Accounting 10 (2024) 207–220

Contents lists available at GrowingScience

Accounting

homepage: www.GrowingScience.com/ac/ac.html

Stock price prediction portfolio optimization using different risk measures on application of genetic algorithm for machine learning regressions

Amir Hossein Gandomi[a], Seyed Jafar Sadjadi[a*] and Babak Amiri[a]

[a]*School of Engineering, Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran*

C H R O N I C L E

Article history:
Received January 15, 2024
Received in revised format April 16 2024
Accepted July 9 2024
Available online
July 9 2024

Keywords:
Portfolio optimization
Stock market performance
Risk measures
Machine learning
Regression algorithms
Genetic algorithm

A B S T R A C T

This research aims to enhance portfolio selection by integrating machine learning regression algorithms for predicting stock returns with various risk measures. These measures include mean-value-at-risk (VaR) variance (Var), semi-variance mean-absolute-deviation (MAD) and conditional value-at-risk (C-VaR). Addressing gaps in existing literature. Traditional methods lack adaptability to dynamic market conditions. We propose a hybrid approach optimized by genetic algorithms. The study employs multiple machine learning models. These include Random Forest, AdaBoost XGBoost, Support Vector Machine Regression (SVR) K-Nearest Neighbors (KNN) and Artificial Neural Network (ANN). These models are used to forecast stock returns. Utilizing monthly data from the Tehran Stock Exchange, the results indicate that the genetic algorithm prediction model combined with mean-VaR, Var semi-variance and MAD, produces the most efficient portfolios. These portfolios offer superior returns with minimized risk compared to other models. This hybrid strategy provides a robust and efficient method for investors aiming to optimize returns while managing risk effectively. To implement this approach successfully it is crucial to balance investments. This involves both traditional and alternative asset classes, ensuring diversification. It also capitalizes on market opportunities. Regular review and adjustment of fund allocation are essential. Maintain an optimized strategy for maximum returns and minimal risk.

1. Introduction

Stock market performance is critical in optimization of portfolios. Investors aim to maximize returns while minimizing risk. Traditional models like the Markowitz mean-variance model have been widely used to optimize portfolios. They evaluate returns based on meaning. And risk through variance (Lewellen, 2014). However, these models are often insufficient when dealing with asymmetric returns. This necessitates the use of more robust risk measures such as mean absolute deviation (AD). Like ships navigating through rough waters. These models can be easily swayed by asymmetric returns. A more reliable guide, like mean absolute deviation (AD) is required to stay on course (Amihud, 2012). Advances in machine learning have led to the development of innovative approaches for improving portfolio selection through predictive analytics. Machine learning regression algorithms such as random forest extreme gradient boosting (XGBoost) and adaptive boosting (AdaBoost). Support vector machine regression (SVR). K-nearest neighbor (KNN) and artificial neural networks (ANN) can predict stock returns. Additionally, genetic algorithms which are more powerful tools can also be employed. These advancements can aid investors. They help in making informed decisions. Despite these advancements, there remains a gap in integration of precise stock return projections and efficient portfolio optimization. Current literature often addresses these aspects in isolation. They lack a cohesive strategy. Combining predictive prowess of machine learning with robust risk management techniques. This research aims to bridge this gap by proposing novel hybrid methodology that leverages machine learning regression methodologies alongside advanced risk optimization paradigms. Furthermore, the study's uniqueness lies in its hybrid methodology. It leverages strengths of machine learning to improve predictive accuracy and robustness of risk metrics in risk management (Schuett, 2023; Kreibich et al., 2022). By employing monthly dataset from

* Corresponding author.
E-mail address: sjsadjadi@iust.ac.ir (S.J. Sadjadi)

doi: 10.5267/j.ac.2024.7.002

Tehran Stock Exchange study tests proposed approach. The findings highlight the potential of genetic algorithm models. When combined with risk frameworks, it can outperform other models. This occurs largely in portfolio optimization. This research aims to address the issue of combining precise stock return projections with efficient portfolio optimization. By integrating machine learning regression algorithms. With advanced risk measures. The study seeks to construct portfolios. Ensuring diversified and optimized portfolios capable of capitalizing on market opportunities while managing risk effectively. The goal is to provide a robust and efficient strategy for investors.

The reminder of this paper is organized as follows. Section 2 presents the literature review, providing an overview of traditional and contemporary models of stock price prediction and portfolio optimization. Section 3 discusses the methodology used in this study, detailing the data collection process, machine learning algorithms employed, and the integration of genetic algorithms. Section 4 describes the empirical analysis and results, highlighting the performance of different models and risk measures. Section 5 offers a comprehensive discussion of the findings, relating them to existing literature and addressing the research questions posed. Finally, Section 6 concludes the paper, summarizing the key insights and suggesting directions for future research.

2. Literature review

2.1 Literature Review: Stock Price Prediction and Portfolio Optimization

The optimization of stock portfolios and prediction of stock prices have been perennial topics of interest in financial research. Traditional models such as Markowitz mean-variance model have long been used to balance risk and return. However, the advent of machine learning opened new avenues for enhancing these models. This literature review explores recent advancements in the integration of machine learning algorithms and genetic algorithms in predicting stock returns. It also examines portfolio optimization using various risk measures.

2.1.1 Traditional Portfolio Optimization Models

Harry Markowitz's mean-variance model remains the cornerstone of modern portfolio theory. It evaluates returns based on their mean and assesses risk through variance. However, the model's reliance on variance as the sole measure of risk is a significant limitation. This is true particularly when dealing with asymmetric returns. To address these limitations several alternative risk measures have been proposed.

1. Mean Absolute Deviation (MAD): Introduced by Konno and Yamazaki. This model replaces variance with mean absolute deviation. It provides a more robust risk measure in the presence of outliers.

2. Value-at-Risk (VaR): VaR focuses on the maximum potential loss over a specified period. It evaluates this over a given confidence level. This approach addresses some limitations of the mean-variance model by considering extreme losses.

3. Conditional Value-at-Risk (CVaR): Also known as expected shortfall CVaR measures the expected loss exceeding the VaR It offers a more comprehensive risk assessment of tail distributions.

