058012	0	0
S-22	0.037793	0.036815	0.023003	0	0	0
S-45	0	0	0	0	0	0
S-36	0.078861	0.129516	0.203266	0.291489	0.352549	0
S-29	0	0	0	0	0	0
S-39	0.135023	0.113568	0.043686	0	0	0
S-46	0.070743	0.007166	0	0	0	0
S-11	0	0	0	0	0	0
S-19	0	0	0	0	0	0
S-49	0	0	0	0	0	0
S-4	0.025964	0.036816	0.037279	0.018837	0	0
S-24	0.017166	0	0	0	0	0
S-34	0	0	0	0	0	0
S-21	0.000824	0.017065	0.039257	0.059636	0.038223	0
S-9	0.000387	0.030412	0.081973	0.142663	0.196574	0.164661
S-6	0.006430	0	0	0	0	0
S-32	0	0	0	0	0	0
S-23	0	0	0	0	0	0
S-8	0	0	0	0	0	0
S-42	0.015828	0.031184	0.007883	0	0	0
S-15	0	0	0	0	0	0
S-41	0	0	0	0	0	0
S-48	0.033574	0.088323	0.15936	0.263349	0.412654	0.835339
S-10	0.003528	0.021427	0.025686	0.002744	0	0
S-7	0.052801	0.033105	0	0	0	0
S-18	0	0	0	0	0	0
S-50	0.052678	0.014464	0	0	0	0
S-16	0	0	0	0	0	0
S-26	0	0	0	0	0	0

Table 7
Proportion of investment on stocks obtained from Ada+MVaR model for different μ_{fix} values.

μ	**0.01**	**0.013**	**0.016**	**0.019**	**0.022**	**0.025**
VaR	**0.048043**	**0.048340**	**0.052495**	**0.060964**	**0.076284**	**0.115289**
σ	**0.033168**	**0.035051**	**0.039140**	**0.045694**	**0.056162**	**0.080165**
S-19	0	0	0	0	0	0
S-28	0.280005	0.262240	0.240329	0.157342	0	0
S-36	0.070143	0.124078	0.196122	0.285372	0.335843	0
S-48	0.038564	0.096237	0.153811	0.231411	0.388029	0.835339
S-29	0	0	0	0	0	0
S-3	0.032506	0.011180	0	0	0	0
S-24	0.003096	0	0	0	0	0
S-18	0	0	0	0	0	0
S-21	0.007233	0.020858	0.035018	0.045365	0.013189	0
S-46	0.050484	0	0	0	0	0
S-2	0	0	0.001426	0	0	0
S-35	0.085109	0.065607	0.032825	0	0	0
S-45	0	0	0	0	0	0
S-50	0.060025	0.015586	0	0	0	0
S-39	0.152496	0.127384	0.049926	0	0	0
S-13	0	0	0	0	0	0
S-44	0.006233	0	0	0	0	0
S-23	0	0	0	0	0	0
S-27	0.142610	0.142269	0.122180	0.060518	0	0
S-4	0.024741	0.030796	0.019551	0	0	0
S-15	0	0	0	0	0	0
S-22	0.039448	0.038081	0.010961	0	0	0
S-49	0	0	0	0	0	0
S-10	0	0.007930	0.002631	0	0	0
S-9	0	0.027459	0.080993	0.141439	0.194249	0.164661
S-6	0	0	0	0	0	0
S-41	0	0	0	0	0	0
S-1	0	0.016246	0.054226	0.078553	0.068690	0
S-42	0.007308	0.014049	0	0	0	0
S-11	0	0	0	0	0	0

For further comparison between the performance of the models, in Fig. 5, the risk-return efficient frontiers are illustrated. As can be seen, Fig. 5 is comprised of two different sub-graphs. To be more specific, the y-axes of two different subgraphs reflect the portfolio expected return of the models RF+MVaR, Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR. On the other hand, the x-axis depicts the obtained VaR as well as the standard deviation of the portfolio, respectively. The first sub-graph of Fig. 5 depicts the relationship between VaR and the portfolio's average return for the models RF+MVaR, Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR. It is clear from the sub-graph that as the value of VaR increases, the value of the return of the portfolio for different models also increases simultaneously. The return of the model Ada+MVaR increases more as compared to the other five models. Also, the portfolio return of the model RF+MVaR is very close to being the same as the portfolio return of the model Ada+MVaR, but the model ANN+MVaR possesses a lower portfolio return in comparison to the other five models. The second sub-graph of Fig. 5 shows the relationship between standard deviation and expected return of the portfolio for the models RF+MVaR, Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR. As can be observed, the return of the model Ada+MVaR increases more as the value of standard deviation rises in comparison to the other models.

Table 8
Proportion of investment on stocks obtained from XGB+MVaR model for different μ_{fix} values.

