Iranian Journal of Numerical Analysis and Optimization
Vol. 15, No. 2, 2025, pp 600–624
https://doi.org/10.22067/ijnao.2025.89520.1499
https://ijnao.um.ac.ir/

Research Article

Multi-objective portfolio optimization using real coded genetic algorithm based support vector machines

B. Surja, L. Chin and F. Kusnadi*

Abstract

Investors need to grasp how liquidity affects both risk and return in order to optimize their portfolio performance. There are three classes of stocks that accommodate those criteria: Liquid, high-yield, and less-risky. Classifying stocks help investors build portfolios that align with their risk profiles and investment goals, in which the model was constructed using the one-versus-one support vector machines method with a radial basis function kernel.

*Corresponding author

Received 21 August 2024; revised 28 Janurary 2025; accepted 23 February 2025

Bianca Surja
Center for Mathematics and Society, Department of Mathematics, Faculty of Science, Parahyangan Catholic University, Bandung, Indonesia. e-mail: biancasurja78@gmail.com

Liem Chin
Center for Mathematics and Society, Department of Mathematics, Faculty of Science, Parahyangan Catholic University, Bandung, Indonesia. e-mail: chin@unpar.ac.id

Felivia Kusnadi
Center for Mathematics and Society, Department of Mathematics, Faculty of Science, Parahyangan Catholic University, Bandung, Indonesia. e-mail: felivia@unpar.ac.id

How to cite this article
Surja, B., Chin, L. and Kusnadi, F., Multi-objective portfolio optimization using real coded genetic algorithm based support vector machines. *Iran. J. Numer. Anal. Optim.*, 2025; 15(2): 600-624. https://doi.org/10.22067/ijnao.2025.89520.1499

This model was trained using a combination of the Kompas100 index and the Indonesian industrial sectors stocks data. Single optimal portfolios were created using the real coded genetic algorithm based on different sets of objectives: Maximizing short-term and long-term returns, maximizing liquidity, and minimizing risk. In conclusion, portfolios with a balance on all these four investment objectives yielded better results compared to those focused on partial objectives. Furthermore, our proposed method for selecting portfolios of top-performing stocks across all criteria outperformed the approach of choosing top stocks based on a single criterion.

AMS subject classifications (2020): Primary 65K10; Secondary 68T05, 62P05.

Keywords: Genetic algorithm; Liquidity; Multi-objective optimization; One-versus-one support vector machines; Radial basis functions.

1 Introduction

Portfolio optimization has long been a challenging, lucrative, and prominent research topic in investment management. Achieving the right balance between risk and return while maintaining liquidity has consistently been essential for effective portfolio selection [7, 3]. The stock market often overlooked changes in liquidity which can yield monthly returns (0.7-1.2%) in the short term [2, 6]. Santoso et al. [25] also suggested that liquidity can act as an early warning indicator for financial instability of an entity.

By taking all of these factors into account, selecting the top-performing stocks has always been a crucial task in successful portfolio optimization. Contemporary research utilized machine learning and artificial intelligence algorithms. Many researchers had noted that artificial neural network converged slowly. For example, finding the optimal number of neurons in hidden layers was often tiresome, as it involved trial-and-error [24].

Research by Huang [15] employed genetic algorithm (GA) for feature selection and parameter optimization. Subsequently, support vector regression (SVR) predicted and identified stocks with high returns serving as surrogates of actual returns, without considering liquidity. Similarly, Ma, Han, and Wang [20] predicted return of stocks by combining machine learning models, namely random forest (RF) and SVR, and deep learning models, such as LSTM neural network, deep multilayer perceptron, and convolutional neural network, with mean-variance (MV) and omega portfolio optimization models. The result showed that the RF-MV optimization model performed the best. More recently, studies by Chen et al. [5], Guijarro [10], Jimbo et al. [17], and Kessaci [19] employed a GA for effective portfolio optimization method.

Gupta, Mehlawat, and Saxena [12] split stocks into three classes: Liquid, high-yield, and less-risky, by utilizing support vector machine (SVM) whose parameters were optimized by the grid-search method. Stocks in the dataset were first filtered for the highly liquid class, then categorized based on high yield, which indicated high returns. The remaining stocks with lower return and liquidity were classified as the residual category. After categorizing each stock into the three different classes, they selected the top-performing stocks in each class by using GA. The final portfolio consisted of all the portfolios created from the three classes.

The previous research studies have several limitations. First of all, liquidity was not considered along with risk and return. Secondly, the allocated number of stocks in the portfolio was known in advance, while in practice, the number of high-performing stocks was not known beforehand. Third of all, portfolios were assembled from each class and then combined to form the final portfolio. This approach may be less effective because the fixed number of stocks chosen for each portfolio might not represent the top-performing stocks in the overall selection. Lastly, the weights of the multi-objective model were not further explored to create several optimal portfolios which were tailored to different investment goals and risk profiles.

The aim of this paper is to demonstrate an extension of the multi-objective model conducted by Gupta et al. [11] by demonstrating how an optimized portfolio is immediately selected from the stocks dataset using real coded genetic algorithm (RCGA), in which it successfully handled economic problems such as portfolio selection and financial budget allocation [4, 23]. The number of stocks desired by an investor for a portfolio is fixed, and the quantity of stocks in each category is adjusted to fit this requirement. One of the classes may contain fewer stocks than the other two in order to create an optimized portfolio that enhanced return and liquidity while reducing risk. Furthermore, we examined different weights to develop various optimal portfolios to accommodate numerous investment objectives of return, liquidity, and risk.