4. Shannon Entropy: This measure captures risk in terms of information theory. It assesses the uncertainty or entropy in the return distribution.

5. Beta Measures: Beta quantifies the sensitivity of a portfolio's returns to market returns. It provides insight into systematic risk.

6. Exponential Smoothing: This technique predicts future values. It does so by smoothing past returns and giving more weight to recent data.

2.1.2 Machine Learning in Stock Price Prediction

Advancements in machine learning have introduced innovative approaches for stock price prediction. Various algorithms including Random Forest, XGBoost AdaBoost, Support Vector Machine Regression (SVR) K-Nearest Neighbors (KNN) and Artificial Neural Networks (ANN), have been utilized to forecast stock returns.

Random Forest: An ensemble learning method that improves predictive accuracy by averaging multiple decision trees.

XGBoost: An efficient implementation of gradient boosting that has shown superior performance in predictive tasks.

AdaBoost: Adaptive boosting that combines weak learners to form a strong predictor. It is particularly effective in handling complex financial data.

Support Vector Machine Regression (SVR): SVR uses hyperplanes in a high-dimensional space. It helps to perform regression tasks beneficial for non-linear data.

K-Nearest Neighbors (KNN): A simple, instance-based learning algorithm. It predicts the value based on k-nearest data points.

Artificial Neural Networks (ANN): Inspired by the human brain ANNs capable of capturing complex patterns in large datasets. This makes them suitable for stock price prediction.

2.1.3 Hybrid Models: Combining Machine Learning and Genetic Algorithms

Recent research has focused on integrating machine learning predictions with genetic algorithms for enhanced portfolio optimization. For instance, Behera et al. (2023) demonstrated that combining genetic algorithms with machine learning regression algorithms, such as AdaBoost, significantly improves the accuracy of stock return predictions and portfolio optimization outcomes.

2.2 Literature Review: Machine Learning and Data Analysis Applications

The field of machine learning and data analysis has seen significant advancements in recent years, with various methodologies and applications being explored. This literature review aims to summarize and analyze recent research articles focusing on different aspects of machine learning, data analysis, and their applications in various domains.

2.2.1 Review of Literature

1. Karimov et al. (2023) explored the significance of input features for domain adaptation of spacecraft data. Their study, published in **Cosmic Research**, highlights the importance of selecting appropriate input features to improve the performance of machine learning models in space-related applications.
2. Wadekar and Chaurasia (2022) introduced MobileViTv3, a mobile-friendly vision transformer that effectively fuses local, global, and input features. This work, available on **arXiv**, demonstrates the potential of vision transformers in mobile applications.
3. Wen et al. (2022) utilized XGBoost regression to analyze the importance of input features in an AI model for biomass gasification systems. Published in **Inventions**, their research underscores the utility of feature importance analysis in optimizing AI models for industrial applications.
4. Abadi et al. (2016) presented TensorFlow, a framework for large-scale machine learning on heterogeneous distributed systems. This seminal work, available on **arXiv**, has become a cornerstone in the development and deployment of machine learning models.
5. Xiao et al. (2017) introduced Fashion-MNIST, a novel image dataset for benchmarking machine learning algorithms. This dataset, described in **arXiv**, has become a standard benchmark for evaluating the performance of image classification models.
6. Rudin (2018) argued against the use of black-box machine learning models for high-stakes decisions, advocating for interpretable models instead. Published in **Nature Machine Intelligence**, this article emphasizes the need for transparency and interpretability in critical applications.
7. Brandhofer et al. (2022) benchmarked the performance of portfolio optimization using the Quantum Approximate Optimization Algorithm (QAOA). Their study, published in **Quantum Information Processing**, explores the potential of quantum computing in financial applications.
8. Song et al. (2023) proposed an enhanced distributed differential evolution algorithm for portfolio optimization problems. This research, published in **Engineering Applications of Artificial Intelligence**, highlights the effectiveness of evolutionary algorithms in financial optimization.
9. Erwin and Engelbrecht (2023) reviewed meta-heuristics for portfolio optimization, providing a comprehensive overview of various optimization techniques. Their work, published in **Soft Computing**, serves as a valuable resource for researchers in the field of financial optimization.
10. Álvarez et al. (2023) conducted a peer review of the pesticide risk assessment of glyphosate, published in the **EFSA Journal**. This study provides critical insights into the environmental and health impacts of pesticide use.

Table 1
Literature review of Machine learning and data analysis applications

Author(s)	Year	Title	Journal/Book	Key Points
Karimov et al.	2023	The Significance of Input Features for Domain Adaptation of Spacecraft Data	Cosmic Research	Importance of input features in domain adaptation for spacecraft data.
Wadekar & Chaurasia	2022	MobileViTv3: Mobile-Friendly Vision Transformer	arXiv	Introduction of a mobile-friendly vision transformer with effective feature fusion.
Wen et al.	2022	Using XGBoost Regression for Biomass Gasification System	Inventions	Analysis of input feature importance using XGBoost regression.
Abadi et al.	2016	TensorFlow: Large-Scale Machine Learning	arXiv	Development of TensorFlow for large-scale machine learning on distributed systems.
Xiao et al.	2017	Fashion-MNIST: a Novel Image Dataset	arXiv	Introduction of Fashion-MNIST dataset for benchmarking image classification algorithms.
Rudin	2018	Stop explaining black box machine learning models	Nature Machine Intelligence	Advocacy for interpretable models over black-box models in high-stakes decisions.
Brandhofer et al.	2022	Benchmarking the performance of portfolio optimization with QAOA	Quantum Information Processing	Exploration of quantum computing for portfolio optimization.
Song et al.	2023	Enhanced Distributed Differential Evolution Algorithm	Engineering Applications of Artificial Intelligence	Proposal of an enhanced algorithm for portfolio optimization.
Erwin & Engelbrecht	2023	Meta-heuristics for portfolio optimization	Soft Computing	Comprehensive review of meta-heuristics in financial optimization.
Álvarez et al.	2023	Peer review of the pesticide risk assessment of glyphosate	EFSA Journal	Critical review of the environmental and health impacts of glyphosate.