μ	**0.01**	**0.013**	**0.016**	**0.019**	**0.022**	**0.025**
VaR	**0.047791**	**0.048539**	**0.054732**	**0.066527**	**0.084558**	**0.121283**
σ	**0.033024**	**0.035165**	**0.040419**	**0.048873**	**0.060890**	**0.083590**
S-39	0.139170	0.117433	0.032980	0	0	0
S-48	0.047738	0	0	0	0	0
S-28	0.277580	0.246909	0.208287	0.105396	0	0
S-36	0.065214	0.129730	0.217838	0.324682	0.402785	0.086606
S-12	0	0	0	0	0	0
S-7	0.045414	0.028542	0	0	0	0
S-45	0	0	0	0	0	0
S-11	0	0	0	0	0	0
S-24	0	0	0	0	0	0
S-29	0	0	0	0	0	0
S-6	0	0	0	0	0	0
S-44	0.006821	0	0	0	0	0
S-35	0.084113	0.067607	0.036214	0	0	0
S-50	0.053252	0.011942	0	0	0	0
S-27	0.152908	0.158760	0.141638	0.099000	0	0
S-18	0	0	0	0	0	0
S-8	0	0	0	0	0	0
S-4	0.017172	0.033235	0.040441	0.021445	0	0
S-16	0	0	0	0	0	0
S-21	0.007133	0.025402	0.046208	0.067405	0.064175	0
S-2	0	0	0.016043	0.027631	0.004479	0
S-13	0	0	0	0	0	0
S-49	0	0	0	0	0	0
S-15	0	0	0	0	0	0
S-42	0.000503	0.001529	0	0	0	0
S-22	0.038083	0.049259	0.035211	0	0	0
S-47	0	0	0	0	0	0
S-43	0	0	0	0	0	0
S-48	0.039141	0.126202	0.225141	0.354441	0.528560	0.913394
S-3	0.025759	0.003450	0	0	0	0

Table 9
Proportion of investment on stocks obtained from SVR+MVaR model for different μ_{fix} values.

μ	**0.01**	**0.013**	**0.016**	**0.019**	**0.022**	**0.025**
VaR	**0.049869**	**0.049978**	**0.054053**	**0.064809**	**0.081867**	**0.117364**
σ	**0.034211**	**0.035987**	**0.040031**	**0.047891**	**0.059353**	**0.081351**
S-19	0	0	0	0	0	0
S-28	0.270176	0.243035	0.221930	0.168837	0.001716	0
S-29	0	0	0	0	0	0
S-39	0.140351	0.127204	0.058310	0	0	0
S-42	0	0.007675	0	0	0	0
S-24	0.002436	0	0	0	0	0
S-49	0	0	0	0	0	0
S-46	0.046485	0	0	0	0	0
S-48	0.042171	0.107317	0.168722	0.234018	0.313707	0.122320
S-50	0.046727	0.001840	0	0	0	0
S-27	0.155824	0.154557	0.120984	0.069951	0	0
S-41	0	0	0	0	0	0
S-22	0.045149	0.058390	0.046241	0.004906	0	0
S-44	0.011257	0	0	0	0	0
S-25	0	0	0	0.005125	0.041667	0.877680
S-47	0	0	0	0	0	0
S-3	0.022095	0.007716	0	0	0	0
S-32	0	0	0	0	0	0
S-36	0.069568	0.132066	0.214700	0.301118	0.388625	0
S-7	0.051509	0.040381	0	0	0	0
S-18	0	0	0	0	0	0
S-15	0	0	0	0	0	0
S-9	0	0.024145	0.073097	0.126331	0.169565	0
S-12	0	0	0	0	0	0
S-16	0	0	0	0	0	0
S-35	0.081210	0.060846	0.031771	0	0	0
S-37	0	0	0.018539	0.039394	0.040952	0
S-31	0	0	0	0	0	0
S-11	0	0	0	0	0	0
S-17	0.015043	0.034828	0.045707	0.050320	0.043768	0

Table 10
Proportion of investment on stocks obtained from KNN+MVaR model for different μ_{fix} values.

μ	**0.01**	**0.013**	**0.016**	**0.019**	**0.022**	**0.025**
VaR	**0.049541**	**0.049704**	**0.055033**	**0.066527**	**0.084558**	**0.121283**
σ	**0.034023**	**0.035831**	**0.040590**	**0.048873**	**0.060890**	**0.083590**
S-39	0.133687	0.126538	0.041152	0	0	0
S-29	0	0	0	0	0	0
S-46	0.058060	0	0	0	0	0
S-28	0.302687	0.277745	0.225522	0.105396	0	0
S-36	0.081246	0.133127	0.218527	0.324683	0.402785	0.086606
S-24	0.002196	0	0	0	0	0
S-44	0.018719	0	0	0	0	0
S-18	0	0	0	0	0	0
S-27	0.164780	0.170505	0.151685	0.099000	0	0
S-6	0.006806	0	0	0	0	0
S-41	0	0	0	0	0	0
S-38	0	0	0	0	0	0
S-7	0.052032	0.036864	0	0	0	0
S-11	0	0	0	0	0	0
S-45	0	0	0	0	0	0
S-47	0.016077	0	0	0	0	0
S-43	0	0	0	0	0	0
S-49	0	0	0	0	0	0
S-21	0.002906	0.020912	0.043276	0.067405	0.064175	0
S-48	0.037701	0.115228	0.216552	0.354441	0.528560	0.913394
S-32	0.002683	0	0	0	0	0
S-8	0	0	0	0	0	0
S-15	0	0	0	0	0	0
S-2	0	0.004783	0.020223	0.027631	0.004479	0
S-22	0.045674	0.048210	0.036577	0	0	0
S-23	0	0	0	0	0	0
S-50	0.052681	0.024171	0	0	0	0
S-40	0	0	0	0	0	0
S-4	0.022064	0.041915	0.046485	0.021445	0	0
S-16	0	0	0	0	0	0

Table 11
Proportion of investment on stocks obtained from ANN+MVaR model for different μ_{fix} values.