Our work is explained as follows. Section 2 provides the brief introduction to the underlying methods employed in this paper. Section 3 describes the dataset and methodology design to conduct the research. Section 4 presents the results and analysis of the study, including a comparative analysis with the methodology put forward by Gupta et al. [11]. Finally, in Section 5, the conclusions are presented along with suggestions to improve for further research.

2 Overview

2.1 Support vector machines

The SVM method is a machine learning technique based on statistical learning theory, applying structural risk minimization to minimize model errors in generalizing data patterns [1]. SVM divides data into two classes, +1 and −1, using a decision boundary known as a hyperplane. Data points closest to the hyperplane that influence its orientation are called support vectors, and the distance between the support vectors is known as the margin. The hyperplane is a line in two-dimensional data, while it being a plane in three-dimensional data.

The primary goal of SVM is to find the optimal hyperplane to avoid misclassification. SVM seeks to maximize the margin so that the model has a large distance between the hyperplane and the support vectors. A model that perfectly separates the data without any data points within the margin and without misclassification errors is called a hard margin SVM. The SVM hyperplane optimization problem is

$$
\begin{aligned}
&\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \\
&\text{subject to } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \geq 0, i = 1, 2, \dots, m.
\end{aligned}
$$

This concept is extended to soft margin SVM problems, where data cannot be perfectly separated by a hyperplane. The main objective is to find a hyperplane that minimizes errors as much as possible [9]. To achieve this objective, slack variables, which allow data points to lie within the margin and account for some classification errors, are referred to as margin errors. The penalty, C, is used to control the model's error in classifying the data.

The SVM method was initially designed for binary classification problems, meaning it can only separate data into two classes. In cases where there are more than two classes, an approach to extend SVM is required. In this study, the multiclass SVM method used is one-versus-one (OVO) SVM, because OVO specifically addresses the differences between each class and has shown good results in previous research [14]. Moreover, its effectiveness has been proven in various economic issues such as stock selection and credit assessment [8, 16]. Suppose that a dataset has k classes. Then one-versus-one support vector machines (OVO SVM) will form $\frac{k}{2}(k-1)$ hyperplanes, where each hyperplane is built based on training data from a pair of classes.

Furthermore, for nonlinear data problems, SVM can utilize kernel functions to transform data into higher dimensions and construct non-linear hyperplanes. Some commonly used kernel functions in SVM are polynomial, radial basis function (RBF), and sigmoid. The kernel used in this study is the RBF kernel, as suggested by Yu, Wang, and Lai [28]. This decision is based on the consideration that the sigmoid kernel behaves similarly to the RBF kernel, while the polynomial kernel requires more time in the SVM training phase. SVM models with RBF kernels have also been proven to perform better than other kernels [11].

2.2 Real coded genetic algorithm

GAs are optimization methods inspired by the principles of natural selection and genetics in biological systems, used to find solutions for problems with infinitely many possible solutions [26]. Essentially, GA represent solutions as chromosomes composed of a series of genes, where each gene represents a variable of the proposed solution. The value of each gene is called an allele. In the search for the optimal solution, GAs operate with a set of chromosomes known as a population. The number of chromosomes in a population (*popsize*) is a GA parameter that must be defined before running the algorithm. In this study, we utilize the RCGA, where alleles are real numbers.

To optimize the solution to an objective function $f(x)$ with q variables, RCGA first constructs a population consisting of n chromosomes. Each chromosome consists of q genes, with each allele assigned a random value. The quality of a chromosome in solving the given problem is evaluated using a fitness value, which indicates how well the chromosome addresses the problem. The primary goal of RCGA is to find the chromosome with the highest fitness value [21]. The initial population that is randomly generated, will go through a series of genetic operators to create a new, more optimal generation [13].

The genetic operations include elitism, selection to choose the best chromosomes as parent, crossover to produce offspring, and mutation to update genetic information. After going through a series of evolutionary processes, a new generation is formed, the fitness values are recalculated, and the RCGA cycle continues until the predetermined number of generations is reached. The optimal solution to the objective function is the chromosome with the best fitness value.

3 Data collection and problem formulation

3.1 Data collection

Based on research conducted by Gupta, Mehlawat, and Saxena [12], there are three classes that mainly represent the characteristics of each stock. Description of each class is as follows.

1. Liquid Class
   Stocks in this class have high liquidity but tend to have lower returns compared to the high-yield class.

2. High-yield Class
   This class includes stocks with high risk and high returns, thus having a high standard deviation. However, liquidity in this class is low.

3. Less-risky Class

   Stocks in this class do not fall into either the liquid or high-yield categories.

Thus, the information needed to build the OVO SVM model included long-term returns, risk, and liquidity. Long-term returns were measured through the average weekly returns over the past year. The choice of weekly returns, as opposed to daily or monthly, was based on considerations to avoid excessive daily fluctuations and minimized the risk of losing sensitivity to market changes. The risk of each stock was analyzed using the standard deviation of weekly returns over the past year. In the context of risk analysis in stock investment, the standard deviation of returns provided an overview of the distribution of possible returns over time. Next, stock liquidity was measured using the turnover rate to analyze the ability to buy or sell a stock without incurring significant losses [11]. The turnover rate was defined as the ratio between the average number of shares traded and the number of shares held by the public [27].

To train and evaluate the classification model, it was essential to have a large and representative sample of stocks to reflect the investment potential in the Indonesian stock market. This dataset was compiled from stocks listed in the Kompas100 index, along with the top ten performing stocks from each sectoral index in Indonesia, covering the period between January 1 and December 31, 2023. The stocks in the Kompas100 index are chosen for their high liquidity and market capitalization, while sectoral stocks represent diverse investment potentials in Indonesia.