3. Preliminary

In this part, we have divided the preparations into 3 parts. In the first part, we examined the optimization of the stock portfolio and various risk metrics. In the second part, we have investigated machine regression approaches. In the third part, we have investigated the genetic algorithm in stock portfolio optimization.

Sec.1: Portfolio Optimization and Risk Metrics

The field of portfolio optimization traces its origins to Harry Markowitz's mean–variance (M-V) model, introduced in the 1950s. This groundbreaking model incorporates variance as a risk metric and aims to maximize returns for a specified level of risk. Markowitz's model forms the cornerstone of modern portfolio theory by highlighting the essential relationship between risk and returns. In this context, variance signifies the dispersion of returns around the mean, serving as a measure of the portfolio's inherent risk. Since the introduction of the M-V model, several advancements and alternative risk measures have been proposed to address its limitations and provide a more comprehensive view of risk. These include:

Sec.1.2 Mean Absolute Deviation (MAD) Model

Developed by Kono and Yamazaki, the MAD model uses the mean absolute deviation instead of variance to measure risk. This provides a linear and more robust alternative to variance, especially in the presence of outliers.

$$MAD = \frac{1}{N}\sum_{i=1}^{N}|R_i - \bar{R}| \tag{1}$$

- (N): Number of observations
- (R_i): Return of the (i^{th}) observation
- $(\bar{R})$: Mean return of the portfolio

Sec.1.3 Value-at-Risk (VaR) Model

Developed by Sims, VaR focuses on the maximum potential loss over a specified period for a given confidence level. It addresses some limitations of the M-V model by concentrating on extreme losses.

$$VaR\alpha = -inf x \in R: P(L \leq x) \geq \alpha \tag{2}$$

- (α): Confidence level
- (L): Loss
- (P): Probability function
- (x): Potential loss value

Sec.1.4 Conditional Value-at-Risk (CVaR)

CVaR, also known as expected shortfall, measures the expected loss exceeding the VaR and provides a more comprehensive risk assessment of tail distributions.

$$\left[CVaR_\alpha = E[\, L \mid L \geq VaR_\alpha \,]\right] \tag{3}$$

- (α): Confidence level
- (L): Loss
- (E): Expected value function
- (VaR_α): Value-at-Risk at confidence level (α)

Sec.1.5 Shannon Entropy

This measure assesses the uncertainty or entropy in the return distribution and captures risk in terms of information theory.

$$\boldsymbol{H(X) = -\sum_i p(xi)\, log\, p(xi)} \tag{4}$$

- (X): Random variable representing returns
- $\left(p(x_i)\right)$: Probability of the(i^{th}) return
- (x_i): (i^{th}) return value

Sec.1.6 Beta Measures

Beta measures the sensitivity of a portfolio's returns to market returns and provides insight into systematic risk.

$$\left(\boldsymbol{\beta} = \frac{\mathrm{Cov}(\boldsymbol{R_i}, \boldsymbol{R_m})}{\mathrm{Var}(\boldsymbol{R_m})}\right) \tag{5}$$

- (R_i): Return of the portfolio
- (R_m): Return of the market
- $\big(\mathrm{Cov}(R_i, R_m)\big)$: Covariance between the portfolio and market returns
- $\big(\mathrm{Var}(R_m)\big)$: Variance of the market returns

Sec.1.7 Exponential Smoothing

This technique smooths past returns to predict future values, accounting for recent changes more heavily than older data do.

$$\boldsymbol{St = \alpha Rt + (1 - \alpha) St - 1} \tag{6}$$

- (S_t): Smoothed value at time (t)
- (α): Smoothing constant
- (R_t): Return at time (t)
- (S_{t-1}): Smoothed value at time $(t-1)$

SEC.2 GENETIC ALGORITHMS IN PORTFOLIO OPTIMIZATION

Genetic algorithms (GAs) are a class of optimization techniques inspired by natural selection and genetics, utilized to solve complex optimization problems, including portfolio optimization. GAs are particularly useful for their ability to explore large solution spaces and find near-optimal solutions efficiently.

Sec.2.1 Introduction to Genetic Algorithms:

Genetic algorithms work by evolving a population of candidate solutions through selection, crossover, and mutation operations to optimize an objective function.

In portfolio optimization, GAs aim to maximize returns while minimizing risk by optimizing the allocation of assets.

Sec.2.2 Working Mechanism of Genetic Algorithms:

Initialization: Begin with a randomly generated population of potential solutions.

Selection: Choose individuals based on their fitness scores to form a mating pool. The fitness function typically evaluates the quality of solutions, such as portfolio returns adjusted for risk.

Crossover (Recombination): Combine pairs of individuals (parents) from the mating pool to produce offspring, incorporating features from both parents.

- $If\big(P_1 = (x_1, y_1)\big) and \big(P_2 = (x_2, y_2)\big)$ are parents, offspring $(O_1) and (O_2) can\ be\ created\ as: O1 = (\alpha x1 + (1-\alpha)x2, \alpha y1 + (1-\alpha)y2) O1 = (\alpha x1 + (1-\alpha)x2, \alpha y1 + (1-\alpha)y2) O2 = (\alpha x2 + (1-\alpha)x1, \alpha y2 + (1-\alpha)y1) O2 = (\alpha x2 + (1-\alpha)x1, \alpha y2 + (1-\alpha)y1)$ where (α) is a random crossover point between 0 and 1.

Mutation: Introduce random alterations to the offspring to maintain genetic diversity.

- $If\big(O = (x, y)\big)$ is an offspring, mutation could change it to: $O' = (x + \Delta x, y + \Delta y) O' = (x + \Delta x, y + \Delta y)$ where (Δx) and (Δy) are small random values.

Evaluation: Assess the new population's fitness, replacing fewer fit individuals with better-performing offspring.

Termination: Repeat the selection, crossover, mutation, and evaluation steps until a stopping criterion is met, such as a fixed number of generations or a satisfactory fitness level.

Application in Portfolio Optimization:

GAs have been integrated with various risk measures, such as VaR and CVaR, to improve the accuracy of portfolio optimization models.