μ	**0.01**	**0.013**	**0.016**	**0.019**	**0.022**	**0.025**
VaR	**0.050648**	**0.053235**	**0.060750**	**0.072754**	**0.090544**	**0.132714**
σ	**0.034656**	**0.037849**	**0.043857**	**0.052431**	**0.064311**	**0.090122**
S-10	0.016931	0.053036	0.049637	0.002882	0	0
S-49	0	0	0	0	0	0
S-42	0.068511	0.091952	0.065551	0	0	0
S-5	0	0.005177	0.038428	0.088196	0.161157	0.335900
S-26	0	0	0.003022	0.041524	0.073963	0.102038
S-13	0	0	0	0	0	0
S-28	0.335337	0.345229	0.352239	0.291290	0.128597	0
S-31	0	0	0	0	0	0
S-11	0	0	0	0	0	0
S-4	0.050957	0.063344	0.063673	0.038373	0	0
S-46	0.003360	0	0	0	0	0
S-43	0	0	0	0	0	0
S-44	0.002872	0	0	0	0	0
S-8	0	0	0	0	0	0
S-39	0.175663	0.084075	0	0	0	0
S-14	0	0	0	0	0	0
S-16	0	0	0	0	0	0
S-1	0	0.023412	0.051969	0.061041	0.043577	0
S-41	0	0	0	0	0	0
S-36	0.108936	0.212795	0.300769	0.406900	0.529231	0.562062
S-24	0.015201	0	0	0	0	0
S-32	0.020069	0	0	0	0	0
S-35	0.085214	0.054386	0.026043	0	0	0
S-6	0	0	0	0	0	0
S-33	0	0	0	0	0	0
S-23	0	0	0	0	0	0
S-50	0.088782	0.025020	0	0	0	0
S-2	0	0.018483	0.047709	0.069792	0.063476	0
S-18	0	0	0	0	0	0
S-22	0.028167	0.023090	0.000959	0	0	0

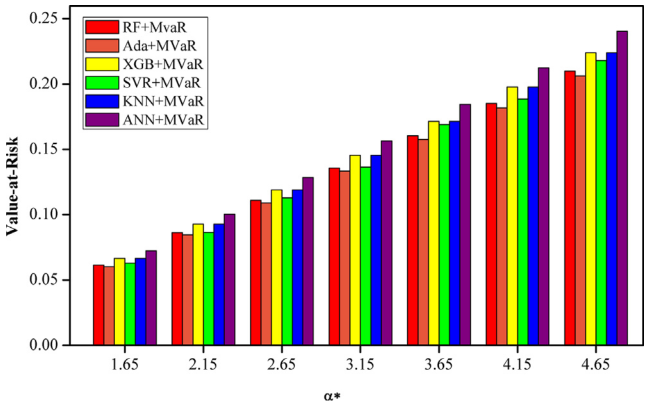


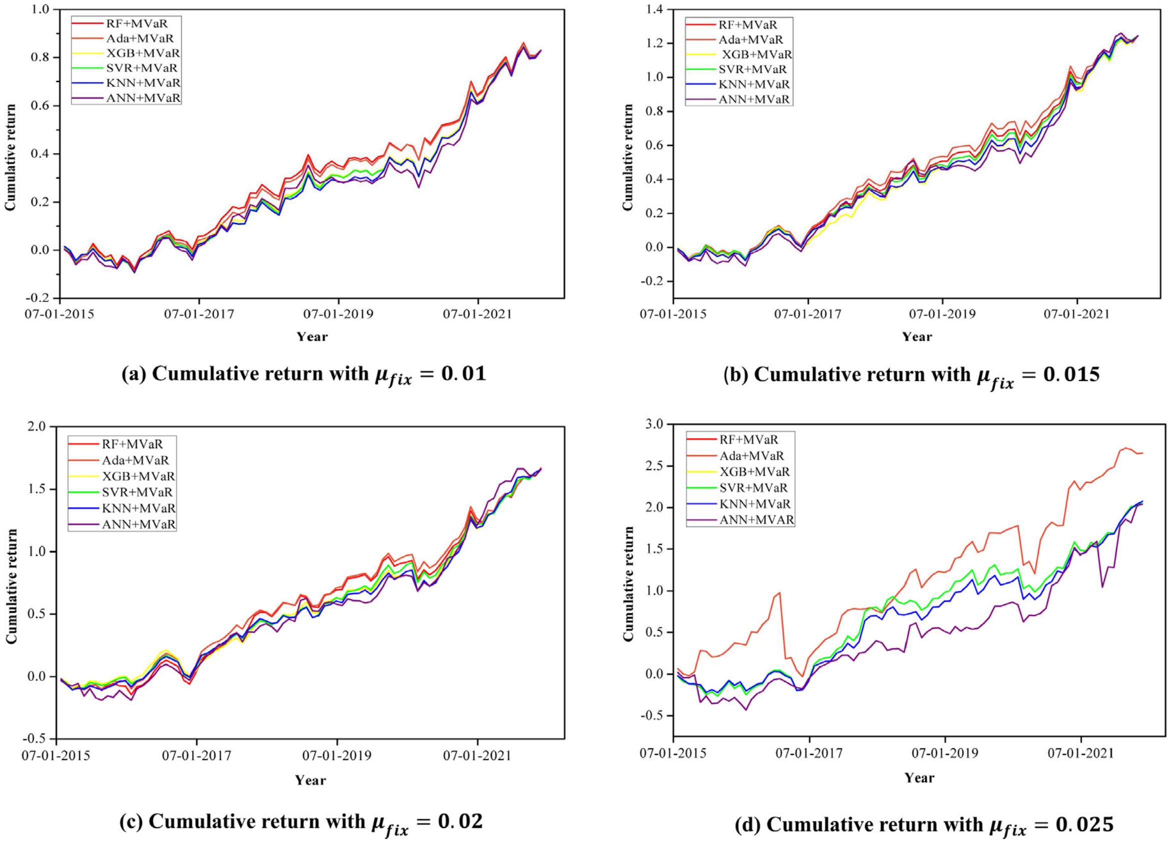
Fig. 6. α^*-VaR barplot for different models.

