Intelligent Systems with Applications 22 (2024) 200348

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

RPS: Portfolio asset selection using graph based representation learning

MohammadAmin Fazli *, Parsa Alian, Ali Owfi, Erfan Loghmani

Department of Computer Engineering, Sharif University of Technology, Azadi St., Tehran, 1458889694, Tehran, Iran

ARTICLE INFO

Keywords:
Portfolio optimization
Portfolio selection
Representation learning
Graph representation learning

ABSTRACT

Portfolio optimization is one of the essential fields of focus in finance. There has been an increasing demand for novel computational methods in this area to compute portfolios with better returns and lower risks in recent years. We present a novel computational method called Representation Portfolio Selection by redefining the distance matrix of financial assets using Representation Learning and Clustering algorithms for portfolio selection to increase diversification. RPS proposes a heuristic for getting closer to the optimal subset of assets. Using empirical results in this paper, we demonstrate that widely used portfolio optimization algorithms, such as Mean-Variance Optimization, Critical Line Algorithm, and Hierarchical Risk Parity can benefit from our asset subset selection.

1. Introduction

Deciding how and where to invest money is one of the main challenges that anyone with savings faces. The complexity of finding an answer to this challenge has increased as the options for investment have grown over time, expanding from traditional assets like gold and land to a myriad of financial instruments such as stocks, currencies, and cryptocurrencies. Moreover, investing money becomes even more crucial when dealing with substantial amounts, as is the case for financial institutions. Any wrong decision or fluctuation in the price of the invested asset can result in considerable losses. In this situation, individuals make investment decisions based on various parameters such as their knowledge, beliefs, and future predictions.

Two crucial concepts help investors diversify their assets. The first is constructing an initial portfolio, which is a collection of financial assets held by an individual. The second is managing the portfolio afterward. While these two concepts are interlinked, the dynamics of assets and trading constraints in markets introduce more complexities than the former, making them two different problems to solve. This paper will focus on the first problem, portfolio construction, and defer the analysis of the complexities of portfolio management to future studies.

Studies on the portfolio optimization problem have a long history. One of the earliest and most famous theories that studied portfolio construction was Markowitz's Portfolio Theory (Markowitz, 1952, 1959, 1987), which laid the foundation for modern portfolio theory. In their study, they used the covariance matrix of financial assets to define the problem as a quadratic programming problem. The success of Markowitz's theory led to the development of many other methods in this field, each with its own unique ideas. However, most portfolio construction and optimization methods have retained the core idea of Markowitz's theory: the lower the correlation among a portfolio's assets, the lower the risk. Naturally, many papers began to study the shortcomings of Markowitz's methods and proposed solutions to address these shortcomings (Elton et al., 1995, King, 1993, Konno & Yamazaki, 1991, Mills, 1997, Mitra et al., 2003, Rockafellar et al., 2000).

One of the Markowitz issues that we mitigate in this paper is the cardinality constraint problem. Markowitz's method outputs the fraction of money to invest on each asset, with the assets and price history as the input. There are thousands of various assets available for investment, which can lead to hardships since managing such portfolios can be complicated. Moreover, it can be shown that limiting assets to a maximum count is a form of the Knapsack problem, which is computationally NP-hard (Garey & Johnson, 1979). Different methods use different types of heuristics to solve this issue. For example, Crama and Schyns (2003) and Maringer (2005) used simulated annealing, while others use clustering methods like k-means and hierarchical clustering (Lemieux et al., 2014, Raffinot, 2017, León et al., 2017). A whole family of other solutions formed based on Mantegna's Minimum Spanning Tree method (Mantegna & Stanley, 1993, Mantegna, 1999). Some other methods try to reformulate the problem and solve it mathematically, such as Cesarone et al. (2013).

With the growing influence of computational and machine learning methods on financial markets, portfolio construction can be seen as an intermediary field from another perspective. A variety of computational

* Corresponding author.
E-mail address: fazli@sharif.edu (M. Fazli).

https://doi.org/10.1016/j.iswa.2024.200348
Received 1 November 2023; Received in revised form 5 February 2024; Accepted 27 February 2024
Available online 4 March 2024
2667-3053/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

methods have started to rise. A core characteristic of the asset markets, in general, is their complexity. Having a robust way to deal with this complexity will be useful to solve the portfolio optimization problem. One of the machine learning-based methods that can be used to do so is Representation learning, which is a learning method that embeds the entities in the problem in a low-dimensional feature space. By reducing the dimensionality of the data while preserving its essential characteristics, representation learning simplifies the computation. A family of representation learning methods is based on generating graph representation vectors for the data, a paradigm known as graph representation learning. Graph representation learning focuses on transforming nodes and edges in a graph into meaningful numerical representations, referred to as embeddings. These embeddings encode both the intrinsic properties of individual nodes and the relational information they share within the graph's context. By capturing such structural and semantic knowledge, graph representation learning facilitates more efficient and effective analysis of complex data.

In this paper, we propose a new portfolio construction method with the aim of diversification based on Node2Vec Grover and Leskovec (2016), a graph representation learning algorithm. We create a graph, which acts as a representation of our data, based on similarity of the given assets using the correlation matrix of assets. Then, we leverage these representations in a two-phased portfolio optimization setting. First, we select a subset of assets and then weigh the obtained assets. While many portfolio optimization methods do not select the assets explicitly before weighing them, we focus on portfolio selection in this paper. We will indicate that better portfolios with higher returns and less risk can be achieved by separating the portfolio asset selection phase from the portfolio weighting phase. Furthermore, we state that doing so would help us to overcome multiple issues. Firstly, it eliminates the covariance estimation inaccuracy which exists in previous portfolio optimization methods. Secondly, it resolves portfolio optimization algorithms such as Mean-Variance Optimization (MVO) (Erlich et al., 2010), which would have convergence problems if given an extensive benchmark of assets by pre-selecting a heuristically optimal subset of assets.

The paper is organized in the following structure. Section 2 provides an in-depth description of the proposed method. Section 3 discusses the baseline methods, metrics, and the datasets that are used for. The results and discussion are provided in section 4. Lastly, section 5 concludes the paper.

1.1. Related works and approaches

After the initial widespread success of Markowitz's technique, lots of other approaches emerged in this area to enhance portfolio construction by modifying elements of the Markowitz technique or utilizing different strategies. Some attempts tried to improve the covariance estimation mechanism (Ledoit & Wolf, 2004, Wong et al., 2003). Other methods have considered higher-order moments to capture the relationships between assets better (Maringer & Parpas, 2009, Khan et al., 2020).

Instead of improving portfolio optimization methods directly, some methods approached the problem by first selecting a subset of assets and then weighting them to construct the portfolio. Mantegna (1999) was one of the works that did so by using graphs to suggest a subset of assets. By finding an MST of the weighted graph, the method could find the hierarchical structure between stocks. Moreover, other combinatorial optimization methods on graphs are used to identify portfolios, as in Boginski et al. (2014b), in which finding relaxed cliques in the network helps identify similar stocks. Marti et al. (2021) has a trough review of the optimization methods using network structure in financial markets.