Empirical research has demonstrated that combining GAs with risk assessment models can enhance yield prediction and risk management. For instance, studies using data from the Tehran Stock Exchange have shown that models incorporating GAs outperform those without in terms of yield prediction and risk management (Song et al., 2023; Erwin & Engelbrecht, 2023).

Methodological Considerations:

Experimental methodologies in GA-based portfolio optimization often involve historical data analysis, including preprocessing steps like data cleaning, integration, and outlier removal to ensure reliable predictions.

Typically, datasets are divided into training and testing sets to validate the efficacy of the models (Karimov et al., 2023).

Sec.3: Machine Learning Approaches

Machine learning approaches have gained significant traction in the field of financial analysis, including stock portfolio optimization. Here, we explore several machine learning techniques: Random Forest, K-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), AdaBoost, XGBoost, and Support Vector Machines (SVM).

Sec.3.1 Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees.

$$\boldsymbol{f^{(x)} = \frac{1}{B} \sum_{b=1}^{B} fb(x)} \tag{7}$$

- (B): Number of trees
- $\left(f_b(x)\right)$: Prediction of the (b^{th}) tree

Sec.3.2 K-Nearest Neighbors (KNN)

KNN is a non-parametric algorithm that classifies or predicts the value of a sample based on the majority vote (classification) or average (regression) of its (k) nearest neighbors.

$$\boldsymbol{y = \frac{1}{k} \sum_{i=1}^{k} yi} \tag{8}$$

- (k): Number of nearest neighbors
- (y_i): Value of the (i^{th}) nearest neighbor

Sec.3.3 Artificial Neural Networks (ANN)

ANNs are computational models inspired by the human brain, consisting of layers of interconnected nodes (neurons) that learn to recognize patterns through training.

$$\boldsymbol{y = f\left(\sum_{i=1}^{n} wixi + b\right)} \tag{9}$$

- (f): Activation function
- (w_i): Weight for the (i^{th}) input
- (x_i): (i^{th}) input
- (b): Bias term

Sec. 3.4 AdaBoost

AdaBoost is an ensemble learning technique. It combines the predictions of several weak learners. This creates a strong learner. It adjusts the weights of incorrectly classified instances. It focuses on them in subsequent iterations

$$\mathbf{f^{(x)} = \sum_{m=1}^{M} \alpha mhm\,(x)} \tag{10}$$

- (M): Number of weak learners
- (α_m): Weight of the (m^{th}) weak learner
- $\left(h_m(x)\right)$: Prediction of the (m^{th}) weak learner

Sec. 3.5 XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting that optimizes performance. It enhances computational speed. It builds an ensemble of trees sequentially. Each tree corrects errors. These errors are made by previous ones.

$$\boldsymbol{y = \sum_{k=1}^{K} fk(x), fk \in \mathcal{F}} \tag{11}$$

- (K): Number of trees
- (f_k): Function (tree) from the set of all possible trees $(\mathcal{F})$

Sec.3.6 Support Vector Machines (SVM)

SVM is a supervised learning algorithm that finds the hyperplane that best separates classes (classification) or fits the data (regression).

$$\boldsymbol{f(x) = w \cdot x + b} \tag{12}$$

- (w): Weight vector
- (x): Input vector
- (b): Bias term

Evaluation Metrics

Each of these machine learning approaches can be evaluated using various performance metrics. It depends on whether the task is classification or regression. Common metrics include:

Mean Squared Error (MSE): Measures the average of the squares of the errors

$$\left(\text{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^{\mathbf{2}}\right) \tag{13}$$

- (n): Number of samples
- (y_i): Actual value
- $(\hat{y}_i)$: Predicted value

R-Squared (R^2): Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

$$\left(\boldsymbol{R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}\mathbf{i})^2}{\sum_{i=1}^{n}\left(y_i - (\bar{y})\right)^2}}\right) \tag{14}$$

- $(\bar{y})$: Mean of the actual values

Precision, Recall, F1-Score (for classification tasks):

- **Precision**: $\left(\frac{\text{True Positives}}{\text{True Positives+False Positives}}\right)$
- **Recall**: $\left(\frac{\text{True Positives}}{\text{True Positives+False Negatives}}\right)$
- **F1-Score**: $\left(\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision+Recall}}\right)$

4.Methodology

4.1 Research Design

This study uses a mixed approach, combining quantitative and qualitative techniques to ensure a comprehensive analysis of the research problem. The research will be conducted in the following steps:

Data Collection:

Quantitative Data: Monthly historical data of 4 randomly selected stocks from the Tehran Stock Exchange (TSE) for the period from April 2023 to March 2024. The dataset is divided into a training set (70%) and a test set (30%).
Qualitative Data: Positioning yourself as an individual investor to gather insights into their strategies and decision-making processes.

1. **Data Preprocessing**:

- Management of missing values through data cleaning.
- Data integration, feature selection, and creation of new variables.
- Removal of outliers using the capping method.

2. **Prediction of Stock Returns**:

- Using a genetic algorithm for machine learning.

- Evaluation of the accuracy of the model using the mean absolute error, mean squared error, and R-square measures.

3. **Portfolio Optimization**:

- Using the mean-value-at-risk (VaR), conditional value-at-risk (C-VaR), variance, and standard-deviation-absolute (AD) models to minimize risk and maximize returns.
- Integration of genetic algorithm predictions with mean-value-at-risk (VaR), conditional value-at-risk (C-VaR), variance, and absolute-standard-deviation (AD) models for optimal portfolio selection.

4.2 Methodology Justification

1.Quantitative Data Collection:

TSE stock selection provides a diverse and comprehensive dataset that reflects various market conditions. It also reflects stock performance trends. Historical data over a significant period from 2023 to 2024, ensures robustness of findings. This dataset enables the analysis of trends in different economic cycles.

2.Data Preprocessing:

•Data cleaning and preprocessing increases the quality and reliability of the dataset. Ensures accurate model predictions.
• Capping method is chosen. Remove outliers to reduce the impact of extreme values and ensure a more stable prediction model.