The stocks used in this study were those with at least one year of availability and high quality. In the data processing stage, ELTY, MYRX, BHIT, BNBR, CASA, BLTA, and TAXI stocks were excluded from the analysis due to many rows of data with zero trading volume and returns. The dataset used in this research consisted of 137 selected stocks, consisting of sectoral stock data in Table 1 and stocks from the Kompas100 index. This dataset was then randomly divided into two sets using the train_test_split module from the sklearn.model_selection library in Python. The first set was the training set, which comprised 94 stocks from both the Kompas100 and sectoral indexes.

Table 1: Sectoral stocks excluding stocks in Kompas100 index.

Sectors	**Stock Ticker**
Energy	BYAN, DSSA, ITMG, TCPI
Industry	IMPC, ARNA, HEXA, MLIA, KBLI, SKRN
Technology	DCII, BELI, MTDL, MCAS, NFCX, WIRG
Property	MPRO, LPKR, MKPI, RISE
Health	CARE, PRDA, TSCP, SAME, BMHS
Finance	SMMA, MEGA, DNET
Transportation	TMAS, BIRD, SDMR, GIAA, HATM, ELPI, IMJS
Non-cyclic	PANI, HMSP, CMRY
Cyclic	MSIN, FILM, KPIG, BOGA

The second set, serving as the test set, consisted of the remaining 43 stocks to form a portfolio.

The first dataset was labeled using the third quartile of each criterion to avoid subjectivity in determining the high and low values of an evaluation metric. The choice of using quartiles is adjusted according to the characteristics and distribution of the data. Stocks with liquidity above the third quartile were first included in the liquid class with label 0, whereas the remaining stocks with risk above the third quartile were included in the high-yield class with label 1. Stocks that had not been labeled were included in the less-risky class. Seventy stocks from the first dataset that already had labels were then randomly selected to be training dataset, while the remaining 24 stocks were the validation dataset.

3.2 Multi-objective model

In this study, an optimal multi-criteria portfolio is constructed using a multi-objective model approach. This approach allows investors to consider multiple criteria simultaneously in their investment decision-making process without having to sacrifice one criterion for another [11].

The multi-objective model is based on the assumption that an investor intends to select h stocks from a set of 43 stocks in the testing dataset to construct a portfolio with minimal risk while maximizing short-term and long-term returns and ensuring high liquidity, using the entire available capital without engaging in short-selling. Let $R_{i,j}$ denote the return for the ith week of the jth stock. The short-term return of the jth stock, represented as R_j^{12}, is calculated based on the average of weekly returns over the last 12 weeks, while the long-term return of the jth stock, namely R_j^{52}, is computed based on the average weekly return over the past year. Finally, the liquidity of the jth stock, denoted as L_j, is calculated as the ratio of the average number of shares traded to the number of shares held by the public.

Assume that there are n stocks in the dataset, that h is the number of stocks in a portfolio, that x_j is the proportion of funds invested in stock j relative to the total funds, that z_j is a binary variable indicating whether stock j is selected (1) or not (0), and that l_j and u_j are the minimum and maximum percentage of funds invested in stock j, respectively. With the assumption of inability to short-sell, the multi-objective model is defined as follows:

$$\max\ f_1(x) = \sum_{j=1}^{n} R_j^{12} x_j, \tag{1}$$

$$\max\ f_2(x) = \sum_{j=1}^{n} R_j^{52} x_j, \tag{2}$$

$$\min\ f_3(x) = \sum_{i=1}^{52} \frac{\left|\sum_{j=1}^{n}(R_{i,j} - R_j^{52})x_j\right| + \sum_{j=1}^{n}(R_j^{52} - R_{i,j})x_j}{104}, \tag{3}$$

$$\max\ f_4(x) = \sum_{j=1}^{n} L_j x_j, \tag{4}$$

subject to

$$\sum_{j=1}^{n} x_j = 1,\ \sum_{j=1}^{n} z_j = h,\ x_j \leqslant u_j z_j,\ x_j \geqslant l_j z_j,\ u_j \geqslant 0,\ z_j \in \{0,1\},$$

for $j = 1, 2, \ldots, n$.

This multi-objective model is used to construct a portfolio considering four different criteria, resulting in four objective functions that need to be optimized according to the investor's preferences. The first objective function (1) aims to maximize short-term returns, which in turn provides an overview of the stock's performance in the short term and is used to evaluate opportunities for quick gains. The model also aims to maximize long-term returns through the second objective function (2), which overviews the stock's performance over a longer period and reflects the stability and growth of the investment.

To avoid the risks arising from stock price fluctuations, the third objective function (3), which utilizes mean semi-absolute deviation, is built with the aim of minimizing risk. In this model, risk considers the difference between the actual weekly returns and the expected long-term returns over a one-year evaluation period. The final objective function (4) aims to maximize liquidity. Analyzing liquidity in a portfolio is important for investors to get an idea of the ability to trade stocks without incurring significant losses. The portfolio is constructed by allocating all investment funds into the selected stocks. To ensure sufficient stock diversification and avoiding inefficient investments, h stocks are selected and indicated by binary variables. Furthermore, upper and lower investment limits were set to restrict the proportion of funds allocated to a single stock.