Meanwhile, the ANN+MVaR model has the lowest portfolio return as compared to the other models.

Moreover, the optimal VaR of the portfolio for the models RF+MVaR, Ada+MVaR, XGB+MVaR, SVR+MVaR, KNN+MVaR, and ANN+MVaR are obtained by varying α^* with a fixed value $\mu_{fix} = 0.02$. Based on these values, a comparison bar plot between the models is provided in Fig. 6. In the figure, the y-axis represents the different values of confidence level α^*. Meanwhile, the x-axis shows the optimal VaR value obtained from different models. From the figure, it is clearly observed that as the value of α^* increases, the value of optimal VaR for different models also increases. Moreover, as the value of α^* increases, the Ada+MVaR model shows the lowest VaR among the six models, followed by RF+MVaR, SVR+MVaR, XGB+MVaR, KNN+MVaR, and ANN+MVaR.

In order to provide a more accurate comparison between the models, Fig. 7 displays the cumulative returns of six models at different μ_{fix} values. As can be seen, Fig. 7 is made up of a total of four sub-graphs. The first sub-graph compares the cumulative returns of the models RF+MVaR, Ada+MVaR, SVR+MVaR, XGB+MVaR, KNN+MVaR, and ANN+MVaR with $\mu_{fix} = 0.01$, where the x-axis is representing the year and the y-axis is representing the cumulative returns of each model. It is clear from looking at the figure that RF+MVaR has the highest cumulative return, followed by Ada+MVaR, SVR+MVaR, XGB+MVaR, KNN+MVaR, and ANN+MVaR. In addition, the cumulative return for each model with μ_{fix} values of 0.015, 0.02, and 0.025 are represented in the second, third, and fourth sub-graphs of Fig. 7, respectively. The cumulative return for each model shows a significant increase throughout all three sub-graphs, with Ada+MVaR demonstrating the highest cumulative return across the models, followed by RF+MVaR, SVR+MVaR, XGB+MVaR, KNN+MVaR, and ANN+MVaR. Based on the findings, we can see that Ada+MVaR performs significantly better than the other models in terms of cumulative return.

Furthermore, the efficacy of the methodology is checked by using the stock data from TSE, Japan, and SSE, China. A set of 50 stocks is selected randomly from each of the markets (See Tables 12 and 13). Next, the same procedure is followed to get the solution. The cumulative return graphs for the models RF+MVaR, Ada+MVaR, SVR+MVaR, XGB+MVaR, KNN+MVaR, and ANN+MVaR at μ_{fix} value of 0.015 are provided in Fig. 8. The figure contains two subgraphs in which the first graph represents the cumulative return graphs for the TSE stock market and the second for the SSE stock market. It can be noticed that the Ada+MVaR model outperforms the other models in terms of cumulative return. Meanwhile, ANN+MVaR has the lowest cumulative return as compared to the other five models. The performance of Ada+MVaR is the best among the models due to the accurate prediction of the model AdaBoost compared to other models. Therefore, this work recommends



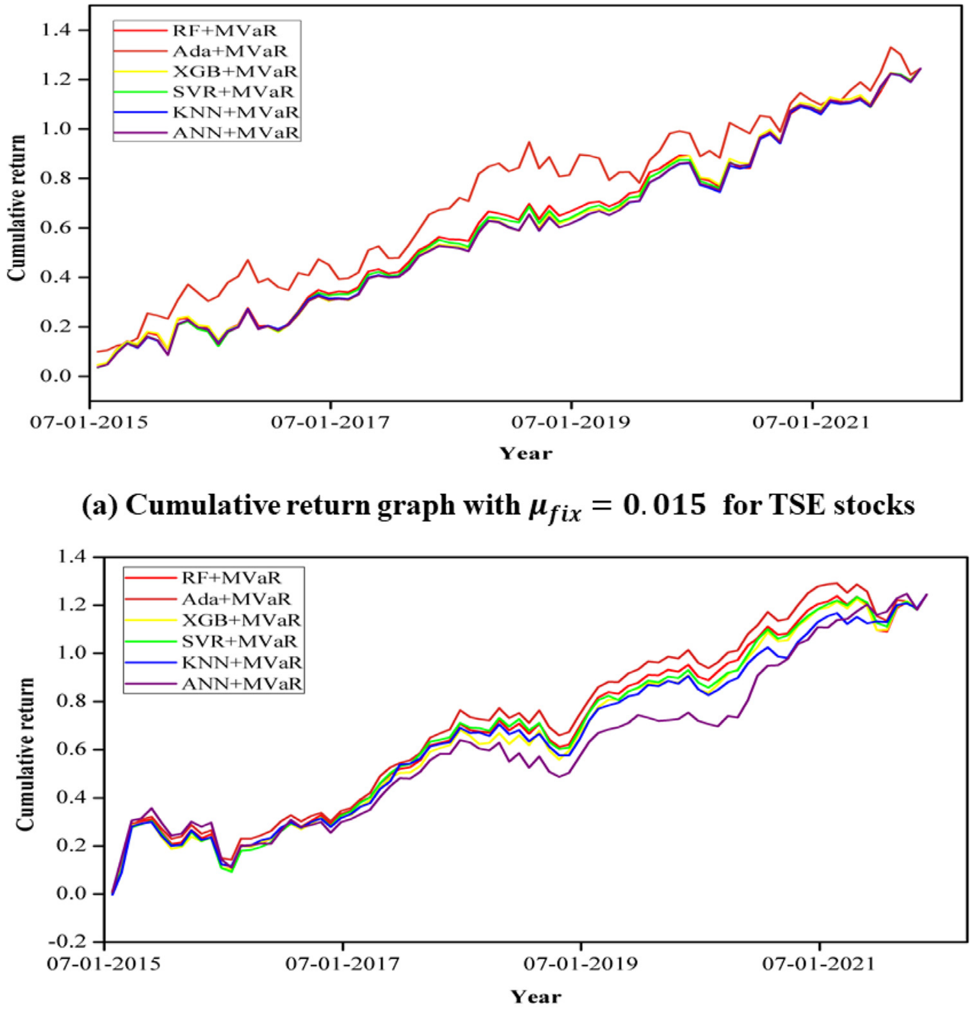
(a) Cumulative return with $\mu_{fix} = 0.01$

