Information Sciences 670 (2024) 120549

Contents lists available at ScienceDirect

Information Sciences

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

Developing a hybrid system for stock selection and portfolio optimization with many-objective optimization based on deep learning and improved NSGA-III

Mengzheng Lv [a], Jianzhou Wang [a,*], Shuai Wang [a], Jialu Gao [b], Honggang Guo [a]

[a] School of Statistics, Dongbei University of Finance and Economics, Dalian 116025, China
[b] Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China

ARTICLE INFO

Keywords:
Portfolio optimization
Many-objective constrained optimization problem
Deep learning
Stock selection
Improved optimization algorithm

ABSTRACT

Portfolio management is a critical aspect of investment strategies, with the goal to balance the low-risk and high-return investments. Despite this, existing portfolios frequently overlook the integration of stock selection outcomes and underutilize data from listed companies, leading to suboptimal portfolio performance. Addressing these shortcomings, this paper introduces a hybrid system involving stock selection and portfolio optimization. In stock selection, the system employs a combination of convolutional neural network and bi-directional recurrent neural network to predict stock trends. This approach enables the identification of stocks likely to appreciate in value, setting the stage for their inclusion in the subsequent optimization process. For portfolio optimization, the study formulates a five-objective optimization problem that incorporates mean, variance, skewness, kurtosis, and distance-to-default as key considerations. To solve the many-objective constrained optimization problem, an advanced strategy employing a static penalty function and an improved Non-dominated Sorting Genetic Algorithm III (NSGA-III) based on tent chaotic mapping is utilized. The efficacy of the proposed hybrid system is rigorously tested through three sets of ablation experiments alongside two discussions focused on its robustness and computational efficiency. The findings from these investigations reveal that the hybrid system outperforms traditional approaches, reducing risks and improving returns for investors.

1. Introduction

The growing awareness of risk and sophistication in investment concepts has led to an increased demand for investments that offer more than just risk-free returns. Consequently, investors, both individual and institutional, are gravitating towards stocks as a preferred investment vehicle due to their potential for high returns accompanied by high risks. This shift underscores the critical role of diversification as a strategy for risk management. In this context, the construction of an effective and efficient portfolio becomes imperative [1], as it serves to optimize the balance between risk and return [2].

The problem of portfolio selection, initially introduced by Markowitz in 1952 [3], marked the beginning of quantitative finance by formulating investment decisions into the mean–variance (M−V) model. This model, by focusing on the risk-return trade-off, aimed to

* Corresponding author.
E-mail addresses: mengzheng730@163.com (M. Lv), wjz@lzu.edu.cn (J. Wang), vvs09061513@163.com (S. Wang), jialu_gao@163.com (J. Gao), ghg970612@163.com (H. Guo).

https://doi.org/10.1016/j.ins.2024.120549
Received 8 February 2023; Received in revised form 1 April 2024; Accepted 2 April 2024
Available online 3 April 2024

minimize unsystematic risk by considering the distribution's risks and returns. However, subsequent research indicated that focusing solely on the first two moments (mean and variance) is insufficient for a comprehensive risk assessment. Maringer and Parpas [4] as well as Liu [5] demonstrated the necessity of incorporating higher-order moments—unless returns follow a normal distribution or investors' utility functions are quadratic. Samuelson [6] further highlighted the asymmetry and fat tails in the distribution of asset returns, prompting a shift towards incorporating skewness and kurtosis into the objective function. Recent advancements have seen the development of models that integrate these aspects. Zhen and Chen [7] obtained a mean–variance-skewness (M−V−S) portfolio frontier and a closed-form strategy for investors with a cubic utility function, while Abid et al. [8] expanded this to include kurtosis, proposing a mean–variance-skewness-kurtosis (M−V−S−K) model for diversified portfolio construction. Li et al. [9] further enriched this framework by adding entropy, resulting in a mean–variance-skewness-kurtosis-entropy (M−V−S−K−E) hybrid approach. Despite these developments, incorporating a company's financial information remains essential, as evidenced by the Kohn-Merchant-Vasicek (KMV) method [10], which emphasizes credit risk assessment through various financial factors. Wang et al.'s [11] introduction of a mean–variance-KMV (M−V−KMV) model signifies a notable improvement over the traditional M−V model by integrating financial information. However, the integration of higher-order moments with financial information into the objective function remains unexplored. **This paper aims to address this gap by considering five variables—risk, return, skewness, kurtosis, and distance-to-default—as the foundational elements of our objective functions.**

In optimizing objective functions, the original M−V model is fundamentally a quadratic programming (QP) problem. Solutions to QP problems, especially those with both equality and inequality constraints, typically employ Lagrange multipliers and Karush-Kuhn-Tucker (KKT) conditions, necessitating convex objective functions [12]. However, confirming the convexity of high-dimensional objective functions poses significant challenges, leading researchers to explore *meta*-heuristic optimization algorithms for approximate solutions. Kaucic et al. [13] employed a level-based learning swarm optimizer (LLSO) for addressing large-scale portfolio optimization challenges, demonstrating superior performance. Similarly, Bedoui et al. [14] applied a genetic algorithm (GA), derived

Table 1
Review of portfolio selection model, optimization algorithm and forecasting model.

Models	References	Notes	Benefit	Drawback
Portfolio Selection Models				
M−V	[3]	**M−V** model is the most classic model to convert portfolio into a mathematical problem.	First convert a portfolio problem to a mathematical model.	These portfolio models only use stock data, but not use financial data of public company.
M−V−S	[7]	**M−V−S** model adds a skewness objective on the basis of **M−V** and enriches the objective function.	It utilizes the information of some higher-order moments.	
M−V−S−K	[8]	**M−V−S−K** model combines objectives of skewness and kurtosis and **M−V** model, which covers more information in the objective function.	More information on high-order moments is exploited.	
M−V−S−K−E	[9]	**M−V−S−K−E** introduces information entropy into M−V−S−K portfolio with two auxiliary models for portfolio optimization problem.	The introduction of **entropy** can eliminate the measure of the information uncertainty.	
M−V−KMV	[11]	**M−V−KMV** model utilizes more information about high-order moments and financial data.	Financial data is used through **KMV** method.	High-order moments information is not applied in this model.
Optimization Algorithms				
LLSO	[13]	**LLSO** is used to solve M−V−S−K and achieved good results.	Meta-heuristic optimization algorithms can solve optimization problems and consume as little computing resources as possible.	These *meta*-heuristic optimization algorithms cannot work well when solving many-objective optimization algorithms.
GA	[14]	**GA-based** heuristic algorithm is used to find the cardinality constrained efficient frontier.		
NSGA-II	[9,17]	**NSGA-II** can solve portfolio optimization and getting weights of each stock.		
Forecasting Model				
DTM	[21]	**DTM** is used to predict price of stocks, providing estimates of the mean and variance.	Fast running speed.	The information on multivariate are not completely used in these models.
AHP + TOPSIS	[22]	**AHP** is used to obtain expert views, and then **TOPSIS** is used to rank and select the best stocks.	Choosing stocks is intuitive.	
RNN	[27]	**RNN** and other variants, such as LSTM has the advantages in processing time series problem.	Solve the vanishing of gradient problem.	
CNN	[26]	**CNN** can extract useful information, and then achieve the purpose of dimensionality reduction.	Automatic feature extraction, suitable for high-dimensional data.	

from deep machine learning, to the Conditional Value at Risk (CVaR) [15] portfolio optimization problem. Zhang et al. [16] introduced a hybrid approach combining Ant colony optimization with the Estimation distribution algorithm for tackling mixed-integer quadratic multi-objective portfolio optimization. While Li et al. [9] used fast and elitist multi-objective non-dominated sorting genetic algorithm (NSGA-II) for portfolio optimization, determining stock weights effectively. Additionally, Li et al. [17] proposed a bi-objective mean–variance-Value at Riak-entropy portfolio model, optimized by NSGA-II, enhancing solution accuracy. Despite the efficacy of *meta*-heuristic algorithms like LLSO, GA, and NSGA-II, their suitability for many-objective optimization problems is limited. Specifically, NSGA-II tends to perform optimally in scenarios involving no more than three dimensions. Beyond this, the exponential increase in non-dominated individuals and the complexities of the Pareto dominance relationship complicate the differentiation between superior and inferior solutions. Addressing this, the current work introduces an evolved version of NSGA-II, named NSGA-III [18,19], focusing on many-objective optimization challenges. **Building upon NSGA-III, we propose an enhanced variant that incorporates tent chaotic mapping for improved population initialization, ensuring a more uniform distribution of the initial population.**