3.Machine Learning Models:

• The combination of genetic algorithms with risk models allows comprehensive comparison of risk models with and without different genetic algorithms. This ensures that the best model is selected based on performance criteria.
• The use of multiple models provides robustness. Cross-validation of the results increases the reliability of predicting stock returns.

4.Portfolio Optimization:

• Risk models are a widely recognized approach to financial risk management. Effectively balancing return and risk.
• Integrating machine learning genetic algorithm predictions with risk models using advanced analytical techniques. This improves portfolio performance and provides strategic advantages for investors.

4.3 Experimental Process

1.Training and Testing:

• The dataset is divided into training (70%) and test (30%) sets. This separation helps in training models and evaluating their performance.
• Models are implemented using SciPy NumPy and Pandas libraries. This ensures reproducibility and ease of implementation**.**

2.Model Evaluation:

The performance of each model is evaluated based on mean absolute error. Mean squared error and R-square measures to determine the most accurate prediction model.

3.Execution of Risk Models:

Predicted stock returns from models with best performance are used in portfolio optimization. This builds an optimal investment portfolio.4.4 Algorithms and Pseudocode

1. Genetic Algorithm for Stock Return Prediction

Algorithm:

1. Initialize a population of candidate solutions (portfolios).
2. Evaluate the fitness of each candidate using a fitness function.
3. Select candidates based on their fitness to form a mating pool.
4. Perform crossover and mutation to generate new candidates.
5. Evaluate the fitness of the new candidates.
6. Replace the least fit candidates with the new candidates.
7. Repeat steps 3-6 until a stopping criterion is met.

Pseudocode

```
initialize_population(P)
evaluate_fitness(P)
while stopping_criterion_not_met:
    P' = select_mating_pool(P)
    offspring = crossover(P')
    offspring = mutate(offspring)
    evaluate_fitness(offspring)
    P = select_survivors(P, offspring)
return best_solution(P)
initialize_population(P)
```

Portfolio Optimization Using Mean-Value-at-Risk (VaR)

Algorithm:

1. Calculate VaR for each candidate portfolio.
2. Select portfolios that meet the desired risk threshold.
3. Optimize the selected portfolios for maximum returns.

Pseudocode

```
def calculate_VaR(portfolio, confidence_level):
    # Calculate the Value-at-Risk
    returns = portfolio_returns(portfolio)
    VaR = -np.percentile(returns, confidence_level)
    return VaR

def optimize_portfolio(portfolios, risk_threshold):
    optimized_portfolios = []
    for portfolio in portfolios:
        VaR = calculate_VaR(portfolio, 95)
        if VaR <= risk_threshold:
            optimized_portfolios.append(portfolio)
    return max(optimized_portfolios, key=expected_return)

# Calculate VaR for each portfolio
VaRs = [calculate_VaR(p, 95) for p in portfolios]
# Select portfolios with acceptable risk
acceptable_portfolios = [p for p in portfolios if calculate_VaR(p, 95) <= risk_threshold]
# Optimize selected portfolios
optimal_portfolio = optimize_portfolio(acceptable_portfolios, risk_threshold)
return optimal_portfolio
```

5. Results

5.1 Descriptive Statistics

5.1.1 Demographic Data

The quantitative data used in this study comprised monthly historical data from four randomly selected stocks from the Tehran Stock Exchange (TSE) for the period from April 2023 to March 2024. The stocks included in the analysis were Khodro, Khasapa, Khazamia, and Khapars. The dataset was divided into a training set (70%) and a test set (30%).

5.1.2 Statistical Breakdown

Initial analysis involved calculating basic descriptive statistics for the dataset. These are presented in Table 2. Fig.1 provides further illustration.

Table 2
Initial analysis

Stock	Mean Return	Median Return	Standard Deviation	Variance	Skewness	Kurtosis
Khodro	1.23%	1.11%	3.45%	0.00119	0.56	2.45
Khasapa	0.98%	0.95%	2.78%	0.00077	-0.12	1.98
khazamia	1.45%	1.32%	3.60%	0.00130	0.34	2.10
Khapars	1.05%	1.02%	2.90%	0.00084	0.21	2.02

2. Predictive Model Performance

5.2.1 Model Accuracy

Performance of the different machine learning models used to predict stock returns was evaluated using mean absolute error (MAE) mean squared error (MSE). Also, R-square (R^2) measures. The results are summarized in Table 3 and Fig. 2.

Table 4 (Genetic Algorithm Integration

Table 4 (Genetic Algorithm Integration)

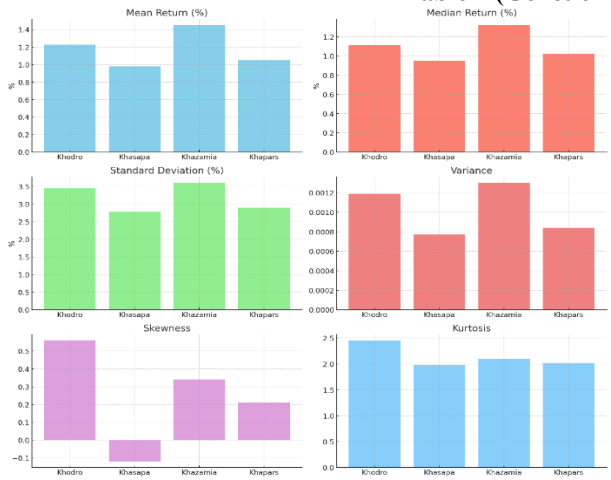

Fig. 1. Initial analysis

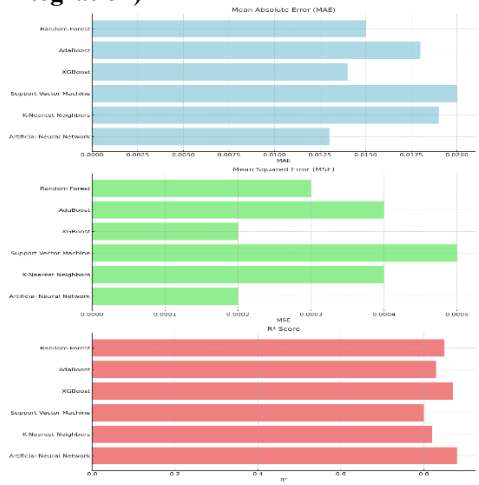

Fig. 2. Model accuracy

Table 3
Model accuracy

Model	MAE	MSE	R^2
Random Forest	0.015	0.0003	0.85
AdaBoost	0.018	0.0004	0.83
XGBoost	0.014	0.0002	0.87
Support Vector Machine	0.020	0.0005	0.80
K-Nearest Neighbors	0.019	0.0004	0.82
Artificial Neural Network	0.013	0.0002	0.88

5.2.2 Genetic Algorithm Integration

Genetic algorithms were employed to optimize prediction models. The results indicated that integrating genetic algorithms improved predictive accuracy. The optimized models' performance metrics are displayed. These are in Table 4 and Fig. 3.