(b) Cumulative return with $\mu_{fix} = 0.015$

(c) Cumulative return with $\mu_{fix} = 0.02$

(d) Cumulative return with $\mu_{fix} = 0.025$

Fig. 7. Cumulative return graph obtained from each model for different μ_{fix} values.



(a) Cumulative return graph with $\mu_{fix} = 0.015$ for TSE stocks

(b) Cumulative return graph with $\mu_{fix} = 0.015$ for SSE stocks

Fig. 8. Cumulative return graph obtained for each model with $\mu_{fix} = 0.015$.

building the MVaR model with AdaBoost return prediction for monthly trading investment.

Apart from these results, we have compared our result with the mean–VaR efficiency (MVE) model and multi-objective mean–VaR (MOMV) model, which are presented in Banihashemi et al. (2016). To obtain the optimal investment strategies for MVE and MOMV models, the stocks mentioned in Table 5 are taken for the experiment. Accordingly, the same procedure has been followed with MVE and MOMV model for the values of $\mu_{fix} = 0.010, 0.013, 0.016, 0.019, 0.022, 0.025$. However, both MVE and MOMV models give non-feasible solutions at all the points except at 0.025. Meanwhile, at 0.025, the MOMV model shows an unbounded solution. However, our model gives feasible solutions at the values mentioned in the μ_{fix} values.

6. Conclusion

Investors need to have an efficient technique in recent days to help them make financial market decisions individually. Such a practice can encourage investors to invest in the financial markets and use their capital to improve their investment efficiently. Meanwhile, machine learning algorithms have become an essential tool in many fields to make more profitable and wiser decisions, including investment in the financial market. Thus this study proposes a portfolio construction technique based on stock return forecasting by combining stock prediction with financial investment. The significant findings of the suggested approach are as follows:

- Initially, it is explained in detail to identify the most suitable machine learning regression model among RF, AdaBoost, XGBoost, SVR, KNN, and ANN, which guarantees the selection of high-quality stocks by forecasting stock returns and, through these, derives a portfolio optimization model.
- Further, an efficient investment strategy based on the MVaR model is advanced with the return predicted by regression models, filling the research gap in the existing literature.
- An experimental analysis has been performed using the stock data taken from different world stock markets such as the Bombay Stock Exchange, India, Tokyo Stock Exchange, Japan, and Shanghai Stock Exchange, China. The experimental result shows that the Ada+MVaR model performs better among all the models.
- According to our results, the suggested approach can predict stock return with good accuracy and construct optimal portfolios based on investor preferences.

Despite the fact that this study offers some helpful insights, it does have some limitations. The following aspects of this work can be

Table 12
List of stock number and stock code for TSE stocks.

Stock No.	S-1	S-2	S-3	S-4	S-5	S-6	S-7	S-8	S-9	S-10
Code	1332.T	1605.T	1925.T	1803.T	2802.T	2801.T	2269.T	3402.T	3861.T	3407.T
Stock No.	S-11	S-12	S-13	S-14	S-15	S-16	S-17	S-18	S-19	S-20
Code	4208.T	4503.T	4519.T	5020.T	5019.T	5108.T	5803.T	5333.T	5406.T	5801.T
Stock No.	S-21	S-22	S-23	S-24	S-25	S-26	S-27	S-28	S-29	S-30
Code	5707.T	6367.T	6302.T	6857.T	7752.T	6724.T	6814.T	7267.T	7272.T	7762.T
Stock No.	S-31	S-32	S-33	S-34	S-35	S-36	S-37	S-38	S-39	S-40
Code	4543.T	8001.T	8015.T	7832.T	7951.T	8267.T	8233.T	8331.T	8355.T	8604.T
Stock No.	S-41	S-42	S-43	S-44	S-45	S-46	S-47	S-48	S-49	S-50
Code	8750.T	8697.T	8802.T	3289.T	9064.T	9433.T	9502.T	9532.T	2432.T	4707.T

Table 13
List of stock number and stock code for SSE stocks.