Before embarking on portfolio construction, selecting promising stocks for investment is a critical preliminary step. Dhaini and Mansour [20] opted not to select individual stocks, instead utilizing the OR-Library datasets, which, despite being validated for representativeness, cannot be considered universally applicable. Wang et al. [11] employed the ARBR indicator for stock screening from a pool of 100 stocks, yet this approach leans more towards emotional judgment than rational analysis. Corberán-Vallet et al. [21] used damped trend model (DTM) to predict the future stocks price, but this model cannot make full use of various information of stocks. Vasantha and Udaya [22] enhanced the Analytic Hierarchy Process (AHP) with expert opinions, subsequently applying the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) for ranking and selecting prime stocks. However, the reliance on subjective expert insights is less effective for the volatile nature of stock assets. Given the stock market's susceptibility to volatility and extreme events, direct predictions of closing prices are generally not recommended [23]. A more viable strategy involves accurately forecasting the directional movement of stock prices, assisting investors in making informed decisions [24,25]. The evolution of computational technology, particularly advancements in CPU and GPU capabilities, has significantly propelled the development of deep learning. This progression has enabled the resolution of complex prediction challenges through deep learning methodologies, such as Convolutional Neural Networks (CNN) [26] and Recurrent Neural Networks (RNN) [27]. **Consequently, this paper adopts deep learning techniques for forecasting stock trends prior to the optimization of portfolios.** An overview of the portfolio optimization and forecasting methods utilized is presented in Table 1.

In summary, inspired by various portfolio works, we developed a hybrid system that integrates stock selection and portfolio optimization. For stock selection, a CNN-BiLSTM model was constructed and utilized to forecast trends over specific periods. Based on the forecasted trends, the Shanghai Securities Exchange 50 (SSE 50) was segmented into three groups: an experimental group and two control groups, with the former analyzed to gauge potential returns for investors and the latter serving comparative purposes. In terms

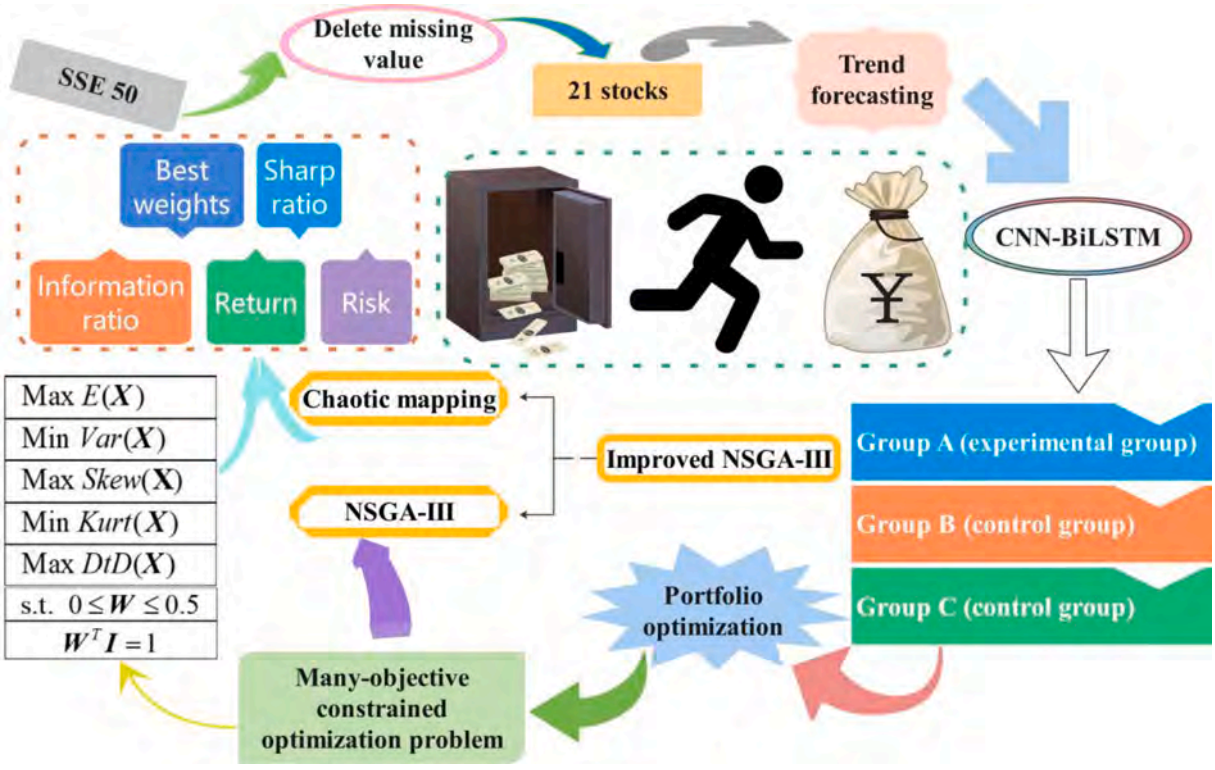

Fig. 1. The flowchart of this article.

of portfolio optimization, we enriched the traditional mean–variance (M−V) model by incorporating higher-order moments and the KMV model, facilitating the creation of many-objective optimizations that consider five factors: mean, variance, skewness, kurtosis, and distance-to-default. NSGA-III, enhanced with a tent chaotic map, was selected as the most suitable algorithm for addressing the complexities of the five-objective problem presented. **The key contributions and innovations of this study are as follows:**

(1) The research transcended conventional stock return analysis by incorporating the listed company's equity value and debt maturity via the KMV method, rendering the investment portfolio more comprehensive;

(2) This study augmented the mean–variance-skewness-kurtosis (M−V−S−K) portfolio with an additional objective function—the distance-to-default variable—aiming for higher returns in the proposed M−V−S−K−K portfolio;

(3) For the five-objective optimization problem, an improved NSGA-III algorithm with chaotic mapping was employed, demonstrating superior performance over other algorithms such as NSGA-II and Strength Pareto Evolutionary Algorithm 2 (SPEA2);

(4) To address the constrained optimization problem, a static penalty function method was applied to transform the constrained problem into an unconstrained one, subsequently determining the optimal penalty coefficient;

(5) This study is pioneering in applying CNN-BiLSTM for stock selection, successfully predicting stock price trends with significantly enhanced performance.

The structure and method of this study are shown in Fig. 1. Section 2 introduces the higher-order moment portfolio and NSGA-III; Section 3 details our hybrid approach; Section 4 presents preliminary research data, evaluation metrics, and the operational environment; Section 5 describes comparative experiments conducted to validate the proposed portfolio's performance; Section 6 explores the sensitivity and computational efficiency of our algorithm; and Section 7 summarizes the research comprehensively.

2. Methodology

For the stock selection part, we developed the CNN-BiLSTM model to forecast the future trends of stocks over a period of time. The principle of CNN and BiLSTM can be seen in [28,29]. Subsequently, in the portfolio optimization part, the KMV model along with an enhanced version of NSGA-III were employed to construct and optimize the M−V−S−K−K portfolio, demonstrating an integration of advanced forecasting and optimization techniques in financial analysis.

2.1. KMV model

The KMV model, developed by KMV Company, is designed to estimate the default probability of borrowing companies [10]. A company is considered to default when the projected market value of its assets falls below the face value of its liabilities. The metric "distance-to-default" (DtD) quantifies the gap between the expected asset value and the threshold at which default occurs. The implementation of the KMV model involves three specific steps:

2.1.1. Calculate the company's market value S_C and volatility σ_C

The Black-Scholes option pricing formula [30] is used to estimate the market value S_C and volatility σ_C of the company's equity.

$$C = S_C \cdot N(D_1) - e^{-rW} \cdot L \cdot N(D_2), \tag{1}$$

where C is the company's equity value, L is the book value of liabilities, S_C is the marked value of the company's assets, W is the debt maturity, r is the risk-free interest rate, and $N(\cdot)$ is the normal distribution cumulative probability function. D_1 and D_2 are determined by Eq. (2–3).

$$D_1 = \frac{\ln(S_C/L) + W \cdot \left[r + (1/2)\sigma_C^2\right]}{\sigma_C \cdot \sqrt{W}}, \tag{2}$$

$$D_2 = D_1 - \sigma_C \cdot \sqrt{W}, \tag{3}$$

where σ_C is the volatility of C.