3.3 Constructing multi-criteria portfolio using RCGA

To simulate the RCGA method, we used the Python software. The first step is population initialization, where each chromosome represents a portfolio. Each gene represents a single stock, while the allele on the gene represents the proportion of that stock in the portfolio. Suppose that an investor wants to form a portfolio based on data containing n stocks. In this study, the number of stocks in the portfolio (h) is randomly determined during the initialization stage, and it is assumed that l and u are the minimum and maximum percentage of the funds invested in each stock, respectively.

First, the investor needs to determine the number of chromosomes in the population (*popsize*) as a limit to the solution search space. Each chromosome constructed has n genes, where the first gene of a chromosome represents the first stock in the data table, the second gene represents the second stock, and so on. Initially, every gene in each chromosome is assigned an allele value of zero. The population initialization stage in this study was the following Algorithm 1.

Algorithm 1 Population initialization

Input: number of stocks in the population (*popsize*), number of stocks in the data (n), lower investment bound (l), and upper investment bound (u).

1: Initialize the population as an empty array to be filled with chromosomes.

2: For each chromosome in *popsize*, do:

- Randomly determine the number of stocks (h) between zero and n.
- Initialize a chromosome as an array of length n with zero values.
- Initialize *selected_gens* as a list of h stocks chosen randomly, ensuring no stock is selected more than once.
- For each selected gene, initialize the allele randomly between l and u.
- Add the new chromosome to the population.

Output: Initial Population.

To evaluate the quality of the formed chromosomes, RCGA enters the evaluation stage. In the context of portfolio formation, the evaluation function is used to assess how well a portfolio meets the established investment criteria. To ensure that the generated chromosomes not exceeding the available capital budget, a penalty function P is applied. This function reduces the fitness value of chromosomes that exceed the budget limit. Chromosomes that produce portfolios with costs exceeding the limit will have lower fitness values. Let x_j be the proportion of stock j. The fifth objective function is

$$f_5(x) = \left| \sum_{j=1}^{n} x_j - 1 \right|,$$

so the penalty function P is defined as

$$P = \begin{cases} 10^4 \cdot f_5(x), & \text{if } f_5(x) \geqslant 10^{-5}, \\ 0, & \text{if } f_5(x) < 10^{-5}. \end{cases}$$

The fitness function for chromosome k, with $k = 1, 2, \ldots, popsize$ with *popsize* being the number of chromosomes in one population, is defined as

$$fit_k = w_1 \cdot f_1(x) + w_2 \cdot f_2(x) - w_3 \cdot f_3(x) + w_4 \cdot f_4(x) - P, \qquad (5)$$

where $w_1, w_2, w_3,$ and w_4 are the weights of the objective functions with $\sum_{i=1}^{4} w_i = 1$ [11]. The weights assigned to the objective functions indicate their importance in building the portfolio. High fitness value of chromosome k indicates how well the solution it provided performs relative to the objective functions. Algorithm 2 simulated the fitness evaluation stage in this study.

Algorithm 2 Fitness evaluation

Input: population, stock risk, long return, short return, liquidity, weights w_1, w_2, w_3, w_4.

1: Initialize $fitness\_results$ as an empty array to be filled with the fitness values of each chromosome.

2: For each chromosome in the population, do:

- Calculate *short_return_value* using equation (1).
- Calculate *return_value* using equation (2).
- Calculate $risk$ using equation (3).
- Calculate *liquidity_value* using equation (4).
- Calculate penalty as the absolute value of the difference between the total investment allocation and one.
- If $penalty > 10^{-5}$, then *penalty* is calculated as $10^4 \times$ penalty, otherwise $penalty = 0$.
- Calculate *fitness_value* using (5).
- Add the fitness value to *fitness_results*.

Output: Fitness evaluation results for each chromosome in the population.

The chromosome with the highest fitness value represents the portfolio with the stock combination that best aligns with the investor's preferences to maximize return and liquidity while minimizing risk. This chromosome is selected as the elite and is directly passed on to the next generation without undergoing evolutionary processes such as crossover and mutation.

Next, the roulette wheel selection operation is performed to select candidate portfolios that best match the investor's preferences as parents. This method works by assigning a selection probability to each chromosome based on its fitness value. Before calculating the selection probabilities, the fitness values of each chromosome are normalized. Normalization ensures that all fitness values fall within a positive range, as fitness values can have both positive and negative ranges. This is done by subtracting the smallest fitness value in the population from the fitness value of each chromosome, then adding a small value such as 10^{-10} to avoid division by zero. The higher the fitness value of a chromosome, the more likely it is to be selected as a parent. The selection stage of RCGA in this study is shown in Algorithm 3.

Algorithm 3 Roulette wheel selection

Input: population, fitness values.

1: initialize $min\_fitness$ with the smallest value from all fitness values in the population.
2: calculate $normalized\_fitness$ by subtracting $min\_fitness$ from the $fitness\_values$ and adding 10^{-10}.
3: calculate $total\_fitness$ by summing all values in $normalized\_fitness$.
4: calculate $probabilities$ for each chromosome by dividing $normalized\_fitness$ by $total\_fitness$.
5: $selected\_index$ is chosen randomly from the population using the previously calculated $probabilities$.

Output: index of the selected chromosome.

The chromosomes selected as parents through the roulette wheel selection operation is paired to create new offspring that combine characteristics from both parents. The offspring are generated using the crossover operation, which swaps portions of genetic information between parent chromosomes.

The next step in this research is Algorithm 4, which denotes the crossover stage of RCGA.

Algorithm 4 *Crossover*

Input: *parent1*, *parent2*, *crossover_rate* (p_c).