Stock No.	S-1	S-2	S-3	S-4	S-5	S-6	S-7	S-8	S-9	S-10
Code	600000	600028	600030	600031	600036	600048	600104	600111	600196	600276
Stock No.	S-11	S-12	S-13	S-14	S-15	S-16	S-17	S-18	S-19	S-20
Code	600309	600346	600436	600438	600519	600570	600585	600588	600690	600745
Stock No.	S-21	S-22	S-23	S-24	S-25	S-26	S-27	S-28	S-29	S-30
Code	600809	600837	600887	600893	600900	600273	601012	600248	601088	600233
Stock No.	S-31	S-32	S-33	S-34	S-35	S-36	S-37	S-38	S-39	S-40
Code	601166	600171	601288	601318	601336	601398	601601	601628	601633	601668
Stock No.	S-41	S-42	S-43	S-44	S-45	S-46	S-47	S-48	S-49	S-50
Code	601688	600131	601857	601888	601899	601919	600039	603288	600282	600299

improved and extended further. The proposed approach is based solely on the MVaR portfolio optimization model. However, for portfolio optimization, multiple risks as objectives, such as semi-variance, mean-absolute deviation, and CVaR, can be considered to more accurately represent the actual investment experience. Moreover, this work uses simple historical returns as the input feature. In light of this, future research could use input features that are more effective, such as economic indicators, technical indicators, news, and so on, to train predictive models and improve the performance of the MVaR model.

CRediT authorship contribution statement

Jyotirmayee Behera: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Ajit Kumar Pasayat:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Harekrushna Behera:** Conception and design of study, Writing – original draft, Writing – review & editing. **Pankaj Kumar:** Conception and design of study, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thanks the referees for their comments and suggestions that led the paper into the current form. All authors approved the version of the manuscript to be published.

Appendix

A.1. Nomenclature

$T(x)$	Regression trees.
M	Number of trees.
$D_t(i)$	Weight of the distribution for case i at iteration t.
$h_t(x)$	Weak learner's error.
Z_t	Normalization factor.
$y^{(t)}_{pred}$	Predicted value at tth training step.
y_{truth}	Ground truth value.
$l(y^{(t)}_{pred}, y_{truth})$	Loss value.
$\Omega(f_k)$	Complexity of kth tree.
γ	Regularization parameter.
λ	Regularization parameter.
$\|w\|^2$	ℓ_2- Norm of all leaf scores.
ϵ	Precision parameter.
ξ	Training error above ϵ.
ξ^*	Training error below $-\epsilon$.
C	Regularization parameter.
$K(x_i, x_j)$	Kernel function.
Z_j	Output from jth unit.
g	Activation function.
w_{ji}	Weight connects the unit j to unit i.
T	Time period ($t = 1, 2, \dots, T$).
r_i	Rate of return of the ith stock.
R	Total return of the portfolio.
μ_i	Expected rate of return of the ith stock.
μ_p	Expected rate of return of the portfolio.
σ_{ij}	Covariance between ith and jth stocks.
σ_p	Standard deviation of the portfolio.
x_i	Proportion of investment of asset i.

References

Ahmadi-Javid, A., Fallah-Tafti, M., 2019. Portfolio optimization with entropic value-at-risk. European J. Oper. Res. 279 (1), 225–241.

Alabdullah, A.A., Iqbal, M., Zahid, M., Khan, K., Amin, M.N., Jalal, F.E., 2022. Prediction of rapid chloride penetration resistance of metakaolin based high strength concrete using light GBM and XGBoost models by incorporating SHAP analysis. Constr. Build. Mater. 345, 128296.

Alexander, G.J., Baptista, A.M., 2002. Economic implications of using a mean-VaR model for portfolio selection: A comparison with mean-variance analysis. J. Econom. Dynam. Control 26 (7–8), 1159–1193.

Alexander, G.J., Baptista, A.M., 2004. A comparison of VaR and CVaR constraints on portfolio selection with the mean-variance model. Manage. Sci. 50 (9), 1261–1273.

Aslan, H.İ., Yılmaz, A.B., Jeong, N., Lee, S., Choi, C., 2022. Prediction of number of suicidal people based on KNN. In: 2022 International Conference on Electronics, Information, and Communication. ICEIC, IEEE, pp. 1–4.

Banihashemi, S., Moayedi Azarpour, A., Navvabpour, H., 2016. Portfolio optimization by mean-value at risk framework. Appl. Math. Inf. Sci. 10 (5), 1935–1948.

Behera, J., Pasayat, A.K., Behera, H., 2022. COVID-19 vaccination effect on stock market and death rate in India. Asia-Pac. Financ. Mark. 1–23.

Björk, T., Murgoci, A., Zhou, X.Y., 2014. Mean–variance portfolio optimization with state-dependent risk aversion. Math. Finance: Int. J. Math. Stat. Financ. Econ. 24 (1), 1–24.

Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. J. Econometrics 31 (3), 307–327.

Boulesteix, A.-L., Schmid, M., 2014. Machine learning versus statistical modeling. Biom. J. 56 (4), 588–593.

Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time Series Analysis: Forecasting and Control. John Wiley & Sons.

Çela, E., Hafner, S., Mestel, R., Pferschy, U., 2021. Mean-variance portfolio optimization based on ordinal information. J. Bank. Financ. 122, 105989.

Chelgani, S.C., Nasiri, H., Alidokht, M., 2021. Interpretable modeling of metallurgical responses for an industrial coal column flotation circuit by XGBoost and SHAP-A "conscious-lab" development. Int. J. Mining Sci. Technol. 31 (6), 1135–1144.