Furthermore, the relationship of σ_C and σ_E can be obtained by Eq. (4).

$$\sigma_E = (S_C/C) \cdot N(D_1) \cdot \sigma_C. \tag{4}$$

2.1.2. Calculate the distance-to-default

The default exercise point is determined by summing the short-term liabilities with half of the long-term liabilities based on a large number of empirical analyses. At the same time, the company reaches a critical point where it becomes prone to credit risk. Eq. (5–7) represents the process of calculating the DtD.

$$DEP = STD + 0.5LTD, \tag{5}$$

where DEP is the default exercise point, and STD and LTD are the short-term and long-term debt, respectively.

Assuming that the asset value follows lognormal distribution, we can calculate the DtD through Eq. ((1)–(5)).

$$ADD = (E(S_C) - DEP)/\sigma_C, \tag{6}$$

where *ADD* is the absolute distance-to-default; we can then divide *ADD* by the asset value to get DtD.

$$DTD = ADD/S_C. \tag{7}$$

2.1.3. Calculate the expected default probability (EDP)

After the DtD is obtained, the EDP can be easily obtained by Eq. (8) under certain technical conditions.

$$EDP = N[(DEP - E(S_C))/E(S_C)\cdot\sigma_C] = N(-DTD). \tag{8}$$

2.2. Improved NSGA-III

In this section, we introduce the principle of NSGA-III and the tent chaotic map, and combine them to develop improved NSGA-III.

2.2.1. NSGA-III

Using the basic framework of NSGA-II, NSGA-III also utilizes fast non-dominated sorting to stratify individuals of merged population R_t into different non-dominated frontiers. Let us suppose that the parent population size is M, and the parent population of ith generation is P_i. In the hierarchical-based non-dominated sorting mechanism, two vectors $\{u_p\}$ and $\{r_p\}$ are defined, and p is the individual in the population. The number of individuals dominating p is u_p, and the number of individuals dominated by p is r_p. Thus, we can calculate u_p and r_p for each individual. Since the set of non-dominant individuals of the population is situated in the first layer, the set of individuals in the first layer is $F_1 = \{q|u_q = 0, q \in R_t\}$. Then, the set of non-dominated individuals will be situated in the second layer after removing the first layer. According to this method, the set of individuals in each layer can be obtained as $F_r = \{\text{all individuals } q\big|u_q - r + 1 = 0\}$.

During the environmental selection stage, the individuals from layer 1 to layer L of the non-dominated frontier are placed into the set A_i. If $|A_i| = M$, the parent of the next generation $P_{i+1} = A_i$. If $|A_i| > M$, then the individuals from layer 1 to (L-1) are placed into P_{i+1}, and $M - |A_i|$ individuals are selected from the layer L. NSGA-II uses the crowded-comparison approach in order to preserve population diversity.

Define reference points

The systematic boundary intersection construction weight method is used in NSGA-III; it places points on a normalized hyperplane: a unit simplex with $(N-1)$ dimensions. Specifically, the hyperplane has equal inclination angles for most targets with an intercept of 1 on each axis. If p partitions are thought to be present in all objects, then the sum of reference points in the $N-$*objective* is $H = C_{N+p-1}^{p}$, where H is the sum of reference points. For instance, this paper has chosen the five-objective optimization problem. Thus, if five divisions are determined—that is, $p = 5$—then $H = C_9^5 = 126$ reference points will be defined.

Normalize adaptively populated members

NSGA-III utilizes extreme point normalization in each generation; this helps to not only maintain diversity adaptively but also perfectly solves the difficulties introduced by target values with different scales.

(1) We use the minimum value $\mathbf{z}_t^{\min}$ of all individuals of the population A_i on each dimensional object to form the ideal points $\bar{\mathbf{z}} = (\mathbf{z}_1^{\min}, \cdots, \mathbf{z}_M^{\min})$ of the current population. Thus, the objective values for all individuals are transformed using the ideal point as a reference. Through Eq. (9), we can change the ideal point to the origin so that the object of individual x is transformed.

$$f_t^{'}(\mathbf{x}) = f_t(\mathbf{x}) - \mathbf{z}_t^{\min}, t = 1, 2, \ldots, N, \tag{9}$$

where f is the objective function before extreme point normalization and $f^{'}$ is the objective function after extreme point normalization.

(2) We calculate the extreme points on the target axis of each dimension. After fixing a target direction, the vector of this direction w_i is equal to 1, and the weights of other directions are set to 0, so $\overline{\mathbf{w}} = (1, 0, \ldots, 0)$. Eq. (10) can be the extreme point in the direction of the target axis.

$$F(\mathbf{x}, \mathbf{w}) = \max_{t=1}^{N} f_t^{'}(\mathbf{x})/w_t, \mathbf{x} \in A_i, \tag{10}$$

where F is the extreme point in the direction of the target axis.

(3) We use the obtained M extreme points to form a linear hyperplane. Next, the intercept a_t in each target direction can be calculated, and consequently, the normalized hyperplane can be calculated in combination with $f_t^{'}(\mathbf{x})$. Eq. (11) combines the intercept and ideal points to standardize the objective function.

$$f_t^n(\mathbf{x}) = f_t^{'}(\mathbf{x})/\left(a_t - \mathbf{z}_t^{\min}\right) = \left(f_t(\mathbf{x}) - \mathbf{z}_t^{\min}\right)/\left(a_t - \mathbf{z}_t^{\min}\right), t = 1, 2, \ldots, N, \tag{11}$$

where $\sum_{t=1}^{N} f_t^n = 1$. Since the normalization process and hyperplane creation are finished at extreme points in each generation, NSGA-III can maintain the variety in the space spanned by A_i members adaptively. Consequently, the problems with Pareto optimal front can be solved using NSGA-III.

Associate reference lines

After the reference points are set, it is important to define the reference line, which is the ray from the origin through the reference points in the target space. The association operation aims to relate the individuals to the corresponding reference lines. The specific

content is intended to calculate the vertical distance in A_i to the reference line, and then associate the individual with the reference line closest to it.

Preserve niche

i individuals are selected from the critical layer F_L to associate the next generation population P_{i+1}. After associating the reference lines, the following two situations may occur: (1) the reference points are associated with one or more individuals; (2) no individual is associated with the reference points. This step counts the population members related to reference points by using $P_{i+1} = A_i/F_L$.

The reference points with smallest p_j are selected. When $p_j = 0$, no reference point can be associated with it in the set $P_{i+1} = A_i/F_L$. The following two situations will occur in the layer F_L. (1) If there are one or more individuals associated with the reference points in F_L, the closest individual will be associated with them, and the individual will be added to P_{t+1}. (2) If there is no individual associated with it in F_L, these reference points are no longer considered. If $p_j \geqslant 1$, an individual in $P_{i+1} = A_i/F_L$ has already been associated with the reference points. If there is an individual in F_L associated with it, one is randomly selected and added to P_{i+1}. Repeat the above-mentioned steps until the size of the population in next generation P_{i+1} is equal to M.

2.2.2. Tent chaotic map

Tent chaotic map is a piecewise linear map that is widely used in chaotic encryption systems [31]. The definition of tent chaotic map is as follows:

$$\mathbf{y_{m+1}} = g(\mathbf{y_m}) = \begin{cases} \mathbf{y_m}/\theta, \mathbf{y_m} \in [0, \theta), \\ (1 - \mathbf{y_m})/(1 - \theta), \mathbf{y_m} \in [\theta, 1], \end{cases} \tag{12}$$

where $0 < \theta < 1$. In the desirable range of θ, the system is considered to be in a chaotic state.

In the context of the optimization algorithm, a uniformly distributed initial population across the search space significantly enhances the algorithm's efficiency and precision. Notably, chaotic sequences produced by the tent map exhibit superior distribution and randomness compared to pseudo-random numbers. Leveraging these properties, selecting diverse initial values allows for the generation of a chaotic sequence $\mathbf{y}$ within the range $[0, 1]$. This approach facilitates the effective initialization of the population.