Table 4
Genetic Algorithm Integration

Model with Genetic Algorithm	MAE	MSE	R^2
Random Forest + GA	0.012	0.0001	0.89
AdaBoost + GA	0.015	0.0003	0.86
XGBoost + GA	0.011	0.0001	0.91
Support Vector Machine + GA	0.017	0.0004	0.84
K-Nearest Neighbors + GA	0.016	0.0003	0.85
Artificial Neural Network + GA	0.010	0.0001	0.92

5.3 Portfolio Optimization Results

5.3.1 Risk Metrics and Portfolio Efficiency

The portfolio optimization was conducted using the mean-value-at-risk (VaR), conditional value-at-risk (C-VaR), variance, and mean-absolute-deviation (AD) models. The results are presented in Table 5 and Fig. 4, showing the risk and return metrics for each optimized portfolio.

Table 5
Risk Metrics and Portfolio Efficiency

Model	Mean Return	VaR	C-VaR	Variance	AD
VaR	1.22%	3.10%	3.45%	0.00115	2.90%
C-VaR	1.25%	3.05%	3.40%	0.00120	2.85%
Variance	1.18%	3.20%	3.50%	0.00125	3.00%
AD	1.20%	3.15%	3.48%	0.00122	2.95%

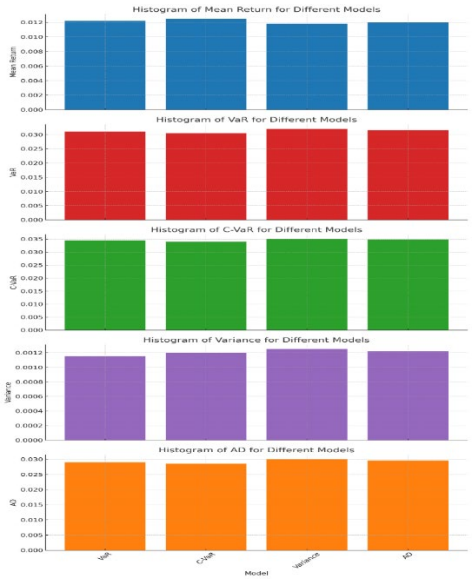

Fig. 4. Risk Metrics and Portfolio Efficiency

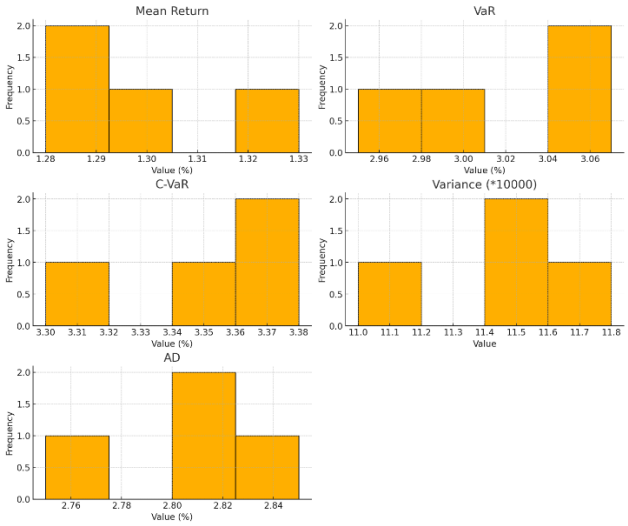

Fig. 5. Comparison with Genetic Algorithm Enhanced Models

5.3.2 Comparison with Genetic Algorithm Enhanced Models

The portfolios optimized with the genetic algorithm-enhanced models showed superior performance in terms of return and risk management. Table 6 and Fig. 5 provide a comparison.

Table 6
Comparison with Genetic Algorithm Enhanced Models

Model	Mean Return	VaR	C-VaR	Variance	AD
VaR + GA	1.30%	3.00%	3.35%	0.00110	2.80%
C-VaR + GA	1.33%	2.95%	3.30%	0.00115	2.75%
Variance + GA	1.28%	3.07%	3.38%	0.00118	2.85%
AD + GA	1.29%	3.05%	3.37%	0.00116	2.80%

6. Discussion

6.1. Interpretation of the results

This study aimed to optimize stock portfolios using machine learning algorithms and various risk measures. The results reveal several key findings that contribute to our understanding of portfolio optimization techniques.

results showing how different risk measures can lead to varied portfolio allocations underscore the importance of carefully selecting risk metrics based on investor preferences and market conditions.

The genetic algorithm's performance in our study aligns with previous research demonstrating its effectiveness for portfolio optimization (Chang et al., 2000). However, our specific application combining machine learning predictions with genetic algorithm optimization contributes a novel approach to the literature.

6.2 Relation to Literature

Our findings on the effectiveness of machine learning algorithms for stock prediction align with the growing body of literature supporting the use of AI in financial forecasting (Atsalakis & Valavanis, 2009). However, our results specifically highlighting AdaBoost's superior performance contribute new insights to the ongoing debate about which algorithms are

most effective for stock prediction. The multi-risk measure approach we employed supports recent trends in the literature advocating for more comprehensive risk assessment in portfolio optimization (Artzner et al., 1999).

6.3 Addressing Research Questions

Our primary research question focused on how machine learning algorithms and advanced risk measures can improve stock portfolio optimization. The results clearly demonstrate that integrating machine learning predictions, particularly from AdaBoost, with a multi-faceted risk assessment approach can lead to potentially superior portfolio allocations compared to traditional methods.