Clustering techniques have also found their way into financial studies. In De Prado (2016), authors use a hierarchical clustering approach to cluster stocks based on distances of their corresponding rows in the covariance matrix. Also, Kumari et al. (2019) uses a k-means clustering approach to identify stock groups with the same characteristics.

Recently, machine learning methods have found their ways in financial market studies (Henrique et al., 2019). Portfolio optimization was not an exception for this trend. Some methods first predict each asset changes in the future and use the results for portfolio construction (Ta et al., 2018, Chen et al., 2021, Ma et al., 2021). Other methods have also been used to directly construct the portfolios by training on historical data, such as Reinforcement Learning (Yu et al., 2019), and Deep Learning (Zhang et al., 2020).

Some methods in this area leverage optimization methods to overcome computational and estimation problems of portfolio optimization methods. Still and Kondor (2010) uses a regularization method to construct weights that are stable and robust to fluctuations. Perrin and Roncalli (2020) investigates optimization methods to find a method that could be practically used for real-world problems with lots of assets. Some other studies use prediction models first to predict each asset's future dynamics and then use the predictions to find optimal assets. Ta et al. (2018), Chen et al. (2021), Ma et al. (2021) investigate a range of machine learning methods from simpler methods (Linear Regression & Support Vector Regression) to more complex methods (XGBoost & Long Short-Term Memory). Reinforcement Learning (RL) methods have also been studied, in which an agent learns how to construct appropriate portfolios by behaving in the environment (Yu et al., 2019).

Due to the recent advances in deep learning methods, methods that try to use deep embeddings for stocks are also developed. For instance, Du and Tanaka-Ishii (2020) uses text data like news articles related to each stock to design an embedding for stocks, then using clustering in the embedding space, the method can identify groups of stocks that had similar price behavior as well as similar news. On the other hand, Hu et al. (2018) uses candlestick images and image neural network architectures to reach a proper representation for each stock. While these representation learning methods use text and image data, to the best of our knowledge, no previous method uses the graph of relationships between stocks to derive embeddings for stocks.

We use two recent surveys that cover different modeling and techniques to position our work in the broad range of methods in the financial market domain. The first one is Marti et al. (2021), which reviews the uses of machine learning methods in different areas of asset management, like price forecasting and portfolio management. Among more than twenty portfolio management papers that the authors review, only three use graph modeling of asset prediction. All three papers use the Hierarchical Risk Parity (HRP) method, which we use as a baseline and show our method's superiority. The second survey, Saha et al. (2022), focuses on graph-based approaches in stock market analysis. The survey discusses five main types of stock market graph clustering methods: hierarchical clustering, role-based clustering, infomap, directed bubble hierarchical tree (DBHT), and spectral clustering. However, representation-learning-based methods do not place in either of these types showing the novelty of our approach.

Table 1 serves as a comprehensive overview of the related literature and methodologies employing graph learning for portfolio selection. Among the recent works in this domain, Rezaee et al. (2023) employ community detection on the graph to construct portfolios. Notably, our approach distinguishes itself by transforming the graph into a continuous representation space before conducting community detection, introducing a novel perspective. Furthermore, the weighting scheme within the graph structures differs in our method, contributing to its uniqueness. Another notable method, Li et al. (2022), utilizes the hypergraph attention mechanism to identify group-wise similarities among stocks to alter the portfolio over time. While their technique is commendable, our approach stands apart as it leverages historical data to suggest portfolios, emphasizing a static decision-making process focused on optimizing return-risk combinations of the portfolio's future performance. These distinctions highlight the diversity of strategies within the field

Table 1
Related Literature on Portfolio Selection.

Citation	Description	Review Paper	Uses Graph	Representation Learning
Saha et al. (2022)	Survey of different graph-based approaches in stock market analysis	✓	✓	✓
Gunjan and Bhattacharyya (2023)	Overview of different methods	✓	✗	✗
Erlich et al. (2010)	Mean-Variance Optimization approach	✗	✗	✗
Mantegna (1999)	Uses minimum spanning trees and hierarchies to find the basket	✗	✓	✗
Mantegna (1999)	Uses minimum spanning trees and hierarchies to find the basket	✗	✓	✗
Pfitzinger et al. (2019)	Uses hierarchical risk parity method	✗	✗	✗
Li et al. (2022)	Portfolio selection based on hypergraph embeddings	✗	✓	✓
Rezaee et al. (2023)	Performs community detection to select nodes	✗	✓	✗
This paper	Graph embedding and clustering to find uncorrelated baskets	✗	✓	✓

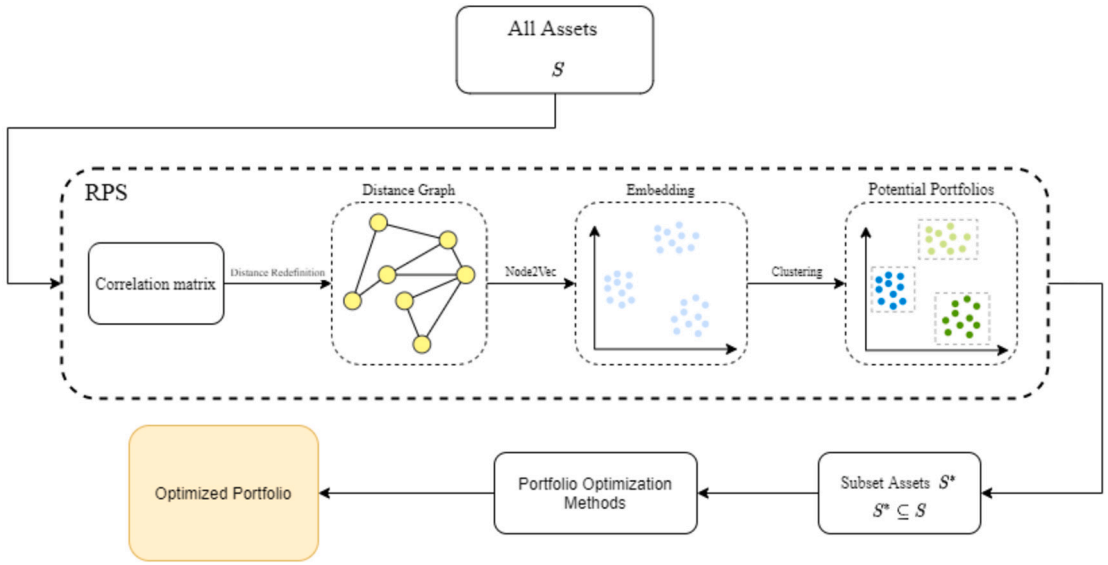

Fig. 1. Stages of RPS algorithm.

and underscore the distinctive contributions of our proposed methodology.