1: Initialize r_c randomly with a value between zero and one.
2: If $r_c > p_c$ then $parent1 = child1$ and $parent2 = child2$, if $r_c < p_c$ proceed with:
- Initialize crossover point (C_p) with a value between one and the length of $parent1$.
- $child1$ is created by combining the initial part of $parent1$ with the latter part of $parent2$ after C_p.
- $child2$ is created by combining the initial part of $parent2$ with the latter part of $parent1$ after C_p.

Output: *child1* and *child2*.

The newly formed offspring through the crossover operation will have some of their information updated using the mutation operation, which is performed using Algorithm 5. Next, RCGA will continuously perform the stages of evaluation, selection, crossover, and mutation until the generation limit set by the investor is reached. The chromosome with the best fitness value from the final generation is considered the portfolio that best aligns with the investor's preferences.

4 Result and analysis

The initial phase of developing the OVO SVM model with the RBF kernel was conducted in Python using the SVC module to build the classification model and the OVO classifier module to handle multi-class problems with an OVO approach. Both modules are sourced from the scikit-learn library. The OVO SVM model has two parameters: the penalty parameter C, which controls the classification error, and the parameter γ, which determines the maximum distance of data points that influence the hyperplane. In the initial

Algorithm 5 Mutation

Input: chromosome, *mutation_rate* (p_m), l, u

1: Initialize *mutated_chromosome* as a copy of the original chromosome.

2: Initialize r_m randomly with a value between zero and one.

3: If $r_m < p_m$ then proceed with:

- Determine *num_mutations* randomly with a value between one and the length of the chromosome.
- Initialize *mutation_indices* as a list of chromosome (gene) indices randomly determined up to *num_mutations*.
- For each selected gene, if the allele value is not zero, the allele value is randomly changed between l and u, while if the allele value is zero, the allele value will not be changed.

Output: *mutated_chromosome*

model, where the parameters C and γ were not explicitly set, the accuracy was only 0.54. The model struggled to recognize data belonging to the high-yield class, as indicated by its very low precision and accuracy for that class.

Parameter optimization was performed to find a combination that could enhance the model's performance using grid-search and GA-SVM. The grid-search method revealed that the best parameter combination for the combined data was $C = 10$ and $\gamma = 0.1$. The OVO SVM model produced with these parameters achieved an accuracy of 0.75 and showed significant overall performance improvement, though it still faced challenges with high-yield class data. The results from the GA-SVM method indicated that the best parameter combination for the data used was $C = 403.4645$ and $\gamma = 0.1242$, with an accuracy of 0.88. Based on the balanced evaluation metrics across each class, this model was the most optimal and thus applied to the test stock data to construct the portfolio.

The OVO SVM model developed using the GA-SVM method classified 15 stocks into the liquid and less-risky classes. Additionally, 13 stocks were classified into the high-yield class. Table 2 presents the average values of each criterion for each class. The liquid class consisted of stocks with high liquidity, as reflected by an average liquidity of 3.1147% for this class. This classification indicated that a significant portion of the stocks in the test dataset has adequate liquidity. The high-yield class, as the name suggested, had the highest average risk of 7.2721% with a long-term average return of 0.8337%, signaling strong investment potential. On the other hand, the less-risky class had an average risk of 3.7069%, which was lower compared to the other two classes, though it did not offer high returns.

Table 2: Average criterion for each class.

Class	**Long return**	**Risk**	**Liquidity**
Liquid	0.180615	5.247010	3.114714
High yield	0.833655	7.272134	1.897383
Less risky	−0.422793	3.706849	1.901513

From the classified testing dataset, the stocks that best align with the investor's preferences are selected to form a portfolio using a multi-objective model solved by RCGA. During the search process, the population size (*pop-size*) is set at 50, and the number of generations is limited to 1,000 to constrain the search space for the optimal solution. To ensure stock diversification, a crossover probability of 0.3 and a mutation probability of 0.2 are specified. Additionally, the minimum and maximum percentage of funds invested in a single stock are set between 0 and 0.2 inclusive. The weights for each objective function are adjusted according to the investor's preferences for short-term return, long-term return, risk, and liquidity criteria.

In this study, nine portfolios were constructed to understand the impact of each criterion on portfolio performance. These nine portfolios were built under several scenarios, such as balancing all four criteria, focusing on a single criterion, and building portfolios formed for extreme cases by ignoring one criterion. Portfolio 1 is constructed under the assumption that the investor considers all criteria equally important, thus the weights for the objective functions in the multi-objective model are set equally at 0.25.

Out of the total 43 stocks in the combined testing data, RCGA allocated investment funds into 11 selected stocks, comprising two stocks from the liquid class, six stocks from the high-yield class, and three stocks from the less-risky class. The distribution of investment funds in Portfolio 1 is shown in Figure 1. The largest fund allocation is given to the TPIA stock from the high-yield class, while the smallest allocation is given to the HEXA stock from the liquid class. The expected weekly return for this portfolio is relatively balanced for both the long term and short term, at 0.6683% and 0.7867%, respectively. Furthermore, Portfolio 1 has risk level of 67.78% and liquidity level of 1.81%.