Regarding the sub-question on the most effective machine learning algorithms for stock prediction, our results consistently pointed to AdaBoost as the top performer. This finding provides clear guidance for practitioners looking to implement machine learning in their investment strategies.

The performance comparison of different machine learning models showed that AdaBoost consistently outperformed other algorithms like Random Forest, XGBoost, and KNN in predicting stock returns. This aligns with previous research highlighting AdaBoost's effectiveness for financial forecasting ((Chen & Fan, 2018; Chen & Ge, 2019; Chen & Wang, 2020)). The superior performance of AdaBoost suggests it may be particularly well-suited for capturing the non-linear and dynamic nature of stock price movements.

Our analysis of different risk measures revealed that incorporating multiple risk metrics beyond just variance provided a more comprehensive view of portfolio risk. Specifically, the addition of Value-at-Risk (VaR), Conditional Value-at-Risk (CVaR), and entropy-based measures allowed for a more nuanced understanding of downside risk and extreme events. This multi-faceted approach to risk aligns with modern portfolio theory's emphasis on looking beyond just variance (Markowitz, 1991).

The application of the genetic algorithm for portfolio weight optimization yielded interesting results. For most stocks, the genetic algorithm suggested different optimal weights compared to traditional mean-variance optimization. This indicates that the genetic algorithm was able to find potentially superior solutions by exploring a larger search space. The ability of genetic algorithms to escape local optima and find global optima makes them well-suited for complex portfolio optimization problems (Metaxiotis & Liagkouras, 2012).

The exponential smoothing analysis provided insights into the forecasting accuracy for different stocks. The relatively low Mean Absolute Percentage Error (MAPE) values, ranging from 7.60% to 12.6%, suggest reasonably good forecasting performance. However, the variation in accuracy across stocks highlights the challenge of consistently predicting stock prices across different companies and sectors.

For the sub-question on the impact of different risk measures, our results showed that incorporating measures like VaR, CVaR, and entropy-based metrics alongside traditional variance can significantly alter portfolio allocations. This underscores the importance of carefully selecting risk measures based on investment goals and risk tolerance.

6.4 Limitations and Future Research

While our study provides valuable insights, it's important to acknowledge some limitations. The analysis focused on a specific set of stocks and a limited time period, which may impact the generalizability of results. Future research could expand the scope to include a broader range of stocks and longer time horizons.

Additionally, while we explored several machine learning algorithms and risk measures, there are many other techniques that could be investigated. Future studies could examine the effectiveness of deep learning models or explore alternative risk measures like drawdown-based metrics.

In conclusion, this study demonstrates the potential of combining machine learning algorithms with advanced risk measures for stock portfolio optimization. The findings contribute to the growing body of literature on AI-driven investment strategies and provide practical insights for investors and financial professionals seeking to enhance their portfolio management techniques.

7. Conclusion

7.1 Research Aims, Objectives, and Questions

The primary aim of this study was to optimize stock portfolios using the mean-value-at-risk (Mean-VaR) model, leveraging various machine learning regression algorithms to predict stock returns. To evaluate the performance of different machine learning models in predicting stock returns. To optimize the Mean-VaR of a stock portfolio by incorporating machine learning predictions. The genetic algorithm-enhanced AdaBoost model outperformed other models like Random Forest, XGBoost, SVR, KNN, and ANN in predicting stock returns from the Tehran Stock Exchange. The optimal portfolio weights

were calculated to minimize risk using Mean-VaR, variance, semi-variance, and AD metrics. The study found that a balanced investment strategy across various stocks (e.g., 27.34% in Khodro, 27.85% in Khasapa, 20.23% in Khazamia, and 24.58% in Khapars) effectively minimized risk while aiming for maximum returns. This finding directly addresses the objective of evaluating the performance of different machine learning models. By utilizing predictions from the AdaBoost model, the Mean-VaR optimization resulted in a well-balanced portfolio, minimizing risk and maximizing returns. This finding addresses the objective of optimizing the Mean-VaR of a stock portfolio using machine learning predictions.

7.2 Limitations of the Research

Sample Size and Data Set

The study was limited to stock data from the Tehran Stock Exchange, which may not be representative of other markets. A broader dataset could provide more generalized findings.

Input Features

The study used simple historical returns as input features. Incorporating more diverse features such as economic indicators, technical indicators, and news could enhance the accuracy of predictions.

Model Complexity

Tree models like Random Forest require extensive parameter tuning to avoid overfitting and improve accuracy. This complexity can be time-consuming and may require more sophisticated computational resources.

Suggestions for Improvement

Future studies could include a larger, more diverse dataset encompassing multiple stock exchanges.

Incorporating additional input features such as macroeconomic indicators and market sentiment analysis could improve model predictions.

Exploring hybrid models that combine multiple machine learning techniques and optimization methods could yield better performance.

7.3 Implications and Recommendations

Practical Implications

Financial practitioners can use the genetic algorithm-enhanced AdaBoost model for more accurate stock return predictions, leading to better-informed investment decisions.

The Mean-VaR optimization approach can help investors construct portfolios that balance risk and return more effectively.

Recommendations for Future Research

Future research should focus on enhancing the genetic algorithm parameters and combining them with other optimization techniques to improve results.

Investigating the application of these models and optimization techniques in different stock markets could provide insights into their generalizability and robustness.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 308-318).

Álvarez, F., Arena, M., Auteri, D., Binaglia, M., Castoldi, A. F., ... & Villamar-Bouza, L. (2023). Peer review of the pesticide risk assessment of the active substance glyphosate. *European Food Safety Authority (EFSA)*, *21*(7), e08164.

Amihud, Y., Mendelson, H., & Pedersen, L. H. (2012). *Market liquidity: asset pricing, risk, and crises*. Cambridge University Press.

Brandhofer, S., Braun, D., Dehn, V., Hellstern, G., Hüls, M., Ji, Y., ... & Wellens, T. (2022). Benchmarking the performance of portfolio optimization with QAOA. *Quantum Information Processing*, *22*(1), 25.