2. Proposed methodology

This paper aims to present a novel utilization of representation learning as a diversification heuristic for a portfolio. The main idea is to choose uncorrelated assets for a portfolio to minimize the portfolio variance, which means less risk for the portfolio. To elaborate, take N assets in a portfolio which are pairwise uncorrelated, i.e., $\forall i \neq j, \rho_{ij} = 0$. Then the variance for this portfolio would be

$$\mathrm{Var}\,(P) = \sum_{i=1}^{N} \left(w_i\right)^2 \sigma_i^2$$

And with the simplifying assumption of equal weights for the assets, i.e., $i, w_i = 1/N$, we will have

$$\mathrm{Var}\,(P) = \frac{1}{N^2} \sum_{i=1}^{N} \sigma_i^2 \leq \frac{\sigma_M^2}{N}$$

where

$$\sigma_M = \max\left[\sigma_i : i = 1, \ldots, N\right]$$

As N grows to infinity, $\mathrm{Var}\,(P)$ will lean towards 0. This illustrates that as we increase the number of assets in our portfolio that are uncorrelated with each other, the overall risk of the portfolio decreases. While it's challenging to find completely uncorrelated assets in real-world scenarios, we can apply this concept by selecting assets with relatively low correlation to construct a well-diversified, low-risk portfolio.

In this paper, we propose the RPS method, a machine-learning-based approach that selects a diversified subset of assets based on the correlation of assets. An overview of this method can be seen in Fig. 1. RPS first builds an augmented graph based on the pairwise correlation of all assets to quantify their similarity. Then by applying a representation-learning method on the resultant graph, it reaches a new distance representation for the assets. Finally, a subset of uncorrelated assets is selected based on the distance representation to fulfill the core idea of having diversified assets. The mentioned steps are further explained below in detail.

To use machine learning and deep learning methods on graph-structured data, we must first embed the graph or its nodes in a vector space. There are different types of methods being used for this purpose. Some of them use discrete metrics to define the similarity between nodes like the number of common neighbors Ahmed et al. (2013), Cao et al. (2015). Others use random walks, and the probability of "walking" from one node to another as the similarity of the two nodes Perozzi et al. (2014), Grover and Leskovec (2016). Both approaches are unsupervised and do not require any labels on data. However, if we have some labeled data, we can employ other methods that utilize the labels to find better representations. These methods have been improved by using other properties of graphs Kipf and Welling (2016), Hamilton et al. (2017). Since our data is unlabeled and our initial metric (correlation) is not discrete, we choose to use Node2Vec. Node2Vec is a graph representation learning algorithm that maps the nodes in a graph to an embedding space. This algorithm employs a flexible random walk strat-

Table 2
Datasets Information.

Index	Asset Count	Train Range	Test Range
S&P 500	465	2019-04-01 to 2019-08-01	2019-08-02 to 2019-09-01
Nikkei 225	225	0 to 200	201 to 290
S&P 100	98	0 to 200	201 to 290

egy that can balance between breadth-first and depth-first exploration of the graph, allowing it to capture both local and global graph patterns effectively. We apply Node2Vec on the correlation graph of the stocks that we constructed to reach an embedding for our data, acting as a new similarity metric for the assets, so that ultimately we can select a set of stocks which gives us a diversified portfolio. The resultant embedding spaces are essentially vectors of features that represent the original nodes in the original graph. We used hyperparameters $walk_length = 2$, $number_walks = 50$, $workers = 7$, and $dimensions = 64$ to run Node2Vec.

To build the described graph, we first need to set weights for the edges. We chose Pearson correlation as the basis of the pairwise similarity measure for the assets. We then augment the basic Pearson Correlation such that the more correlated the two assets are, the less the weight of their intermediate edge will be and vice versa. This augmentation is done so that when Node2Vec is applied to the graph, there is more probability of walking towards less correlated assets rather than correlated pairs. In a random walk conducted by Node2Vec at a given node, it will iterate through the edge j with a probability of $\frac{w_j}{\sum_{i=1}^{n} w_i}$, where n is the number of edges at the starting node. Thus, when the weight of an edge between to assets are smaller, there is smaller chance of a random walk passing through their corresponding edge. Furthermore, to ensure that we will not be visiting highly correlated assets for a starting node in a Node2Vec with a walk length of greater than 1, we should also consider these properties in our augmentation function:

- As the corresponding correlation of an edge approaches 0, the edge's weight should approach infinity.
- As the corresponding correlation of an edge approaches 1, the edge's weight should approach 0.

To this end, we used hyperbolic cotangent for our weight adjustment function as it supports all of the mentioned properties. Thus, the augmented redefinition of pairwise similarity between assets, which is also used as the weight of the edges in our graph, is calculated this way:

$$w_{ij} = |\coth(Correlation(i,j))| - \coth(1) \tag{1}$$

As our graph is built, the next step is to run Node2Vec on our graph. For each node, multiple random walks are executed based on the weights of its edges. That is, in each step, the algorithm visits the neighbor j of a starting node with a probability of $\frac{w_j}{\sum_{i=1}^{n} w_i}$, where w_j is the weight of the edge between the starting node and neighbor j, and the algorithm iterates over the graph in this manner l times for each execution, where l is the walk-length of the algorithm. In the end, a new pair-wise distance representation is created based on the nodes visited during the execution of the algorithm for each starting node. Since we built the graph so that the edges with corresponding uncorrelated nodes to have greater weight than correlated ones, we will visit uncorrelated nodes for each node after this algorithm.

After embedding the nodes of the graph, we are able to use different clustering algorithms to reach our final portfolios. We use two different clustering methods. The first clustering algorithms is k-means, and the second is the fuzzy c-means algorithm, which allows each node to be present in more than one subset. Our initial belief was that this relaxation might allow the procedure to achieve better performance. Since our representation of the graph is produced so that the uncorrelated assets are closer to each other, we will reach clusters of relatively uncorrelated assets, which can be interpreted as selected assets for our diversified portfolio.

The last step in the method is to input the resultant portfolios into a portfolio optimization method to calculate the fraction of wealth to be invest in each of them. In this paper, we made use of three different optimization methods: MVO Erlich et al. (2010), Hierarchical Risk Parity (HRP) Pfitzinger et al. (2019), and Critical Line Algorithm (CLA) Niedermayer and Niedermayer (2010).

3. Evaluation

3.1. Datasets

To test our method in an empirical experiment, we chose three datasets from three different stock market indices in different timeframes. Descriptions of the datasets that we used can be observed in Table 2. The S&P 500 data is available in daily resolution The Nikkei 225 and S&P 100 datasets were obtained from Indextrack datasets Beasley et al. (2003). The prices in this dataset are indexed from 0 to 291, which are the weekly prices between March 1992 to September 1997, and the train and test ranges are expressed as an index value in Table 2.

3.2. Metrics

We used two approaches to evaluate the performance of the algorithms used in our paper. We measured the future performances of their output portfolios using several financial metrics and assessed their stability both in time and against noise via computational methods.