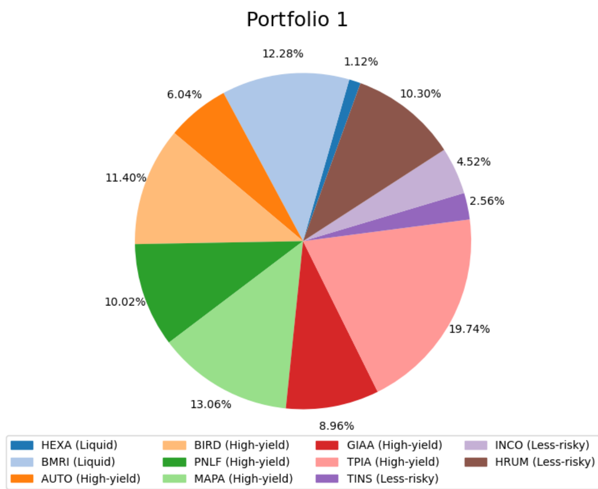

Figure 1: Stocks allocation in Portfolio 1.

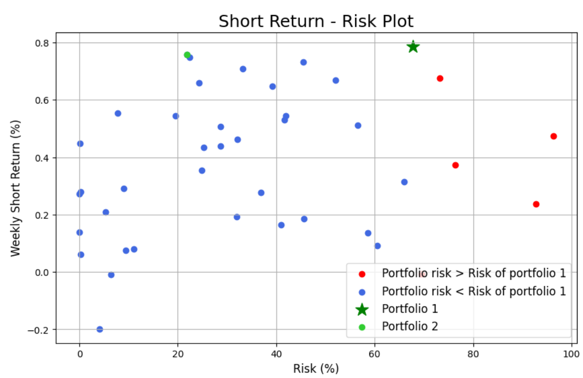

Figure 2: Plot of short return by risk of all optimal portfolios.

To examine the relationship between risk and return, both in the short-term and long-term, steps similar to those in the previous analysis are taken.

Portfolio 2 is built as a modification of equation (5). In this construction, the first objective function f_1 is set to be a constraint, that is, $f_1(x) = r_{sr}$, with $r_{sr} \in \{0.01, 0.02, 0.03, \dots, 0.80\}$, while the other objective functions still remained the same. Thus, the weights of $w_2, w_3,$ and w_4 are equal to $\frac{1}{3}$. The following is the fitness function of Portfolio 2

$$fit_k = w_2 \cdot f_2(x) - w_3 \cdot f_3(x) + w_4 \cdot f_4(x) - P_1,$$

where $P_1 = P + P_{sr}$, with $P_{sr} = |f_1(x) - r_{sr}| \cdot 10^4$.

Figure 2 illustrates the relationship between risk and short-term return, obtained by first determining the short-term return target and then constructing a portfolio that balances all other criteria. These portfolios are generated by first setting the short-term return target, after which the RCGA is executed to maximize liquidity and long-term return while minimizing risk in a balanced manner.

Portfolio 1 has the highest short-term return with medium risk, similar to Portfolio 2 which is also set with equal weights for all objective functions. This means that high risk does not guarantee high short-term return. The red circles represent portfolios with higher risk than Portfolio 1 but with lower short-term returns. There is one green circle labeled Portfolio 2 with a relatively high short-term return expectation and lower risk compared to Portfolio 1. Figure 3 provides an illustration of the relationship between long-term return and risk. The mapped portfolios do not form a diagonal line because Portfolio 1 has a higher long-term return expectation. Both Figures 2 and 3 show that Portfolio 1 is similarly optimal.

Portfolio 1 was constructed under the assumption that investors balance all four criteria, but in reality, some investors tend to lean more heavily towards one criterion over others. Table 3 summarizes the weights used to build Portfolios 3, 4, and 5. Portfolio 3 was constructed with a primary focus on return, assuming that the investor readily takes on high risk or low liquidity. Portfolio 4 was built with an emphasis on risk, where the investor seeks stability while disregarding return and liquidity. Portfolio 5 was designed to seek out stocks with high liquidity.

Additionally, portfolios are created for extreme scenarios where the investor intentionally ignores one criterion in the portfolio construction process.

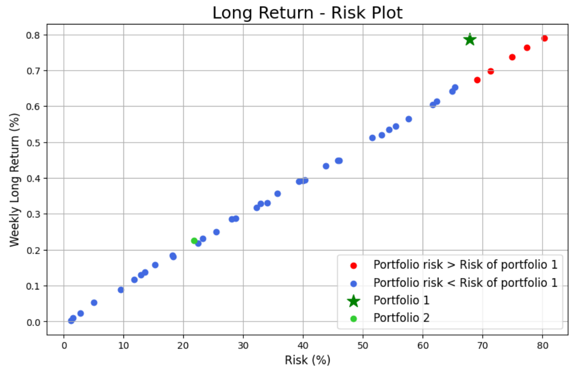

Figure 3: Plot of long return against risk of all optimal portfolios.

Table 3: Weights of Portfolios 3, 4, and 5.

Portfolio	$\mathbf{w_1}$	$\mathbf{w_2}$	$\mathbf{w_3}$	$\mathbf{w_4}$
3	0.3	0.4	0.15	0.15
4	0.15	0.15	0.6	0.1
5	0.1	0.1	0.1	0.7

This approach is taken to understand the impact of a particular criterion on portfolio performance. Portfolio 6 was formed under the assumption that the investor does not consider risk at all. Portfolio 7 is designed for investors who seek quick gains, while disregarding long-term returns. Portfolio 8 is constructed by omitting short-term return considerations entirely. The final extreme case is Portfolio 9, where the investor does not consider liquidity during the portfolio construction process. The weights used to build these three portfolios are summarized in Table 4.

Table 4: Weights of Portfolios 6, 7, 8, and 9.