2.2.3. Improved NSGA-III

Building on the foundation of NSGA-III, this study proposes the substitution of pseudo-random number generation with tent chaotic maps for initializing the population's position, thereby enhancing the NSGA-III algorithm. A detailed visualization of the improved NSGA-III and its pseudo-code is provided in Appendices A and B.

3. The hybrid system

The hybrid system encompasses both stock selection and portfolio optimization. Consequently, this chapter outlines the specific steps involved in each component.

3.1. Stock selection

The SSE 50 index constituent stock, as of March 31, 2022, was preliminarily chosen as the experimental dataset. To ensure accessibility to data post-2007, stocks exhibiting extensive missing data and brief historical periods were excluded. Ultimately, a selection of 21 stocks was made for trend forecasting analysis.

The essence of trend forecasting is classification rather than regression. Specifically, we developed a multivariate time-series classification model based on CNN-BiLSTM [32] to classify **stock trend** ($\mathbf{Y}$) using **closing price** ($\mathbf{X_1}$), **trading volume** ($\mathbf{X_2}$), **weekly turnover rate** ($\mathbf{X_3}$) and **weekly return** ($\mathbf{X_4}$), where $\mathbf{Y} = (y_1, y_2, \cdots)^T$, $\mathbf{X_1} = (x_{1,1}, x_{1,2}, \cdots)^T$, $\mathbf{X_2} = (x_{2,1}, x_{2,2}, \cdots)^T$,$\mathbf{X_3} = (x_{3,1}, x_{3,2}, \cdots)^T$ and $\mathbf{X_4} = (x_{4,1}, x_{4,2}, \cdots)$. Given that past values of these factors could influence future trends, a sliding window approach [33] was implemented in the stock selection phase. Formula (13) can describe the multivariate time-series classification when the sliding window size is equal to k. A detailed schematic of the trend forecasting model is provided in Appendix C.

$$\begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \cdots \\ x_{1,k} \end{pmatrix}, \begin{pmatrix} x_{2,1} \\ x_{2,2} \\ \cdots \\ x_{2,k} \end{pmatrix}, \begin{pmatrix} x_{3,1} \\ x_{3,2} \\ \cdots \\ x_{3,k} \end{pmatrix}, \begin{pmatrix} x_{4,1} \\ x_{4,2} \\ \cdots \\ x_{4,k} \end{pmatrix} \Rightarrow y_{k+1}. \tag{13}$$

The CNN-BiLSTM model developed in this study offers several key advantages. Firstly, the CNN component is capable of autonomously extracting features due to its inherent local connection and weight sharing properties. Secondly, the BiLSTM component effectively processes both past and future information, enhancing the model's ability to discern long-term dependencies within the input sequences. By integrating CNN and BiLSTM, the model can consider temporal and spatial changes concurrently. Through training and testing on individual stock data, high-performance models were derived and subsequently employed for predicting short-term stock trends. To facilitate portfolio optimization, the 21 selected stocks were categorized into three distinct groups.

3.2. Portfolio optimization

This section describes how the portfolio was built and optimized in terms of objection function and the method used for solving the optimization.

3.2.1. Objective function and constraint condition

The classical M−V model can be viewed as a two-objective problem with equality and inequality constraints; see following:

$$\begin{cases} \text{Max } E(\mathbf{X}) \\ \text{Min } Var(\mathbf{X}) \\ \text{s.t. } \mathbf{W} \geqslant 0, \\ \mathbf{W}^T\mathbf{I} = 1. \end{cases} \tag{14}$$

Regarding formula (14), the two objective functions—expectation and variance—are shown in Eq. ((15)–(16)).

$$E(\mathbf{X}) = \mathbf{W}^T \cdot \mathbf{X} = \sum_{i=1}^{n} w_i \cdot X_i, \tag{15}$$

$$Var(\mathbf{X}) = \mathbf{W}^T \cdot \overline{\overline{\mathbf{V}}} \cdot \mathbf{W} = \sum_{i=1}^{n} w_i^2 \cdot \sigma_i^2 + 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_i \cdot w_j \cdot \sigma_{ij}, \tag{16}$$

where $\mathbf{I}$ is a row vector with all-one value, $\mathbf{X} = (X_1, X_2, \cdots, X_n)^T$ constitute the expected return of many stocks, $\mathbf{W} = (w_1, w_2, \cdots, w_n)$ constitute the weights of corresponding stocks, $\sigma_{ij} = cov(X_i, X_j)$ constitute the covariance of the returns, and $\overline{\overline{\mathbf{V}}}$ is the covariance matrix; see Eq. (17).

$$\overline{\overline{\mathbf{V}}} = \begin{bmatrix} \sigma_{11}\cdots\sigma_{1n} \\ \cdots\cdots\cdots \\ \sigma_{n1}\cdots\sigma_{nn} \end{bmatrix}. \tag{17}$$

Regarding the proposed M−V−S−K−K model, three new objective functions about higher-order moment and DtD can be added; see Eq. (18–20).

$$Skew(\mathbf{X}) = E\left(\mathbf{W}^T(\mathbf{X} - \overline{\mathbf{X}})^3\right), \tag{18}$$

$$Kurt(\mathbf{X}) = E\left(\mathbf{W}^T(\mathbf{X} - \overline{\mathbf{X}})^4\right), \tag{19}$$

$$DtD(X) = \sum_{i=1}^{n} X_i \cdot dtd_i, \tag{20}$$

where dtd_i is the distance-to-default of the ith stock.

Investors typically seek to maximize returns while minimizing risks. To this end, expectation maximization and variance minimization serve as intuitive methods.

A positive skewness implies a distribution where the mean exceeds the median, leading to a right-skewed characteristic. This means the right tail of the probability density function (PDF) extends longer than the left, with most values clustering to the left of the mean. **Therefore, maximizing skewness is crucial to enhance the likelihood of positive returns.**

The kurtosis of asset returns is used to measure the leptokurtosis and fat-tail characteristics of the PDF. Higher kurtosis suggests a distribution with pronounced peaks and heavier tails, indicating a greater probability of extreme outcomes. **Thus, minimizing kurtosis is strategic for reducing the incidence of extreme events.**

The DtD measures the gap between the expected value and the default point, with a larger distance signifying a reduced likelihood of company default. **Maximizing DtD is essential for ensuring robust risk management.**

The portfolio's constraints were defined by an equality and an inequality condition based on formula (14), emphasizing that asset weights cannot be negative—a no-short selling constraint—and limiting the weight of any single asset to no more than 0.5 to promote diversification and ensure the effective utilization of funds. Consequently, the sum of the weights must equal one, adhering to portfolio balance.

Thus, the M−V−S−K−K portfolio can be seen in formula (21).

$$\begin{cases} \text{Max } E(\mathbf{X}) \\ \text{Min } Var(\mathbf{X}) \\ \text{Max } Skew(\boldsymbol{X}) \\ \text{Min } Kurt(\mathbf{X}) \\ \text{Max } DtD(\mathbf{X}) \\ \text{s.t.} 0 \leqslant \mathbf{W} \leqslant 0.5, \\ \mathbf{W}^T\mathbf{I} = 1. \end{cases} \tag{21}$$

3.2.2. Optimization solution

Set $\overline{\overline{\mathbf{O}}}(\mathbf{X})$ represent a matrix encompassing all the objective functions. Given that NSGA-III inherently seeks to minimize the objective function values, an inversion of sign is applied to any objective function that necessitates maximization, as illustrated in Eq. (22).

$$\overline{\overline{\mathbf{O}}}(\mathbf{X}) = [-E(\mathbf{X}), Var(\mathbf{X}), -Skew(\mathbf{X}), Kurt(\mathbf{X}), -DtD(\mathbf{X})]^T. \tag{22}$$

The M−V−S−K−K model incorporates two constraints, necessitating the adaptation of the problem into an unconstrained format for simplification. This transformation employs the penalty function method, specifically utilizing the static penalties approach [34], detailed in Eq. (23).

$$\mathbf{fit}(\vec{\mathbf{a}}) = \mathbf{h}(\vec{\mathbf{a}}) + \sum_{k=1}^{m} \left(C_k \cdot \max[0, f_k(\vec{\mathbf{a}})]^2\right), \tag{23}$$

where C_k is the coefficients of static penalties, m is the number of constraints, $\mathbf{h}(\vec{\mathbf{a}})$ is the original objective function when unpenalized, $\mathbf{fit}(\vec{\mathbf{a}})$ is the penalized objective function, and $f_k(\vec{\mathbf{a}})$ is the constraint condition. When the constraints are satisfied, denoted by $f_k(\vec{\mathbf{a}}) = 0$, no penalty is imposed. Conversely, when the constraints are violated, indicated by $f_k(\vec{\mathbf{a}}) > 0$, a penalty is applied, resulting in an increase in the fitness value.