The financial ratios which we used to evaluate the future performance of portfolios were as following:

- Correlation: Since our method's primary focus was to minimize the correlation between different assets' price values, we must evaluate how minimization of correlation in train data relates to the correlation of portfolios in the test data. All of the other measures are a byproduct of this value.
- Return: The return of the portfolios was evaluated in the test range.
- Risk: The risk of the portfolios is defined as the standard deviation of the asset returns in a given time range.
- Sharpe Ratio:

$$Sharpe\ Ratio = \frac{R_p - r_f}{\sigma_p} \tag{2}$$

Where R_p is the return of the portfolio, r_f is the risk-free rate of return, and σp is the standard deviation of the portfolio.

- Information Ratio:

$$Information\ Ratio = \frac{R_p - R_b}{\sigma_{R_p - r_b}} \tag{3}$$

Where R_p is return of the portfolio, R_b is return of the benchmark, and $\sigma_{R_p - r_b}$ is the standard deviation of the excess return.

- M2 Measure (Modigliani):

$$M2\ Measure = SR * \sigma_b + r_f \tag{4}$$

Where SR is the Sharpe Ratio, r_f is the risk-free rate of return, and σ_b is the standard deviation of the benchmark.

To evaluate the stability of the algorithms, we took another approach.

Definition 1. *Suppose that the training phase of an algorithm has resulted in k_1 different portfolios, where each is a set of assets. If the model is trained again under different circumstances, the result of the train would be k_2 portfolios. A stability matrix (SM) is defined as a $k_1 \times k_2$ matrix where SM_{ij} is the similarity value between portfolio i of the first train phase and portfolio j of the second train phase.*

The similarity metric we used in this paper was Jaccard Similarity measures, which is defined as below between set A and set B:

$$JaccardSim(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

Where $|S|$ is the size of set S. We use a matrix instead of a list because we cannot determine an injective function between two phases of training, and therefore no direct mapping exists between two sets of portfolios. After forming the stability matrix, we can extract different measures from it. First of all, we calculate the maximum similarity value for each portfolio to find a mapping between phases. One problem in this process is that the count of portfolios can vary between phases. For example, since the Louvain clustering algorithm does not provide input for several clusters, the output cluster count might differ in different training sets. Furthermore, some of the weighting algorithms might not reach a conversion point for a specific portfolio, and therefore the subset would not be present in the training process results. As a result of this problem, the column-wise maximum of the matrix is not necessarily equal to the row-wise maximum. We combine the row-wise and the column-wise maximums before any further inspections to create a symmetric measure from the similarity matrix.

After taking the maximums, we use the average of maximums to compute the stability of the algorithm. As mentioned before, we also use two different stability tests:

- Noise Stability: The stability of the method if a minuscule amount of Gaussian noise is applied to the correlation matrix of the assets.
- Time Stability: The method's stability if the time range of the training dataset is shifted for a small amount.

Note that these stability functions cannot be applied to Random and Simulated Annealing selection methods since they are statistical approaches for the optimization problem and unstable. No two consecutive runs with similar conditions would result in the same portfolio using these methods.

3.3. Benchmark

Minimum spanning tree (MST) and hierarchical clustering. The approach described in Mantegna (1999) uses Kruskal's algorithm to build an MST over the complete graph of the market. The edges' weights are determined by the relation below.

$$d_{ij} = \sqrt{2(1 - \rho_{ij})} \tag{6}$$

where ρ_{ij} is the correlation between asset i and asset j. This approach relaxes the market graph and makes it easier for further operations. The paper itself does not specify a way to reach subsets of assets. We use Louvain Clustering Algorithm (Blondel et al., 2008) to extract smaller subsets of the market for evaluation. The clusters can be given to a weighting method, similar to RPS.

Graph Splex. The Graph Splex (Boginski et al., 2014a) tries to reach a diversified portfolio by creating a clique-like substructure of the market assets. We implemented this method using the pseudo-code described in the body of Boginski et al. (2014a). This subdivision can be weighted using the weight methods.

Simulated annealing. Overall, various hill-climbing algorithms can be used as the solution for portfolio optimization cardinality problems, such as Simulated Annealing and Genetic Algorithms. The method starts with a random set of states for a subset of assets and changes the weights until a stable state is reached (Crama & Schyns, 2003, Maringer, 2005).

Random subsets. Since our goal is to constraint the cardinality of assets, we can use random divisions of the market as a baseline method. In this approach, since random subdivision has access to every market subset, there is a probability that it can reach some of the best baskets. After selecting this subdivision, we can run the optimization methods on the portfolios.

4. Results

4.1. Future performance

To evaluate the performance of RPS, we constructed several portfolios using different methods and compared their performances via the metrics that we described above. Firstly, we built a set of portfolios with a two-phased approach that used RPS for their asset selection in their first phase, and then used one of the portfolio optimization methods CLA Niedermayer and Niedermayer (2010), MVO Erlich et al. (2010), or HRP Pfitzinger et al. (2019). Then we created a set of other portfolios using our benchmark methods. Three out of four of these methods, Mantegna (MTN), Random (RND), and Graph Splex (SPX), can be used in a two-phased fashion like ours. After computing the subsets, we used the same portfolio optimization methods mentioned above. The Simulated Annealing (SA) is the only one-phased benchmark method, and does not use an optimization method to set weights for the selected assets in a portfolio.

Furthermore, RPS and MTN methods result in multiple portfolios. We run RND and SA multiple times to compare all of the benchmarks in a fair comparison, but the SPX approach outputs a consistent and deterministic result for a given market. Lastly, we could not run the SPX method on Nikkei 225 and S&P 500 datasets since it did not converge on our systems in a bounded time.

It should be noted that the number of clusters used in the clustering phase of RPS was set to 12. This hyperparameter was empirically set to this value as it was large enough to provide a good range of portfolios, but wasn't too big to result in some output portfolios having only one or a few assets in them.

The top 10 portfolios in the training range of each method were then picked, and their performances were evaluated in the test range. Portfolios were sorted using Sharpe value to maximize the return while minimizing the risk. The risk-return graph depicting the discussed portfolios can be observed in Fig. 2.

Moreover, the best value of future metrics for each of the methods are available in Tables 3, 4, and 5.