Portfolio	$\mathbf{w_1}$	$\mathbf{w_2}$	$\mathbf{w_3}$	$\mathbf{w_4}$
6	0.4	0.4	0	0.2
7	0.8	0	0.1	0.1
8	0	0.8	0.1	0.1
9	0.3	0.3	0.4	0

In Figure 4, it is observed that the portfolio with the highest risk is Portfolio 6, which was constructed by ignoring the risk criterion. Although this portfolio does have the highest expected long-term return, there are several other alternative portfolios that offer higher short-term return expectations with lower risk levels. Portfolio 9, which disregards liquidity considerations, is expected to have a negative short-term return and a low long-term return, though it does feature low risk.

For comparison purposes, the portfolio selection method proposed by Gupta et al. [11] was applied. The dataset consisting of 43 testing stocks was categorized into three distinct portfolios by GA-SVM: one with 15 liquid stocks, another with 13 high-yield stocks, and the final one containing 15 less-risky stocks. The RCGA method was used for each portfolio to identify the top-performing stocks, applying weights that prioritized specific criteria, as detailed in Table 5. Portfolio A, which emphasized liquidity, selected 9 out of the 15 liquid stocks. Portfolio B, focused on returns, included 12 of the 13 high-yield stocks. Lastly, Portfolio C, which prioritized risk, chose 10 of the 15 less-risky stocks.

Table 5: Weights of Portfolios A, B, and C.

Portfolio	$\mathbf{w_1}$	$\mathbf{w_2}$	$\mathbf{w_3}$	$\mathbf{w_4}$
A	0.2	0.25	0.2	0.35
B	0.3	0.35	0.2	0.15
C	0.17	0.23	0.45	0.15

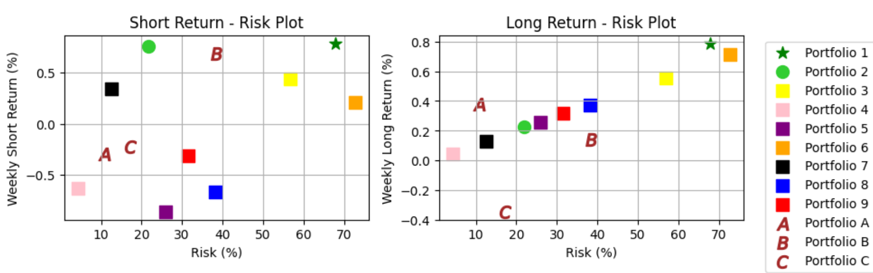

Figure 4: Plot of return compared to risk for all optimal portfolios.

Figure 4 evaluates the performance of all portfolios, including the nine portfolios using our proposed methodology and the three portfolios based on the methodology developed by Gupta et al. [11]. Similar to the previous analysis, Portfolio 1, which is built by balancing all the criteria, has the highest short-term return expectation with risk level above 50%. This portfolio also has a high long-term return expectation, making it a good choice for risk-taker investors. For moderate investors, forming a portfolio by setting a short-term return target and balancing all criteria, as in Portfolio 2, is a good alternative. Furthermore, for conventional investors who avoid high risk, a portfolio could be constructed by ignoring the long-term return criterion, as in Portfolio 7. While Portfolio A has higher long-term return and lower risk compared to Portfolios 1 and 2, its projected short-term return indicates a negative outcome which is lower than Portfolio 7. Portfolio B exhibits a lower long-term return, but a higher short-term return compared to Portfolio 8. Nonetheless, its short-term return is still less than that of Portfolio 2, which also carries a lower risk. Portfolio C is not advisable, as both its short-term and long-term returns are expected to be negative. The other portfolios are considered suboptimal due to the availability of alternative portfolios that offer higher returns for similar risk levels.

5 Conclusions

By classifying Indonesian stocks into three classes, it is concluded that the OVO SVM classification model optimized using the GA-SVM method is better than the model optimized using the grid-search method. The OVO SVM model classifies 13 stocks into the high-yield class, reflecting the growth potential and opportunities for satisfactory returns in the Indonesian stock market. The Indonesian stock market also shows adequate liquidity, with 15 stocks in the liquid class. Additionally, there are 15 stocks with minimal risk in the less-risky class, providing options for investors. Portfolios that balance the four investment criteria have high gains in both short-term and long-term. Investors aiming to boost returns might consider adding more high-yield stocks, whereas those wanting to lower risk could invest in stocks from a less-risky category. It is also observed that by including liquidity criteria, the long-term return performance is enhanced, as shown in Portfolio A. This indicates that adding the liquidity criteria is crucial for optimizing portfolio performance over the long term.

The portfolios in this study were constructed using RCGA to solve the multi-objective model. One characteristic of RCGA is the population initiation stage, which involves randomly generating chromosomes and their allele values, leading to different portfolio compositions each time RCGA is run. This results in inconsistent outcomes, although the fitness values obtained tend to be around the same range. The computational time of RCGA also depends on the number of chromosomes in the population, the number of generations, and the complexity of the fitness function. Future research could explore exact optimization approaches, such as CPLEX suggested by Mutunge and Haugland [22], Jin, Qu, and Atkin [18], and Zhang [29, 30, 31] or data envelopment analysis as suggested by Zhou et al. [32, 33], to solve multi-objective problems and achieve more consistent solutions.

References

[1] Awad, M., and Khanna, R. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers.* Springer Nature, New York, 2015.

[2] Bali, T.G., Peng, L., Shen, Y., and Tang, Y. *Liquidity shocks and stock market reactions,* Rev. Financ. Stud. 27(5) (2014) 1434–1485.

[3] Beardsley, X.W., Field, B., and Xiao, M. *Mean-variance-skewness-kurtosis portfolio optimization with return and liquidity,* Commun. Math. Financ. 1(1) (2012) 13–49.