Based on Eq. (23), we made some small improvements to static penalties. In the first constraint in M−V−S−K−K, if one of the

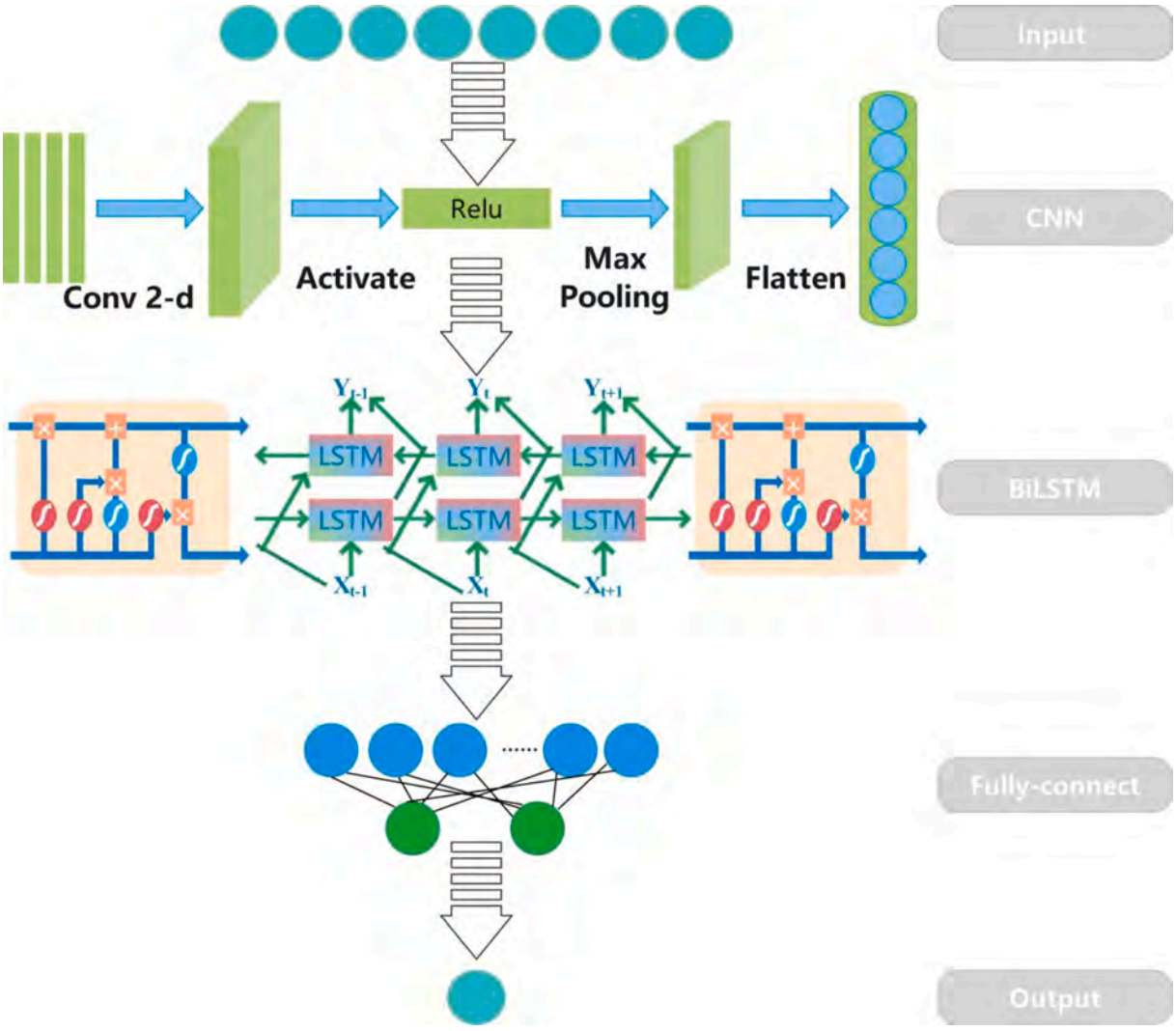

Fig. 2. The structure of CNN-BiLSTM.

weights w_i was found to be between 0 and 0.5, the coefficient $C_{1,i}$ was considered to 0; otherwise, the value was considered 1. In the second constraint, if the difference between the sum of the weights and 1 was found to be within the interval of −1e-3 and 1e-3, set C_2 was considered to be 0; otherwise, the value was considered 1. Then, we used C to combine two coefficients using the following formula: $C = \sum_{i=1}^{n} C_{1,i} + C_2$. Accordingly, the final penalized objective function can be seen in Eq. (24).

$$\overline{\overline{\mathbf{fit}}}(\mathbf{X}) = \overline{\overline{\mathbf{O}}}(\mathbf{X}) + R \cdot C, \tag{24}$$

where R is defined by ourselves, which is necessary to determine the optimal coefficient through multiple experiments.

4. Preliminary

Prior to conducting the experiment, the research data about assets, evaluation metrics of forecasting and portfolios, and operating environments were introduced as the preliminary.

4.1. Research data

The experimental dataset used in this study was initially derived from the China SSE 50 index available in the RESSET Database. This index features stocks ranked by total market capitalization, highlighting those most indicative of the A-share market. However, due to significant missing data, 29 stocks were initially excluded, resulting in a final dataset comprising 21 stocks. Appendix D provides a basic statistical description of these stocks, including daily and weekly data.

In the hybrid system, we chose weekly and daily data for follow-up experiments. Weekly data, covering the period from January 5, 2007, to March 31, 2022, included a total of 784 data points and were utilized for trend forecasting. Daily data, spanning from January 4, 2016, to March 31, 2022, consisted of 1519 data points and were applied in portfolio assessment. To accurately evaluate the hybrid system's performance, the dataset was split into training and test sets at a 4:1 ratio. Within the weekly data subset, the training set encompassed 627 instances, while the test set comprised 157 instances. Similarly, in the daily data subset, we allocated 1215 instances to the training set and 304 instances to the test set. The training set facilitated the optimization of CNN-BiLSTM parameters for trend prediction, whereas the test set was used to assess classification performance. The structure of the CNN-BiLSTM network is shown in Fig. 2. Regarding portfolio optimization, the training set assisted in deriving optimal stock weights via the improved NSGA-III, and the test set served to evaluate portfolio performance.

4.2. Evaluation metrics

In this part, the metrics of classification and portfolios for performance evaluation are introduced in 4.2.1 and 4.2.2.

4.2.1. Classification metrics

Predicting the future trend of stocks is essentially a binary classification problem. Therefore, in evaluating such a problem, four key metrics—***Accuracy***, ***Precision***, ***Recall*** and ***F1-score***—are commonly utilized for assessment; these metrics are also prevalent in most classification [35,36]. To define these terms:

- ***True Positive*** (*TP*) signifies instances where a sample is correctly identified as positive.
- ***True Negative*** (*TN*) refers to instances where a sample is correctly identified as negative.
- ***False Positive*** (*FP*) denotes instances where a negative sample is incorrectly classified as positive.
- ***False Negative*** (*FN*) indicates instances where a positive sample is incorrectly classified as negative.

These metrics are formally defined and their equations are provided in Table 2. Although ***Accuracy*** is the most direct measure of performance, it fails to provide an objective evaluation of an algorithm's effectiveness, especially in scenarios where the data categories are imbalanced or exhibit extreme bias. ***Precision*** and ***Recall*** have an inverse relationship; an increase in ***Precision*** often results

Table 2
Evaluation Metrics in Classification.