The tables reveal that within the S&P 500 and Nikkei 225 indices, the RPS method attains the optimal return when compared to other methods, concurrently achieving the highest correlation value. Notably, the act of selection in each instance results in a decrease in the correlation value from the original correlation of the assets. Nevertheless, within the S&P 100 database, diverse outcomes in performance emerge. The results on this dataset can be explained by examining the correlation between assets and the total number of assets. The overall average correlation among S&P 100 assets is determined to be 0.418337. Additionally, this dataset encompasses fewer assets in comparison to other datasets, providing an advantageous environment for random subsets to perform commendably. Two distinctive characteristics contribute to the superiority of random selection results in this dataset. Firstly, the high correlation among assets makes it challenging for selection algorithms to identify highly uncorrelated subsets, leading most portfolios to align with the general trend. Secondly, the reduced number of assets results in a smaller pool of possible subsets, allowing random samples to

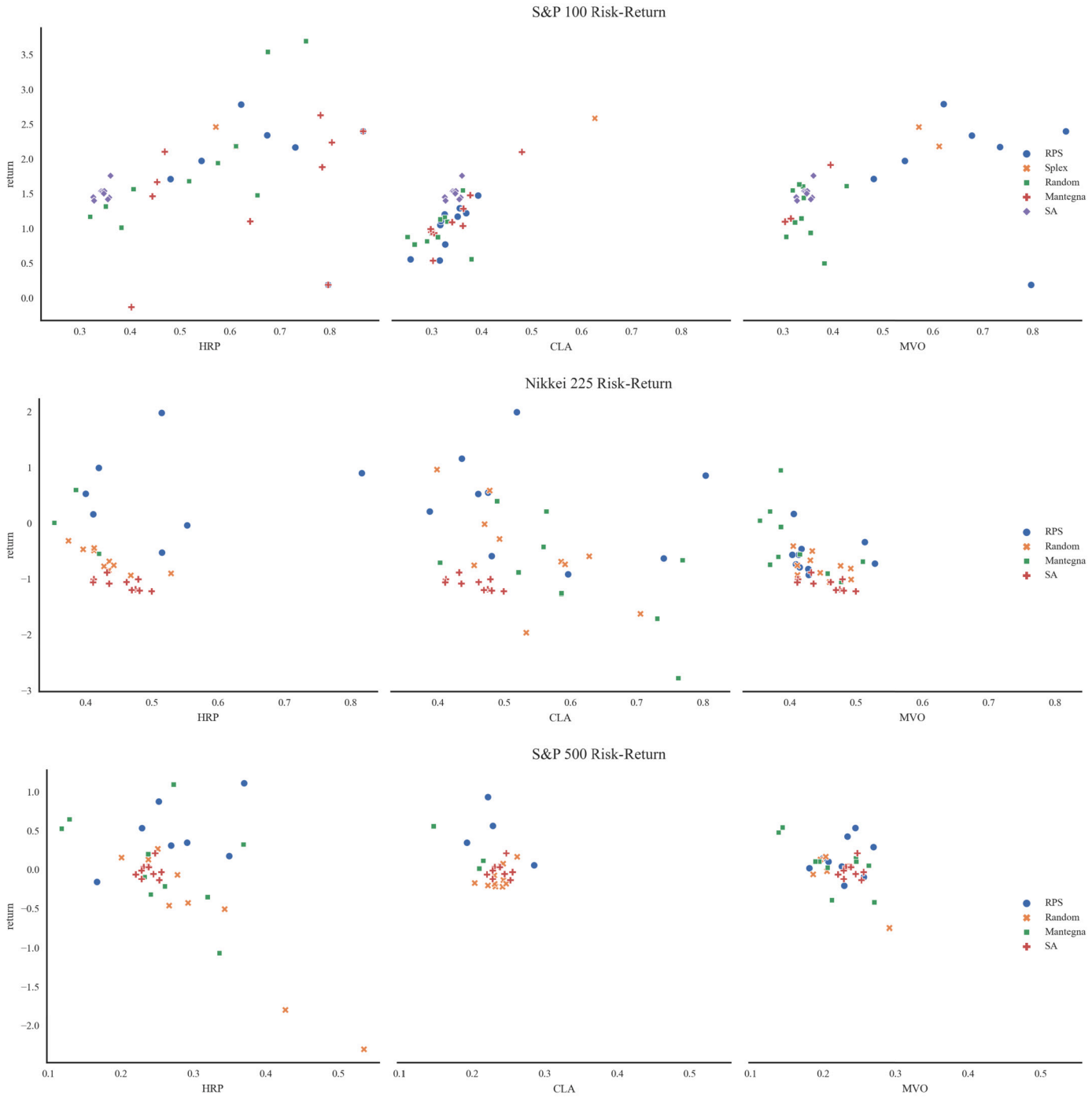

Fig. 2. The efficient-frontier plot for the top 10 portfolios of the training set for RPS+optimization versus benchmark algorithms.

yield instances of high performance. This aligns with the conceptual understanding that a phase of random selection can yield portfolios with relatively high performance due to its capability to traverse any point within the portfolio space.

The return of all portfolios in the markets is graphically represented in Fig. 3. It can be seen that RPS achieves a higher average return. In S&P 100, few of the RND portfolios were able to reach a high return value, and also SPX achieved a higher average return, but as can be seen in the figure, RPS can obtain a higher average performance. Despite RPS's overall better performance compared to the other baseline models, it should be noted that, as RPS is based on the estimation of correlations between the assets, it may be prone to instability and degraded performance when dealing with a limited dataset. An inaccurate estimation of the correlation between the assets may lead to the selection of assets in a portfolio that are not as uncorrelated, and hence the selected portfolio may have higher risk.

Moreover, it should be noted that while RPS and other data-driven and ML-based techniques undoubtedly offer valuable insights for portfolio selection, it's essential to acknowledge potential limitations and associated risks in their practical application. The inherent complexity of financial markets introduces an element of risk. ML models, dependent on historical data patterns, may encounter challenges in adapting to unforeseen market fluctuations, potentially impacting the efficacy of portfolio selection strategies. Therefore, a balanced approach, incorporating risk-aware methodologies and robust privacy measures, is pivotal

Table 3
Future Performance Measures for S&P 500 Dataset.

Method	Correlation	Return	Risk	Sharpe Ratio	Information Ratio	M2
Vanilla CLA	0.162075	0.027747	0.114838	0.164118	0.242098	0.027636
RPS+CLA	0.138791	**1.110586**	0.369897	3.435445	**4.721627**	0.817560
MTN+CLA	0.196280	1.094718	0.369051	**4.948828**	4.654215	**1.173791**
RND+CLA	0.171342	0.267575	0.535209	1.031657	1.140248	0.251739
Vanilla HRP	0.162075	0.011727	**0.095295**	0.216459	0.103681	0.015811
RPS+HRP	**0.008178**	0.535214	0.269456	2.153089	2.277263	0.515710
MTN+HRP	0.196280	0.541221	0.270475	3.698830	2.302783	0.879557
RND+HRP	0.191148	0.166312	0.291356	0.773329	0.710050	0.190932
Vanilla MVO	0.162075	0.027747	0.114838	0.164118	0.242098	0.027636
RPS+MVO	0.138791	0.932556	0.285581	4.163090	3.965298	0.988838
MTN+MVO	0.235134	0.557674	0.215301	3.736161	2.372682	0.888345
RND+MVO	0.174462	0.165436	0.262077	0.597288	0.706327	0.149494
SA	0.167465	0.273648	0.210593	1.117717	1.166047	0.271996

Table 4
Future Performance Measures for Nikkei 225 Dataset.