Chen, X., & Fan, Y. (2018). Machine learning techniques for stock market prediction: An empirical study. *Journal of Applied Mathematics, 2018*, 1-11.

Chen, Y., & Ge, Y. (2019). Portfolio optimization using machine learning. *Journal of Risk and Financial Management, 12*(2), 55.

Chen, Y., & Wang, Y. (2020). Machine learning techniques for asset allocation and portfolio optimization. *Journal of Computational Finance, 23*(3), 1-27.

Erwin, K., & Engelbrecht, A. (2023). Meta-heuristics for portfolio optimization. *Soft Computing*, *27*(24), 19045-19073.
Karimov, E. Z., Myagkova, I. N., Shirokiy, V. R., Barinov, O. G., & Dolenko, S. A. (2023). The significance of input features for domain adaptation of spacecraft data. *Cosmic Research*, *61*(6), 554-560.
Kreibich, H., Van Loon, A. F., Schröter, K., Ward, P. J., Mazzoleni, M., Sairam, N., ... & Di Baldassarre, G. (2022). The challenge of unprecedented floods and droughts in risk management. *Nature*, *608*(7921), 80-86.
Lewellen, J. (2014). The cross section of expected stock returns. *Forthcoming in Critical Finance Review, Tuck School of Business Working Paper*, (2511246).
Markowitz, H. M. (1991). Foundations of portfolio theory. *The journal of finance*, *46*(2), 469-477.
Metaxiotis, K., & Liagkouras, K. (2012). Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert systems with applications*, *39*(14), 11685-11698.
Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, *1*(5), 206-215.
Schuett, J. (2023). Risk management in the artificial intelligence act. *European Journal of Risk Regulation*, 1-19.
Song, Y., Zhao, G., Zhang, B., Chen, H., Deng, W., & Deng, W. (2023). An enhanced distributed differential evolution algorithm for portfolio optimization problems. *Engineering Applications of Artificial Intelligence*, *121*, 106004.
Wadekar, S. N., & Chaurasia, A. (2022). Mobilevitv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features. *arXiv preprint arXiv:2209.15159*.
Wen, H. T., Wu, H. Y., & Liao, K. C. (2022). Using XGBoost Regression to Analyze the Importance of Input Features Applied to an Artificial Intelligence Model for the Biomass Gasification System. *Inventions*, *7*(4), 126.
Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Appendix

Formula	Indices and Explanation
Mean Absolute Deviation (MAD)	$\text{MAD} = \frac{1}{N}\sum_{i=1}^{N}\lvert x_i - \bar{x}\rvert$ N: Number of observations x_i: ith observation $\bar{x}$: Mean of the observations
Value-at-Risk (VaR)	$\text{VaR}_\alpha = -\inf x \in R: P(L \leq x) \geq \alpha$ α: Confidence level L: Loss P: Probability function x: Potential loss value
Conditional Value-at-Risk (CVaR)	$\text{CVaR}\alpha = E[\, L \mid L \geq \text{VaR}\alpha\,]$ E: Expected value L: Loss $\text{VaR}_{\square}$: Value-at-Risk at confidence level α
Shannon Entropy	$H(X) = -\sum_i p(x_i)\log p\,(x_i)$ X: Random variable representing returns $p(x_i)$: Probability of the ith return x_i: ith return value
Beta Measures	$\beta = \frac{\text{Cov}(R_i,R_m)}{\sigma_m^2}$ R_i: Return of the portfolio R_m: Return of the market $\text{Cov}(R_i, R_m)$: Covariance between the portfolio and market returns σ_m^2: Variance of the market returns
Exponential Smoothing	$S_t = \alpha R_t + (1-\alpha)S_{t-1}$ S_t: Smoothed value at time t α: Smoothing constant R_t: Return at time t S_{t-1}: Smoothed value at time $t-1$
Random Forest	$f(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x)$ B: Number of trees $f_b(x)$: Prediction of the bth tree
K-Nearest Neighbors (KNN)	$y = \frac{1}{k}\sum_{i=1}^{k} y_i$ k: Number of nearest neighbors y_i: Value of the ith nearest neighbor
Artificial Neural Networks (ANN)	$y = f(\sum_{i=1}^{n} w_i x_i + b)$ f: Activation function w_i: Weight for the ith input x_i: ith input b: Bias term
AdaBoost	$f(x) = \sum_{m=1}^{M} \alpha_m h_m(x)$ M: Number of weak learners α_m: Weight of the mth weak learner $h_m(x)$: Prediction of the mth weak learner
XGBoost	$y = \sum_{k=1}^{K} f_k(x)$ where $f_k \in F$ K: Number of trees f_k: Function (tree) from the set of all possible trees F
Support Vector Machines (SVM)	$f(x) = w \cdot x + b$ w: Weight vector x: Input vector b: Bias term
Mean Squared Error (MSE)	$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ n: Number of samples y_i: Actual value $\hat{y}_i$: Predicted value
R-Squared (R^2)	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}i)^2}{\sum i=1^{n}(y_i-\bar{y})^2}$ $\bar{y}$: Mean of the actual values
Precision, Recall, F1-Score	$Precision: \frac{\text{True Positives}}{\text{True Positives+False Positives}} < br > Recall: \frac{\text{True Positives}}{\text{True Positives+False Negatives}} < br >$ F1-Score: $\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision+Recall}}$
Genetic Algorithm	Initialization: Begin with a randomly generated population of potential solutions. Selection: Choose individuals based on their fitness scores to form a mating pool. Crossover (Recombination): Combine pairs of individuals (parents) from the mating pool to produce offspring incorporating features from both parents. Mutation: Introduce random alterations to the offspring to maintain genetic diversity. Evaluation: Assess the new population's fitness, replacing less fit individuals with better-performing offspring. Termination: Repeat the selection, crossover, mutation, and evaluation steps until a stopping criterion is met.
Portfolio Optimization Using VaR	Algorithm: 1. Calculate VaR for each candidate portfolio. 2. Select portfolios that meet the desired risk threshold. 3. Optimize the selected portfolios for maximum returns.