Metric	Definition	Equation
Accuracy	All predictions are positive as a percentage of the total.	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	Correct predictions are positive as a percentage of all predictions are positive.	$\frac{TP}{TP+FP}$
Recall	Correct predictions are positive as a percentage of all true values are positive.	$\frac{TP}{TP+FN}$
F1-score	The Harmonic Mean of ***Precision*** and ***Recall***.	$2\times\frac{\mathbf{Precision}\times\mathbf{Recall}}{\mathbf{Precision}+\mathbf{Recall}}$

Note: $TP+TN+FP+FN=N$ and N is the total number of samples.

in a decrease in ***Recall***. In stock trend prediction, ***Precision*** is prioritized as it measures the accuracy of correctly predicted positive samples (e.g., stocks expected to rise) which are of greater investment interest. The ***F1-score*** serves as a balanced metric that combines ***Precision*** and ***Recall***, offering a harmonic mean of the two. Consequently, ***Precision*** and ***F1-score*** are given greater emphasis due to their relevance in accurately assessing predictive performance in stock trend analysis.

4.2.2. Portfolio metrics

To assess the excess returns of both the proposed portfolios and their comparative counterparts, this study introduces two pivotal metrics: the **Sharpe ratio** [37] and the **Information ratio** [38,39]. The **Sharpe ratio** stands as one of the most foundational indicators within financial analysis, offering a comprehensive measure that simultaneously evaluates risk and return. Conversely, the **Information ratio** is predominantly utilized to gauge the active management capability of fund managers, effectively representing the excess returns generated from taking active risks. Therefore, it is reasonable to choose these two indicators to measure the performance of portfolio. The calculation formulas of these two metrics are shown in Appendix E.

4.3. Operating environment

In this study, all experimental procedures were executed utilizing Matlab 2020a on a Windows 10 operating system. The specific hardware components are detailed as follows: AMD Ryzen 7 5800H 3.20 GHz, 16 GB RAM, and RTX 3050 GPU (4G) were used for Compute Unified Device Architecture (CUDA) acceleration in the convolutional and recurrent neural network. Additionally, to ensure the precision and reliability of the experimental results, each experiment was repeated 100 times. Subsequently, the average value of these iterations was calculated to minimize the potential impact of randomness inherent in the algorithms used, thereby providing more accurate and consistent outcomes.

5. Experiment

This chapter details how the proposed hybrid system was applied to the real stock data. Specifically, the hybrid system was divided into two parts: trend forecasting and portfolio optimization. In order to verify the strength of the hybrid system, different models were compared with our models in the two parts.

5.1. Data preprocessing

Before implementing the experiment, data preprocessing is a necessary step, which involves checking for and addressing missing values and outliers.

5.1.1. Missing values processing

In the majority of cases, if the data is missing at a certain point in time, “NA”, “NAN” or null value will be displayed. However, in some rare instances, a “0” might also signify a missing value. To handle these missing entries, we employed interpolation using the data from the previous day.

5.1.2. Outlier values processing

Outlier means that the value of an individual value in the sample significantly deviates from the rest of the observed value of the sample. Since the daily limit of the stock in the SSE 50 is 10 points, it is normal as long as the daily fluctuation of the daily data is within 10 %. Upon computation, it was found that the daily data has no outlier values.

5.2. Trend forecasting

As stated in section 3.1, we used **closing price** ($\mathbf{X_1}$), **trading volume** ($\mathbf{X_2}$), **weekly turnover rate** ($\mathbf{X_3}$) and **weekly return** ($\mathbf{X_4}$) variables to forecast the trends of the stocks. To highlight the validity and rationality of our system, some classification models were introduced for comparison with CNN-BiLSTM (e.g., traditional machine learning models, neural network models, and more popular deep learning models). The specific information of classification performance is provided in Appendix F-H. Furthermore, the parameters used in these classification models are shown in Appendix I.

5.2.1. Comparison with traditional Machine learning models

In this part, the performance of Support Vector Classification (SVC), K-Nearest Neighbor (KNN), Naive Bayes (NB), and Decision Tree (DT) was evaluated in comparison with CNN-BiLSTM. For these four comparison models, grid search was primarily employed to optimize most of their parameters. Specifically for SVC, the Gaussian RBF kernel function was initially chosen due to its common application; parameters γ_s and c_s were varied within certain ranges until optimal values were identified. Finally, we determined $\gamma_s = 1$, and $c_s = 0.1$. Using this method, we finally determined the number of neighbors $n_k = 20$ in KNN, the maximum depth $h_d = 15$ and maximum leaf nodes $\lambda_d = 10$ in DT. Appendix I shows that the classification performances of SVC, KNN, NB, and DT were much worse than CNN-BiLSTM. This is mainly because the four traditional models could not implement multivariate time series prediction and could only use $\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{X_3}$, and $\mathbf{X_4}$ from the previous period as input for forecasting. Consequently, it is inferred that stock prices are influenced by the data from several previous periods, underscoring the superior predictive capability of CNN-BiLSTM in capturing the

complexities of stock trend forecasting through multivariate analysis.

5.2.2. Comparison with neural network models

In this part, Back Propagation Neural Network (BPNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and Bi-directional Long Short-Term Memory (BiLSTM) were compared with CNN-BiLSTM. Focusing on these neural network models, consider BPNN as a case study. The initial learning rate was set to $l_b = 0.01$, which, being slightly high, necessitated adjustments for optimal training efficiency. Therefore, a drop period ($l_{bp} = 50$) and a drop factor ($l_{bf} = 0.5$) were implemented to progressively reduce the learning rate, facilitating the model's convergence towards a global optimum. Appendix G shows that the ***Accuracy*** and ***F1-score*** of the neural network models were higher than traditional machine learning models. In particular, BiLSTM outperformed the other three neural network models, though its result was still not as good as CNN-BiLSTM (except for stock 600000). That is, the feature extraction ability of CNN could make BiLSTM better for classification.

5.2.3. Comparison with more popular deep learning models

Through the aforementioned two sets of experiments, we have largely ascertained that the CNN-BiLSTM yields the best results in stock trend prediction. However, given the rapid advancements in artificial intelligence in recent years, more popular models such as Transformer [40], Generative Adversarial Network (GAN) [41], and Graph Neural Network (GNN) [42] have achieved significant success in the field of time series. Thus, it's imperative to compare our CNN-BiLSTM model with these models.

Specifically, we constructed two Transformer-based models: Transformer_1 and Transformer_2. Transformer_1 is based on the multi-head self-attention mechanism, and its architecture comprises an input layer, a fully connected layer, a multi-head self-attention layer, a global average pooling layer, and an output layer. In the fully connected layer, we utilized the Rectified Linear Unit (ReLU) activation function to transform the input features into a new feature space. The self-attention layer, being the core of Transformer_1, employs the multi-head attention mechanism, where each head attempts to capture different representations. The pooling layer calculates the average for each feature, thus reducing the number of model parameters and facilitating subsequent feature integration. The output layer, in essence, is also a fully connected layer and employs the Sigmoid activation function for binary classification. Transformer_2, an enhancement of Transformer_1, introduces positional information into the input vector at each sequence position by using sine and cosine functions to generate a positional encoding for every position in the sequence. This positional encoding layer is between the fully connected layer and the self-attention layer, with the rest of the architecture mirroring that of Transformer_1.

Furthermore, we also developed a model based on the Generative Adversarial Network (GAN) and another model on the Graph Neural Network (GNN). The GAN consists of a generator and a discriminator, where the former aims to produce fake data, and the latter endeavors to differentiate between real and fake data. Both the generator and discriminator are neural networks composed of an input layer, a fully connected layer, and an output layer. The distinction lies in the output layer of the generator using the Hyperbolic Tangent (Tanh) activation function to produce fake data, while the discriminator's output layer uses the Sigmoid activation function for binary classification. The GNN updates node representations by aggregating information from neighboring nodes on the graph, capturing the graph's topological structure and relationships between nodes. The GNN consists of two graph neural network layers and an output layer, where data is first multiplied with a learnable adjacency matrix and then further multiplied with a weight matrix. The Sigmoid activation function is still used to implement binary classification when the data passes through the output layer.