Method	Correlation	Return	Risk	Sharpe Ratio	Information Ratio	M2
Vanilla CLA	0.450845	0.143052	0.436134	0.261482	0.278425	0.142717
RPS+CLA	**0.094671**	**1.991665**	0.387936	3.818792	**4.599676**	1.665158
MTN+CLA	0.210820	0.395800	0.403417	0.790543	0.920129	0.351768
RND+CLA	0.189425	0.962127	0.398781	2.390352	2.225895	1.045625
Vanilla HRP	0.450845	0.042383	0.422741	0.069599	0.081713	0.044518
RPS+HRP	0.176072	0.165529	0.403649	0.385897	0.389200	0.176268
MTN+HRP	0.210820	0.946481	0.354701	2.427145	2.189820	1.061583
RND+HRP	0.265348	-0.411154	0.405019	-1.037120	-0.940445	-0.440912
Vanilla MVO	0.450845	0.143052	0.436134	0.261482	0.278425	0.142717
RPS+MVO	**0.094671**	1.978367	0.400116	**3.826142**	4.569014	**1.668345**
MTN+MVO	0.210820	0.594692	**0.352680**	1.520505	1.378710	0.668362
RND+MVO	0.105741	-0.315837	0.374047	-0.868171	-0.720674	-0.367636
SA	0.558230	-0.937416	0.420207	-2.070724	-2.153834	-0.889199

Table 5
Future Performance Measures for S&P 100 Dataset.

Method	Correlation	Return	Risk	Sharpe Ratio	Information Ratio	M2
Vanilla CLA	0.418337	1.269428	0.198022	4.348930	4.385291	1.262862
RPS+CLA	0.177503	2.783124	0.481028	4.457300	9.635014	1.294110
MTN+CLA	0.155783	2.630134	0.402634	4.461950	9.104422	1.295450
SPX+CLA	0.870104	2.461136	0.571726	4.289180	8.518312	1.245634
RND+CLA	0.208234	**3.696725**	0.320223	**5.226347**	**12.803521**	**1.515855**
Vanilla HRP	0.418337	1.278512	0.195795	4.449693	4.416796	1.291916
RPS+HRP	0.255930	1.473516	0.257972	3.724968	5.093098	1.082950
MTN+HRP	0.155783	2.100452	0.298370	4.351074	7.267409	1.263480
SPX+HRP	0.870104	2.585765	0.627108	4.109122	8.950544	1.193717
RND+HRP	0.206733	1.548159	**0.252117**	4.243590	5.351972	1.232489
Vanilla MVO	0.418337	1.269428	0.198022	4.348930	4.385291	1.262862
RPS+MVO	0.177503	2.790539	0.481578	4.476027	9.660732	1.299509
MTN+MVO	0.339761	1.914374	0.303960	4.823667	6.622061	1.399747
SPX+MVO	0.870104	2.461136	0.571726	4.289180	8.518312	1.245634
RND+MVO	**0.043606**	1.634323	0.306306	4.889193	5.650800	1.418641
SA	0.217874	1.759462	0.326334	4.851917	6.084802	1.407893

to harnessing the full potential of data-driven and ML methods in portfolio management.

4.2. Computational complexity of the RPS method

The RPS method not only proves effective in optimizing portfolio returns but also significantly reduces the computational complexity, enhancing solvability in comparison to traditional methods. To illustrate, the MVO method necessitates solving a quadratic programming problem, where each step involves $\mathcal{O}(n^3)$ operations, where n is the number of assets. This posits a considerable computational burden. The selection phase in RPS however aids the optimization process by strategically reducing the number of assets for subsequent methods.

When considering the correlation matrix as input, the time complexity of the RPS method can be delineated into three distinct steps: finding

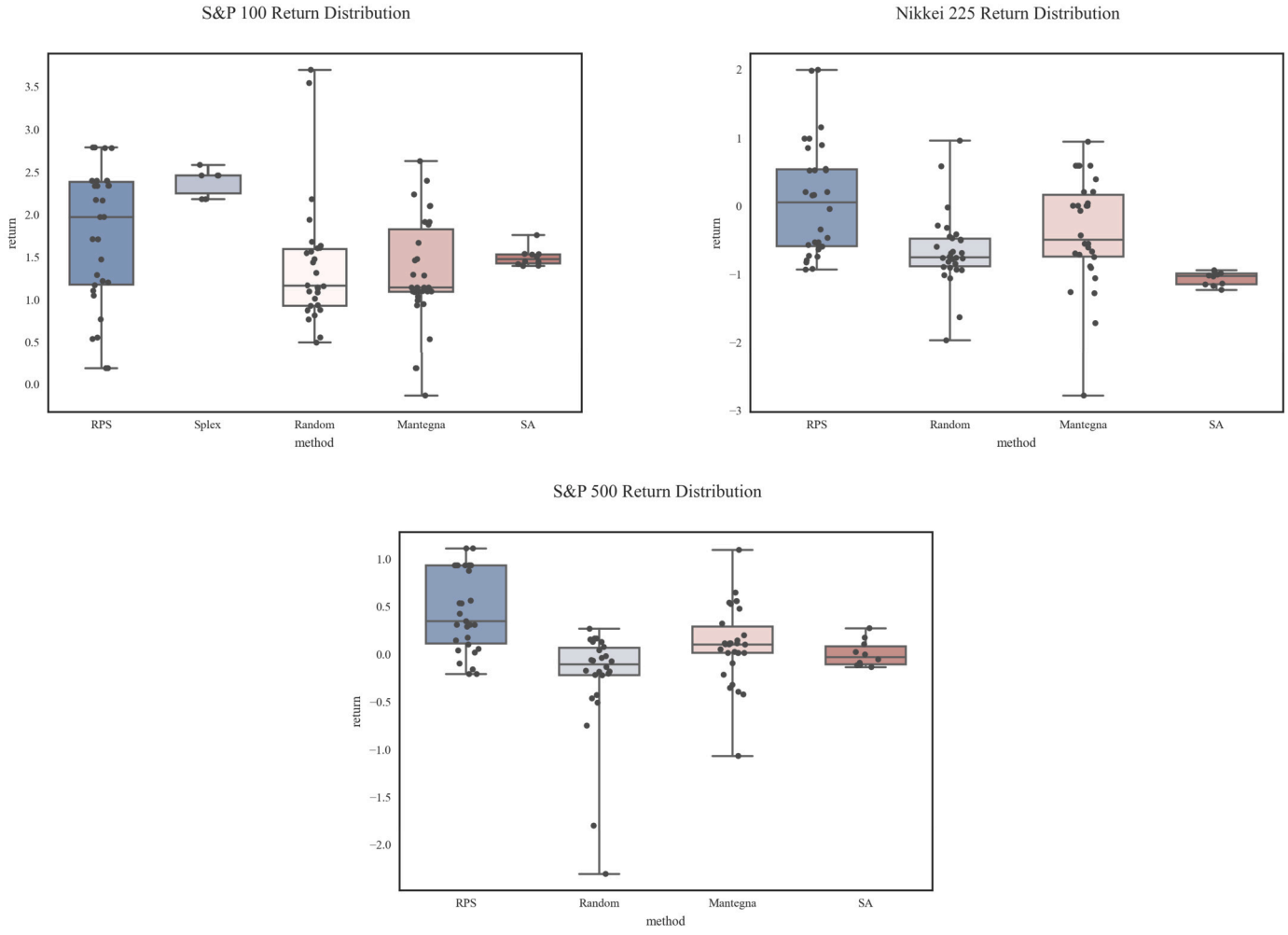

Fig. 3. Return distribution.

the graph embeddings, clustering, and portfolio optimization. The initial step of identifying graph embeddings exhibits a highly scalable time complexity of O(n log(n)) Pimentel et al. (2018), ensuring efficiency in handling datasets of varying sizes. The clustering step employs the k-means algorithm, where each iteration involves $\mathcal{O}(kn)$ vector arithmetic calculations. Subsequently, the average cluster size becomes n/k. This effectively diminishes the downstream portfolio optimization complexity, as the subsequent methods only need to contend with n/k assets on average. This reduction in computational load streamlines the optimization process, rendering it more easily solvable.