Cont, R., Deguest, R., Scandolo, G., 2010. Robustness and sensitivity analysis of risk measurement procedures. Quant. Finance 10 (6), 593–606.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.

Day, M.-Y., Lin, J.-T., 2019. Artificial intelligence for ETF market prediction and portfolio optimization. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 1026–1033.

Deng, S., Min, X., 2013. Applied optimization in global efficient portfolio construction using earning forecasts. J. Invest. 22 (4), 104–114.

Drucker, H., Burges, C.J., Kaufman, L., Smola, A., Vapnik, V., 1996. Support vector regression machines. Adv. Neural Inf. Process. Syst. 9.

Elavarasan, D., Vincent, D.R., Sharma, V., Zomaya, A.Y., Srinivasan, K., 2018. Forecasting yield by integrating agrarian factors and machine learning models: A survey. Comput. Electron. Agric. 155, 257–282.

Engle, R.F., 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica 987–1007.

Freitas, F.D., De Souza, A.F., De Almeida, A.R., 2009. Prediction-based portfolio optimization model using neural networks. Neurocomputing 72 (10–12), 2155–2170.

Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. System Sci. 55 (1), 119–139.

Garcia, R.C., Contreras, J., Van Akkeren, M., Garcia, J.B.C., 2005. A GARCH forecasting model to predict day-ahead electricity prices. IEEE Trans. Power Syst. 20 (2), 867–874.

Gawri, B., Kasturi, A., Neti, L.B.M., Hota, C., 2022. An efficient approach to kNN algorithm for IoT devices. In: 2022 14th International Conference on COMmunication Systems & NETworkS. COMSNETS, IEEE, pp. 734–738.

Gupta, P., Mehlawat, M.K., Mittal, G., 2012. Asset portfolio optimization using support vector machines and real-coded genetic algorithm. J. Global Optim. 53 (2), 297–315.

Ho, T.K., 1995. Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, Vol. 1. IEEE, pp. 278–282.

Ho, W.K., Tang, B.-S., Wong, S.W., 2021. Predicting property prices with machine learning algorithms. J. Prop. Res. 38 (1), 48–70.

Jiang, Z., Ji, R., Chang, K.-C., 2020. A machine learning integrated portfolio rebalance framework with risk-aversion adjustment. J. Risk Financ. Manag. 13 (7), 155.

Kaczmarek, T., Perez, K., 2021. Building portfolios based on machine learning predictions. Econ. Res.-Ekon. Istraž. 1–19.

Kamil, A.A., Ibrahim, K., 2007. Mean-absolute deviation portfolio optimization problem. J. Inf. Optim. Sci. 28 (6), 935–944.

Katris, C., 2021. A time series-based statistical approach for outbreak spread forecasting: Application of COVID-19 in Greece. Expert Syst. Appl. 166, 114077.

Katsikis, V.N., Mourtas, S.D., Stanimirović, P.S., Li, S., Cao, X., 2021. Time-varying mean-variance portfolio selection under transaction costs and cardinality constraint problem via beetle antennae search algorithm (BAS). In: Operations Research Forum, Vol. 2. Springer, pp. 1–26.

Kavzoglu, T., Teke, A., 2022. Predictive Performances of ensemble machine learning algorithms in landslide susceptibility mapping using random forest, extreme gradient boosting (XGBoost) and natural gradient boosting (NGBoost). Arab. J. Sci. Eng. 1–19.

Kenyon, C.M., Savage, S., Ball, B., 1999. Equivalence of linear deviation about the mean and mean absolute deviation about the mean objective functions. Oper. Res. Lett. 24 (4), 181–185.

Konno, H., Yamazaki, H., 1991. Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. Manage. Sci. 37 (5), 519–531.

Korkmaz, D., Acikgoz, H., 2022. An efficient fault classification method in solar photovoltaic modules using transfer learning and multi-scale convolutional neural network. Eng. Appl. Artif. Intell. 113, 104959.

Kou, S., Peng, X., Heyde, C.C., 2013. External risk measures and Basel accords. Math. Oper. Res. 38 (3), 393–417.

Kumar, P., Behera, J., Bhurjee, A., 2022. Solving mean-VaR portfolio selection model with interval-typed random parameter using interval analysis. OPSEARCH 59 (1), 41–77.

Liagkouras, K., Metaxiotis, K., 2018. Multi-period mean–variance fuzzy portfolio optimization model with transaction costs. Eng. Appl. Artif. Intell. 67, 260–269.

Lin, G., Lin, A., Gu, D., 2022. Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient. Inform. Sci. 608, 517–531.

Liu, Y., Ahmadzade, H., Farahikia, M., 2021. Portfolio selection of uncertain random returns based on value at risk. Soft Comput. 25 (8), 6339–6346.

Liu, Y.-C., Yeh, I.-C., 2017. Using mixture design and neural networks to build stock selection decision support systems. Neural Comput. Appl. 28 (3), 521–535.

Ma, Y., Han, R., Wang, W., 2021. Portfolio optimization with return prediction using deep learning and machine learning. Expert Syst. Appl. 165, 113973.

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

Mishra, R.K., Choudhary, A., Fatima, S., Mohanty, A.R., Panigrahi, B.K., 2022a. A self-adaptive multiple fault diagnosis system for rolling element bearings. Meas. Sci. Technol..