The performance comparing the four aforementioned models with CNN-BiLSTM are illustrated in Appendix H, and all the parameters utilized can be found in Appendix I. It's evident that, with the exception of a few stocks where the ***Recall*** outperforms CNN-BiLSTM, the rest lag significantly behind the CNN-BiLSTM. A primary reason for this observation is that although the Transformer, GAN, and GNN are outstanding models proven to be applicable in the time series domain, they are often constrained by the quantity of data, typically performing better with larger datasets. The stock data utilized in this study is based on weekly data, with a sample size of less than 1000, which accounts for the subpar performance of these four models. Moreover, the fact that the recall is considerably high while the other three metrics are not can be attributed to the majority of predictions being positive, resulting in a larger count of *TP* and *TN*, and a smaller count of *FP* and *FN*. As evidenced by the calculation formulas in Table 2, the ***Recall*** is notably higher than the other three metrics, such as the performance of Transformer_1 on stock 600519 and GNN on stock 600585. This is similar to cases where the ***Recall*** is very low because the majority of predictions are negative, as seen with Transformer_1 on stock 600276 and Transformer_2 on stock 600050. Such results don't necessarily signify superior model performance but rather highlight the model's inadequacy in classification capabilities.

5.2.4. Grouping stocks by classification

The classification outcomes unequivocally demonstrate the superior performance of the CNN-BiLSTM model, affirming its efficacy in forecasting the future trends of the 21 stocks under study. After that, 7 stocks (600031, 600036, 600196, 600519, 600585, 600690, 600809) could rise in the future, and another 14 stocks could fall; therefore, we divided the stocks into three groups. We focused on the

Table 3
Classification results.

Group	Stock
Group A	600031, 600036, 600196, 600519, 600585, 600690, 600809
Group B	600000, 600028, 600030, 600048, 600050, 600104, 600276
Group C	600436, 600547, 600570, 600588, 600837, 600887, 601398

Table 4
Stock weights and optimization performance by comparing with classic portfolios.

Group A	600031	600036	600196	600519	600585	600690	600809	Average Return	System Risk	Skew	Kurt	DtD	SR	IR
P_1	0.0988	0.1308	0.0016	0.3485	0.2745	0.0174	0.1293	1.05E-03	3.30E-02	–	–	–	2.051	–
P_2	0.1298	0.2001	0.0108	0.3194	0.1181	0.1659	0.0557	1.15E-03	3.49E-02	0.013	–	–	2.149	0.305
P_3	0.0423	0.1718	0.0816	0.2977	0.2074	0.0603	0.1389	1.18E-03	3.40E-02	0.005	0.722	–	2.276	0.343
P_4	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	1.11E-03	3.41E-02	–	–	–	2.487	0.361
P_*	0.1265	0.1877	0.1113	0.1431	0.1285	0.1158	0.1869	1.25E-03	3.23E-02	0.003	0.733	1.447	**2.572**	**0.397**

Group B	600000	600028	600030	600048	600050	600104	600276	Average Return	System Risk	Skew	Kurt	DtD	SR	IR
P_1	0.0414	0.0408	0.1813	0.2608	0.0277	0.0932	0.3544	3.58E-04	3.45E-02	–	–	–	0.197	–
P_2	0.0069	0.0614	0.1650	0.3919	0.1225	0.1542	0.0975	2.90E-04	3.71E-02	0.056	–	–	0.201	−0.043
P_3	0.0210	0.0415	0.1815	0.3222	0.0736	0.1065	0.2531	2.01E-04	3.71E-02	0.107	1.721	–	0.210	−0.076
P_4	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	1.25E-04	3.77E-02	–	–	–	0.208	−0.051
P_*	0.0511	0.1538	0.3641	0.1514	0.0973	0.0841	0.0986	3.88E-04	3.17E-02	0.073	1.587	1.267	**0.313**	**0.098**

Group C	600436	600547	600570	600588	600837	600887	601398	Average Return	System Risk	Skew	Kurt	DtD	SR	IR
P_1	0.1153	0.2728	0.2280	0.0631	0.0166	0.1498	0.1544	4.89E-04	3.54E-02	–	–	–	0.209	–
P_2	0.1205	0.1073	0.1105	0.0883	0.0421	0.2053	0.3260	4.41E-04	3.47E-02	0.119	–	–	0.410	−0.095
P_3	0.0156	0.1188	0.1701	0.0927	0.0953	0.4262	0.0807	5.05E-04	3.64E-02	0.059	1.445	–	0.372	0.109
P_4	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	4.34E-04	3.53E-02	–	–	–	0.303	−0.097
P_*	0.2614	0.0669	0.1068	0.0649	0.0928	0.2173	0.1904	5.13E-04	3.04E-02	0.188	3.479	1.474	**0.898**	**0.127**

Note: Skew, **Kurt**, **DtD**, **SR**, **IR** indicates that Skewness, Kurtosis, Distance-to-default, Sharpe ratio and Information ratio. P_1, P_2, P_3, P_4 and P_* are M−V, M−V−S, M−V−S−K, same weights portfolio and proposed portfolio.

Table 5
Stock weights and optimization performance by comparing with *meta*-heuristic-portfolios.

Group A	600031	600036	600196	600519	600585	600690	600809	Average Return	System Risk	Skew	Kurt	DtD	SR	IR
P_5	0.1425	0.4164	0.0586	0.1383	0.0852	0.1117	0.0475	9.41E-04	**3.20E-02**	0.011	0.736	1.297	1.944	−0.097
P_6	0.0237	0.0129	0.1338	0.1427	0.2479	0.3501	0.0888	1.17E-03	4.48E-02	0.001	0.638	2.485	1.437	0.296
P_7	0.1763	0.0175	0.0476	0.3672	0.1663	0.1514	0.0738	1.19E-03	3.47E-02	0.007	0.794	1.594	2.231	0.275
P_8	0.2023	0.0000	0.0000	0.3672	0.1276	0.0827	0.2146	9.25E-04	3.48E-02	0.015	0.992	1.041	2.389	0.073
P_9	0.1458	0.1535	0.0000	0.1369	0.0401	0.2463	0.2819	1.99E-04	4.47E-02	0.020	1.018	1.079	1.238	−0.186
P_{10}	0.0852	0.0004	0.1627	0.0616	0.2384	0.3768	0.0784	8.99E-04	3.77E-02	0.006	0.747	1.610	1.910	0.175
P_{11}	0.1526	0.2400	0.0868	0.1165	0.0894	0.1825	0.1322	9.85E-04	3.39E-02	0.007	0.743	1.389	2.415	0.218
P_*	**0.1265**	**0.1877**	**0.1113**	**0.1431**	**0.1285**	**0.1158**	**0.1869**	1.25E-03	3.23E-02	0.003	0.733	1.447	**2.572**	**0.397**

Group B	600000	600028	600030	600048	600050	600104	600276	Average Return	System Risk	Skew	Kurt	DtD	SR	IR
P_5	0.1171	0.0609	0.2382	0.0979	0.1996	0.1886	0.0976	1.13E-04	3.25E-02	0.096	1.842	1.198	0.175	0.042
P_6	0.0981	0.1837	0.1173	0	0.3228	0.2539	0.0242	5.53E-05	3.23E-02	0.092	1.658	0.796	−0.224	−0.140
P_7	0.0337	0.1407	0.1036	0.3493	0.2154	0.1264	0.0310	1.72E-04	3.60E-02	0.051	1.301	1.186	−0.102	−0.069
P_8	0.0000	0.0000	0.2520	0.2695	0.0000	0.0000	0.4712	2.92E-04	5.93E-02	0.237	2.973	1.084	0.194	−0.081
P_9	0.0000	0.0103	0.0816	0.6442	0.0000	0.0079	0.2580	1.62E-04	4.24E-02	0.231	2.690	1.175	0.090	−0.047
P_{10}	0.2193	0.0000	0.1067	0.0000	0.3975	0.0243	0.2523	3.47E-05	3.67E-02	0.163	2.644	0.981	−0.322	0.074
P_{11}	0.0135	0.1902	0.0808	0.2372	0.1761	0.0876	0.2148	2.58E-04	3.44E-02	0.099	1.589	1.124	−0.159	−0.033
P_*	0.0511	0.1538	0.3641	0.1514	0.0973	0.0841	0.0986	3.88E-04	3.17E-02	0.073	1.587	1.267	**0.313**	**0.098**