The final step involves portfolio optimization based on the clustered subsets. With the reduced set of assets in each cluster, the optimization process becomes more computationally efficient, contributing to the overall effectiveness of the RPS method. Based on this evaluation, the RPS method presents a favorable computational landscape, offering a scalable and efficient approach to address the challenges posed by traditional methods, particularly in scenarios with large datasets.

4.3. Stability

The time and noise stability metrics are shown in Table 6 and Table 7. Gaussian noise with $\mu = 0$ and $\sigma = 0.01$ was applied to the market correlation matrix for noise stability. For time stability, the train time ranges were shifted to 20 data points for each dataset.

In noise stability metrics, MTN was able to outperform RPS and SPX methods. However, the margin of superiority was not significant in the S&P 100 dataset, resulting from the smaller size of the database and the lower number of options available in the selection phase. As the size of the dataset grows, stability drops for all of the methods.

In order to assess the stability of our proposed method under varying subsets of the dataset, we conducted an additional stability analysis using the S&P500 dataset. The approach involved randomly selecting subsets of assets, varying the ratios at 95%, 90%, and 80%. Subsequently, we compared the portfolios generated by each method on these subsets to those derived from the original data. The clustering method was applied to identify 60 clusters, utilizing clusters with non-trivial subsets to measure the similarity between portfolios.

Table 6
Noise Stability.

Method	S&P 100	Nikkei 225	S&P 500
Mantegna	**0.528711**	**0.479277**	**0.427159**
RPS	0.460720	0.373794	0.085135
Splex	0.000000	-	-

Table 7
Time Stability.

Method	S&P 100	Nikkei 225	S&P 500
Mantegna	0.336598	**0.360965**	**0.228190**
RPS	**0.557390**	0.231124	0.119387
Splex	0.400000	-	-

Table 8
Subset Stability on S&P500.

Method	95%	90%	80%
RPS	0.2765 (0.0098)	0.2512 (0.0108)	0.2049 (0.0120)
Mantegna	**0.6866** (0.0180)	**0.5765** (0.0109)	**0.4403** (0.0126)

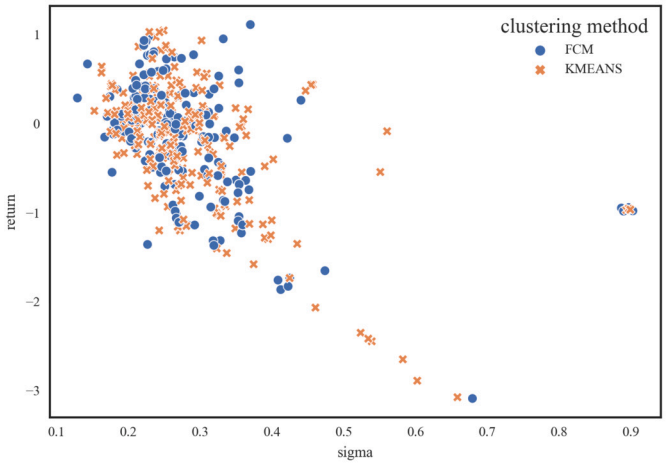

Fig. 4. Risk-Return distribution of RPS on S&P 500 dataset, separated by clustering method.

Table 8 presents the outcomes of this analysis. Each column represents the similarity of portfolios when selecting a specific ratio of assets. Consistent with our previous findings, the MTN method exhibits a higher level of stability. However, it is noteworthy that as the sample size decreases, the similarity measure declines more rapidly for the MTN method.

4.4. The effect of the clustering method

Another issue to analyze is whether the clustering method (k-means or fuzzy c-means) affects outcoming portfolios or not. Fig. 4 depicts the risk-return relation for all different setups of RPS ran in the test range.

As seen in this figure, there are no significant differences between the portfolios using these two clustering methods.

5. Conclusion and future work

This paper introduced an novel portfolio selection method, denoted as RPS (Representation-based Portfolio Selection), leveraging representation learning and graph embedding techniques. As evidenced by our empirical results, the incorporation of RPS into various portfolio weight optimization methods consistently led to enhanced performance compared to the vanilla usage of those methods. Furthermore, our findings demonstrated that portfolios constructed using RPS consistently exhibited superior returns when compared to their counterparts.

This study primarily utilizes the correlation matrix of asset prices to generate graph embeddings for assets. Future research could explore enriching the embeddings of assets by incorporating additional information from diverse sources, such as news and social media. Such enhancements might contribute to a more informative embedding of assets and improve portfolio selection accuracy. Furthermore, an intriguing avenue for future research involves considering ensemble approaches by combining RPS with different selection methods. While RPS excels at integrating correlation information into embeddings, an ensemble approach could capitalize on the strengths of various methods. We leave the exploration of this avenue as a direction for future research.

Another future work could involve the extension of RPS into a dynamic portfolio management algorithm. The focus will be on enhancing the adaptability of RPS by incorporating real-time data analysis, allowing the algorithm to dynamically respond to changes in market conditions. This entails developing a framework that continuously analyzes incoming market data to capture evolving trends in changing market conditions.

Our results demonstrated that RPS, as a portfolio selection method, improves the overall performance when integrated with a portfolio optimization method in a 2-step portfolio optimization, compared to when the portfolio optimization methods are directly applied to the dataset. However, among the tested portfolio optimization methods combined with RPS, we could not find a method that outperforms others in all cases. A future investigation could involve finding characteristics of the dataset that indicate which portfolio optimization method, when combined with RPS, would perform the best.

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.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve grammar. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., & Smola, A. J. (2013). Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 37–48).

Beasley, J., Meade, N., & Chang, T. J. (2003). An evolutionary heuristic for the index tracking problem. *European Journal of Operational Research*, *148*, 621–643. https://doi.org/10.1016/S0377-2217(02)00425-3. https://www.sciencedirect.com/science/article/pii/S0377221702004253.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*, Article P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008.

Boginski, V., Butenko, S., Shirokikh, O., Trukhanov, S., & Gil Lafuente, J. (2014a). A network-based data mining approach to portfolio selection via weighted clique relaxations. *Annals of Operations Research*, *216*. https://doi.org/10.1007/s10479-013-1395-3.