[4] Chen, J.S., Hou, J.L., Wu, S.M., and Chang-Chien, Y.W. *Constructing investment strategy portfolios by combination genetic algorithms,* Expert Syst. Appl. 36(2) (2009) 3824–3828.

[5] Chen, W., Wang, Y., Gupta, P., and Mehlawat, M.K. *A novel hybrid heuristic algorithm for a new uncertain mean-variance-skewness portfolio selection model with real constraints,* Appl. Intell. 48 (2018) 2996–3018.

[6] Chordia, T., Roll, R., and Subrahmanyam, A. *Liquidity and market efficiency,* Journal of Financial Economics, 87(2) (2008) 249–268.

[7] Fabozzi, F.J., Gupta, F., and Markowitz, H.M. *The legacy of modern portfolio theory,* J. Invest. 11(3) (2002) 7–22.

[8] Fan, A., and Palaniswami, M. *Stock selection using support vector machines,* In International Joint Conference on Neural Networks. Proceedings, (Cat. No. 01CH37222) (2001) 1793–1798.

[9] Flach, P.*Machine learning: The art and science of algorithms that make sense of data.* Cambridge University Press, Cambridge, 2012.

[10] Guijarro, F. *A similarity measure for the cardinality constrained frontier in the mean–variance optimization model,* J. Oper. Res. Soc. 69(6) (2018) 928–945.

[11] Gupta, P., Mehlawat, M.K., Inuiguchi, M., and Chandra, S. *Multi-criteria portfolio optimization using support vector machines and genetic algorithms,* Studies in Fuzziness and Soft Computing, vol 316. (pp. 283–309) Springer, Berlin, Heidelberg, 2014.

[12] Gupta, P., Mehlawat, M.K., and Saxena, A. *A hybrid approach to asset allocation with simultaneous consideration of suitability and optimality,* Inf. Sci. 180(11) (2010) 2264–2285.

[13] Herrera, F., Lozano, M., and Verdegay, J.L. *Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis,* Artif. Intell. Rev. 12 (1998) 265–319.

[14] Hsu, C.W., and Lin, C.J. *A comparison of methods for multiclass support vector machines,* IEEE Trans. Neural Networks Learn. Syst. 13(2) (2002) 415–425.

[15] Huang, C.F. *A hybrid stock selection model using genetic algorithms and support vector regression,* Appl. Soft Comput. 12(2) (2012) 807–818.

[16] Huang, Z., Chen, H., Hsu, C.J., Chen, W.H., and Wu, S. *Credit rating analysis with support vector machines and neural networks: A market comparative study,* Decis. Support Syst. 37(4) (2004) 543–558.

[17] Jimbo, H.C., Ngongo, I.S., Andjiga, N.G., Suzuki, T., and Onana, C.A. *Portfolio optimization under cardinality constraints: A comparative study,* Open J. Stat. 7(4) (2017) 731–742.

[18] Jin, Y., Qu, R., and Atkin, J. *Constrained portfolio optimisation: The state-of-the-art markowitz models,* In International Conference on Operations Research and Enterprise Systems (2016) 388–395.

[19] Kessaci, Y. *A multi-objective continuous genetic algorithm for financial portfolio optimization problem,* In Proceedings of the Genetic and Evolutionary Computation Conference Companion (2017) 151–152.

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

[21] Mitchell, M. An introduction to genetic algorithms. MIT Press, Cambridge, 1998.

[22] Mutunge, P., and Haugland, D. *Minimizing the tracking error of cardinality constrained portfolios,* Computers and Operations Research, 90 (2018) 33–41.

[23] Packard, N.H. *A genetic learning algorithm for the analysis of complex data,* Complex Syst. 4(5) (1990) 543–572.

[24] Rather, A.M., Sastry, V.N., and Agarwal, A. *Stock market prediction and portfolio selection models: A survey,* Opsearch, 54 (2017) 558–579.

[25] Santoso, W., Harun, C.A., Hidayat, T., and Wonida, H. *Market liquidity risk as an indicator of financial stability: The case of Indonesia,* Bank of Indonesia, (2010). 1–22.

[26] Sheppard, C. *Genetic algorithms with python.* Smashwords Edition, Austin, TX, USA 2017.

[27] Vogel, H.L. *Entertainment industry economics: A guide for financial analysis.* Cambridge University Press, Cambridge, 2020 .

[28] Yu, L., Wang, S., and Lai, K.K. *Neural network-based mean–variance–skewness model for portfolio selection,* Comput. Oper. Res. 35(1) (2008) 34–46.

[29] Zhang, P. *Multi-period possibilistic mean semivariance portfolio selection with cardinality constraints and its algorithm,* J. Math. Model. Algor. Oper. Res. 14 (2015) 239–253.

[30] Zhang, P. *An interval mean–average absolute deviation model for multiperiod portfolio selection with risk control and cardinality constraints,* Soft Comput. 20 (2016) 1203–1212.

[31] Zhang, P. *Multiperiod mean absolute deviation uncertain portfolio selection,* Ind. Eng. Manag. Syst. 15(1) (2016) 63–76.

[32] Zhou, Z., Jin, Q., Xiao, H., Wu, Q., and Liu, W. *Estimation of cardinality constrained portfolio efficiency via segmented DEA,* Omega, 76 (2018) 28–37.

[33] Zhou, Z., Liu, X., Xiao, H., Wu, S., and Liu, Y. *A DEA-based MOEA/D algorithm for portfolio optimization,* Clust. Comput. 22(Suppl 6) (2019) 14477–14486.