Mishra, R.K., Choudhary, A., Mohanty, A., Fatima, S., 2022b. An intelligent bearing fault diagnosis based on hybrid signal processing and Henry gas solubility optimization. Proc. Inst. Mech. Eng. C 09544062221101737.

Moon, Y., Yao, T., 2011. A robust mean absolute deviation model for portfolio optimization. Comput. Oper. Res. 38 (9), 1251–1258.

Padhi, D.K., Padhy, N., Bhoi, A.K., Shafi, J., Yesuf, S.H., 2022. An intelligent fusion model with portfolio selection and machine learning for stock market prediction. Comput. Intell. Neurosci. 2022.

Pant, M., Kumar, S., 2022. Fuzzy time series forecasting based on hesitant fuzzy sets, particle swarm optimization and support vector machine-based hybrid method. Granul. Comput. 7 (4), 861–879.

Pasayat, A.K., Mitra, A., Bhowmick, B., 2022. Determination of essential features for predicting start-up success: an empirical approach using machine learning. Technol. Anal. Strateg. Manag. 1–19.

Qin, Z., Wen, M., Gu, C., 2011. Mean-absolute deviation portfolio selection model with fuzzy returns. Iran. J. Fuzzy Syst. 8 (4), 61–75.

Qiu, Y., Garg, D., Kim, S.-M., Mudawar, I., Kharangate, C.R., 2021. Machine learning algorithms to predict flow boiling pressure drop in mini/micro-channels based on universal consolidated data. Int. J. Heat Mass Transfer 178, 121607.

Rosadi, D., Setiawan, E.P., Templ, M., Filzmoser, P., 2020. Robust covariance estimators for mean-variance portfolio optimization with transaction lots. Oper. Res. Perspect. 7, 100154.

Sahoo, R., Pasayat, A.K., Bhowmick, B., Fernandes, K., Tiwari, M.K., 2022. A hybrid ensemble learning-based prediction model to minimise delay in air cargo transport using bagging and stacking. Int. J. Prod. Res. 60 (2), 644–660.

Sartakhti, J.S., Afrabandpey, H., Ghadiri, N., 2019. Fuzzy least squares twin support vector machines. Eng. Appl. Artif. Intell. 85, 402–409.

Schapire, R.E., 1999. A brief introduction to boosting. In: Ijcai, Vol. 99. Citeseer, pp. 1401–1406.

Shang, M., Li, H., Ahmad, A., Ahmad, W., Ostrowski, K.A., Aslam, F., Joyklad, P., Majka, T.M., 2022. Predicting the mechanical properties of RCA-based concrete using supervised machine learning algorithms. Materials 15 (2), 647.

Sheng, Z., Benshan, S., Zhongping, W., 2012. Analysis of mean-VaR model for financial risk control. Syst. Eng. Procedia 4, 40–45.

Shephard, N., 2020. Statistical aspects of ARCH and stochastic volatility. In: Time Series Models. Chapman and Hall/CRC, pp. 1–67.

Song, X., Mitnitski, A., Cox, J., Rockwood, K., 2004. Comparison of machine learning techniques with classical statistical models in predicting health outcomes. In: MEDINFO 2004. IOS Press, pp. 736–740.

Tang, J., Henderson, A., Gardner, P., 2021. Exploring AdaBoost and random forests machine learning approaches for infrared pathology on unbalanced data sets. Analyst 146 (19), 5880–5891.

Uimonen, S., Tukia, T., Ekström, J., Siikonen, M.-L., Lehtonen, M., 2020. A machine learning approach to modelling escalator demand response. Eng. Appl. Artif. Intell. 90, 103521.

Ustun, O., Kasimbeyli, R., 2012. Combined forecasts in portfolio optimization: a generalized approach. Comput. Oper. Res. 39 (4), 805–819.

Villegas, M.A., Pedregal, D.J., Trapero, J.R., 2018. A support vector machine for model selection in demand forecasting applications. Comput. Ind. Eng. 121, 1–7.

Wang, D., Chen, Y., Wang, H., Huang, M., 2020. Formulation of the non-parametric value at risk portfolio selection problem considering symmetry. Symmetry 12 (10), 1639.

Wang, J.-Z., Wang, J.-J., Zhang, Z.-G., Guo, S.-P., 2011. Forecasting stock indices with back propagation neural network. Expert Syst. Appl. 38 (11), 14346–14355.

Xia, J., Wang, Z., Yang, D., Li, R., Liang, G., Chen, H., Heidari, A.A., Turabieh, H., Mafarja, M., Pan, Z., 2022. Performance optimization of support vector machine with oppositional grasshopper optimization for acute appendicitis diagnosis. Comput. Biol. Med. 143, 105206.

Yan, Z., Chen, H., Dong, X., Zhou, K., Xu, Z., 2022. Research on prediction of multi-class theft crimes by an optimized decomposition and fusion method based on XGBoost. Expert Syst. Appl. 207, 117943.

Yang, F., Chen, Z., Li, J., Tang, L., 2019. A novel hybrid stock selection method with stock prediction. Appl. Soft Comput. 80, 820–831.

Zhang, J., Li, Z., Pu, Z., Xu, C., 2018. Comparing prediction performance for crash injury severity among various machine learning and statistical methods. IEEE Access 6, 60079–60087.

Zhou, J., Zhu, S., Qiu, Y., Armaghani, D.J., Zhou, A., Yong, W., 2022. Predicting tunnel squeezing using support vector machine optimized by whale optimization algorithm. Acta Geotech. 17 (4), 1343–1366.