Boginski, V., Butenko, S., Shirokikh, O., Trukhanov, S., & Gil Lafuent, J. G. (2014b). A network-based data mining approach to portfolio selection via weighted clique relaxations. *Annals of Operations Research*, *216*, 23–34.

Cao, S., Lu, W., & Xu, Q. (2015). Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (pp. 891–900).

Cesarone, F., Scozzari, A., & Tardella, F. (2013). A new method for mean-variance portfolio optimization with cardinality constraints. *Annals of Operations Research*, *205*, 213–234.

Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean–variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, *100*, Article 106943.

Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 546–571.

De Prado, M. L. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, *42*, 59–69.

Du, X., & Tanaka-Ishii, K. (2020). Stock embeddings acquired from news articles and price history, and an application to portfolio optimization. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 3353–3363).

Elton, E. J., Gruber, M. J., & Blake, C. R. (1995). Fundamental economic variables, expected returns, and bond fund performance. *The Journal of Finance*, *50*, 1229–1256. http://www.jstor.org/stable/2329350.

Erlich, I., Venayagamoorthy, G. K., & Worawat, N. (2010). A mean-variance optimization algorithm. In *IEEE congress on evolutionary computation* (pp. 1–6). IEEE.

Garey, Michael R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman.

Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864).

Gunjan, A., & Bhattacharyya, S. (2023). A brief review of portfolio optimization techniques. *Artificial Intelligence Review*, *56*, 3847–3886.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. arXiv preprint arXiv:1706.02216.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226–251.

Hu, G., Hu, Y., Yang, K., Yu, Z., Sung, F., Zhang, Z., Xie, F., Liu, J., Robertson, N., Hospedales, T., et al. (2018). Deep stock representation learning: From candlestick charts to investment decisions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2706–2710). IEEE.

Khan, K. I., Naqvi, S. M., Ghafoor, M. M., & Akash, R. S. I. (2020). Sustainable portfolio optimization with higher-order moments of risk. *Sustainability*, *12*, 2006.

King, A. (1993). Asymmetric risk measures and tracking models for portfolio optimization under uncertainty. *Annals of Operations Research*, *45*, 165–177. https://doi.org/10.1007/BF02282047.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Konno, H., & Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management Science*, *37*, 519–531. http://www.jstor.org/stable/2632458.

Kumari, S. K., Kumar, P., Priya, J., Surya, S., & Bhurjee, A. (2019). *Mean-value at risk portfolio selection problem using clustering technique: A case study*. *AIP conference proceedings*. AIP Publishing LLC (p. 020178).

Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, *88*, 365–411.

Lemieux, V., Rahmdel, P. S., Walker, R., Wong, B. W., & Flood, M. (2014). Clustering techniques and their effect on portfolio formation and risk analysis. In *Proceedings of the international workshop on data science for macro-modeling* (pp. 1–6).

León, D., Aragón, A., Sandoval, J., Hernández, G., Arévalo, A., & Niño, J. (2017). Clustering algorithms for risk-adjusted portfolio construction. *Procedia Computer Science*, *108*, 1334–1343.

Li, X., Cui, C., Cao, D., Du, J., & Zhang, C. (2022). Hypergraph-based reinforcement learning for stock portfolio selection. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE (pp. 4028–4032).

Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, *165*, Article 113973.

Mantegna, R. N. (1999). Hierarchical structure in financial markets. *The European Physical Journal B. Condensed Matter and Complex Systems*, *11*, 193–197.

Mantegna, R. N., & Stanley, H. E. (1993). *Introduction to econophysics: Correlations and complexity in finance*. Cambridge University Press.

Maringer, D. (2005). *Advances in computational management science: Vol. 8*. Springer.

Maringer, D., & Parpas, P. (2009). Global optimization of higher order moments in portfolio selection. *Journal of Global Optimization*, *43*, 219–230.

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

Markowitz, H. (1959). *Portfolio selection: Efficient diversification of investments, vol. 16*. *Cowles foundation for research in economics at Yale university, monograph: Vol. 16*.

Markowitz, H. (1987). *Mean-variance analysis in portfolio choice and capital markets*. Oxford: Basil Blackwell.

Marti, G., Nielsen, F., Bińkowski, M., & Donnat, P. (2021). A review of two decades of correlations, hierarchies, networks and clustering in financial markets. In *Progress in information geometry: Theory and applications* (pp. 245–274).

Mills, T. (1997). Stylized facts on the temporal and distributional properties of daily ft-se returns. *Applied Financial Economics*, *7*, 599–604.

Mitra, G., Kyriakis, T., Lucas, C., & Pirbhad, M. (2003). A review of portfolio planning: Models and systems. In *Advances in portfolio construction and implementation* (pp. 1–39).

Niedermayer, A., & Niedermayer, D. (2010). Applying Markowitz's critical line algorithm. In *Handbook of portfolio construction* (pp. 383–400). Springer.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 701–710).

Perrin, S., & Roncalli, T. (2020). Machine learning optimization algorithms & portfolio allocation. In *Machine learning for asset management: New developments and financial applications* (pp. 261–328).

Pfitzinger, J., Katzke, N., et al. (2019). *A constrained hierarchical risk parity algorithm with cluster-based capital allocation*. Stellenbosch University, Department of Economics.

Pimentel, T., Veloso, A., & Ziviani, N. (2018). *Fast node embeddings: Learning ego-centric representations*.

Raffinot, T. (2017). Hierarchical clustering-based asset allocation. *The Journal of Portfolio Management*, *44*, 89–99.

Rezaee, F., Ahmadi, J., & Haratizadeh, S. (2023). Gps: A graph-based approach to portfolio selection. In *2023 28th international computer conference, computer society of Iran (CSICC)*. IEEE (pp. 1–3).

Rockafellar, R. T., Uryasev, S., et al. (2000). Optimization of conditional value-at-risk. *The Journal of Risk*, *2*, 21–41.

Saha, S., Gao, J., & Gerlach, R. (2022). A survey of the application of graph-based approaches in stock market analysis and prediction. *International Journal of Data Science and Analytics*, *14*, 1–15.

Still, S., & Kondor, I. (2010). Regularizing portfolio optimization. *New Journal of Physics*, *12*, Article 075034.

Ta, V. D., Liu, C. M., & Addis, D. (2018). Prediction and portfolio optimization in quantitative trading using machine learning techniques. In *Proceedings of the ninth international symposium on information and communication technology* (pp. 98–105).

Wong, F., Carter, C. K., & Kohn, R. (2003). Efficient estimation of covariance selection models. *Biometrika*, *90*, 809–830.

Yu, P., Lee, J. S., Kulyatin, I., Shi, Z., & Dasgupta, S. (2019). Model-based deep reinforcement learning for dynamic portfolio optimization. arXiv preprint arXiv:1901.08740.

Zhang, Z., Zohren, S., & Roberts, S. (2020). Deep learning for portfolio optimization. *The Journal of Financial Data Science*, *2*, 8–20.