Group C	600436	600547	600570	600588	600837	600887	601398	Average Return	System Risk	Skew	Kurt	DtD	SR	IR
P_5	0.0985	0.0428	0.0327	0.3368	0.2671	0.1207	0.1012	3.35E-04	4.17E-02	0.113	2.347	1.200	0.308	0.087
P_6	0.0273	0.0108	0	0.4576	0.0027	0.1348	0.3667	8.10E-05	2.77E-02	0.198	2.538	0.930	0.359	0.063
P_7	0.0472	0.1030	0.1285	0.0374	0.1384	0.3581	0.1875	4.36E-04	3.09E-02	0.064	1.691	1.305	0.423	0.095
P_8	0.2904	0.2924	0.1284	0.0248	0.0000	0.0443	0.2233	3.41E-04	4.28E-02	0.234	4.324	0.956	0.573	0.046
P_9	0.1160	0.2809	0.1629	0.0727	0.0000	0.1924	0.1852	3.19E-04	4.37E-02	0.173	3.350	0.868	0.295	−0.076
P_{10}	0.3874	0.1152	0.0761	0.1056	0.0166	0.2838	0.0150	4.18E-04	4.02E-02	0.273	4.618	1.786	0.585	0.118
P_{11}	0.0159	0.0938	0.0872	0.0707	0.1529	0.2324	0.3471	1.57E-04	2.96E-02	0.050	1.465	1.162	0.095	0.057
P_*	0.2614	0.0669	0.1068	0.0649	0.0928	0.2173	0.1904	5.13E-04	3.04E-02	0.188	3.479	1.474	**0.898**	**0.127**

Note: P_5, P_6, P_7, and P_* are NSGA-portfolio, NSGA-II-portfolio, NSGA-III-portfolio, and proposed portfolio. P_8, P_9, P_{10}, P_{11}, and P_* are PSO-portfolio, ICA-portfolio, SPEA2-portfolio, MOGOA-portfolio and proposed portfolio.

first group to earn income, and the other two groups were the control groups. The specific grouping results are shown in Table 3. **Remark:** Through comparison of different models, we used CNN-BiLSTM for making trend predictions about stocks. Finally, 7 stocks were classified into Group A for subsequent portfolio optimization.

5.3. Portfolio optimization

Regarding optimization, different optimization algorithms were used to optimize the weights of stocks. The parameters in these algorithms are shown in Appendix J.

5.3.1. Comparison with classic portfolios

Classic portfolio mainly refers to portfolios that have achieved good results and have been verified by many scholars (e.g., M−V portfolio). Now, we defined P_1 as the M−V, P_2 as the M−V−S, P_3 as the M−V−S−K, and P_4 as the same weight portfolio; P_* was the proposed portfolio. Furthermore, in this comparison, aiming to ensure the accuracy of the verification results, we unified the optimization algorithm as improved NSGA-III. In order to make the results more stable while preserving the emergence of new species, we set the crossover probability $Pc = 0.7$, mutation probability $Pm = 0.3$, and $Mr = 0.02$. Furthermore, N_{off} and N_{mut} were set to 36 and 20, respectively, through the grid search method. The results are shown in Table 4.

Analyzing the optimization performance, the Sharpe ratio and Information ratio of P_* were found to be the highest among the five portfolios, with value of 2.572 and 0.397 respectively in Group A; P_* performed better in the field of access return and system risk. In Groups B and C, even though metrics such as **SR** and **IR** decreased, P_* was still the best performer; therefore, it can be concluded that the proposed portfolio was more efficient than several traditional portfolios. As for P_*, the weights of 600031, 600036, 600196, 600519, 600585, 600690, and 600809 in Group A were 0.1265, 0.1877, 0.1113, 0.1431, 0.1285, 0.1185, and 0.1869, respectively. The results were similar to P_4 mainly because, after stock selection, no stocks had poor returns. Conversely, in Group B and C, P_* allocated weights below 0.1, indicating their unsuitability for investment. This reflects the effectiveness of trend prediction based on deep learning.

5.3.2. Comparison with Genetic-Portfolios

When the same objective functions exist in various portfolios, different optimization algorithms will have a direct impact on the final results. Although improved NSGA-III was used in this paper, many other genetic algorithms have been used in other portfolio optimization problems. Therefore, we defined P_5 as the portfolio optimized by GA, P_6 as the portfolio optimized by NSGA-II, and P_7 as the portfolio optimized by NSGA-III; P_* was also our proposed portfolio. These results can be seen in Table 5.

Taking the results in Group A as an example, $\mathbf{SR}_{P_*} = 2.572 > \overrightarrow{\mathbf{SR}_{P_i}}, i = 5,6,7$, which means that in order to take the total risk of 1, P_* can generate the excess return of 2.572, which is much higher than the remaining portfolios. $\mathbf{IR}_{P_*} = 0.397 > \overrightarrow{\mathbf{IR}_{P_i}} = [-0.097, 0.296, 0.275], i = 5,6,7$, which shows that P_5 is not as good as P_1, and P_6; furthermore, P_7 is more profitable than P_1, but P_* has the highest return compared to P_1. Similar conclusions are drawn in Groups B and C. Therefore, we can conclude that among various genetic-portfolios, the selected portfolio showed the best performance after improved NSGA-III optimization was applied.

5.3.3. Comparison with Meta-heuristic-Portfolios

In addition to some genetic algorithms, many *meta*-heuristic optimization algorithms can be used in portfolio optimization; therefore, we considered some portfolios optimized by Particle Swarm Optimization (PSO), Imperialist Competitive Algorithm (ICA), SPEA2, and Multi-objective Grasshopper Optimization Algorithm (MOGOA) and defined them as P_8, P_9, P_{10}, and P_{11}, respectively. P_* was the portfolio optimized by improved NSGA-III. The detailed experimental results are shown in Table 5.

PSO, ICA, and MOGOA are typical swarm intelligence optimization algorithms, which cannot have sophisticated internal designs; however, they are robust, stable, and adaptive based on simple individuals and rules. Conversely, SPEA2 is an evolutionary algorithm with a fixed archive size; it uses proximity estimation to make the search more accurate. For these four algorithms, we set archive maximum size and maximum iterations equal to 50 and 500. Moreover, regarding specific parameters, we set $Iw = 0.99$, $P_\kappa = 2$, and $G_\kappa = 2$ in PSO, and $Ne = 5$, $Cc = 0.1$, and $Sp = 1$ in ICA. However, these four algorithms did not optimize the portfolios for higher returns. For example, in Group A, the **SR** and **IR** of these portfolios were as follows: $\overrightarrow{\mathbf{SR}_{P_i}} = [2.389, 1.238, 1.910, 2.415]$ and $\overrightarrow{\mathbf{IR}_{P_i}} = [0.073, -0.186, 0.175, 0.218], i = 8, \ldots, 11$, which is far below the metrics of P_*.

5.3.4. Comparison with state-of-the-art portfolios

In the previous subsection, we examined classical, genetic, and *meta*-heuristic portfolios, identifying that the proposed portfolio utilizing the improved NSGA-III demonstrated superior performance. This section will detail how we found the current state-of-the-art (SOTA) portfolios and compared them with the proposed portfolio.

The first SOTA model [13], named P_{12}, extended the classical M−V portfolio by maximizing modified Sharpe ratio with cardinality, box, and budget constraints, and it proposed a level-based learning swarm optimizer (LLSO) [43] to solve this optimization problem. The second SOTA model [44], named P_{13}, developed a hybrid genetic algorithm to solve the problem with the objective of expected return and skewness maximization as well as mean absolute semi-deviation minimization. The next SOTA model [45], named P_{14}, proposed a co-evolutionary multi-swarm adaptive differential evolution algorithm to minimize the Mean-Conditional Value-at-Risk

(MCVaR) and solve the constrained optimization. The last SOTA model [46], named P_{15}, used AdaBoost algorithm to forecast stock values for the next periods and then used mean-VaR to build the portfolio. In order to make the comparison results more reasonable, we used the stocks in Groups A, B, and C and did not use the prediction stage in P_{15} to select the stocks. The above-mentioned four SOTA models have previously proved that they can achieve sufficiently high returns; their performance on the data of this current study is recorded in Table 6.

Table 6 reveals several key findings: (1) These four SOTA models can achieve high returns, with the exception of the P_{15} is negative; (2) P_* performs better than the other four models; (3) All the models perform best in Group A, followed by Group C and least effectively in Group B.

Therefore, several interesting conclusions can be drawn: (1) P_{15} is the only portfolio model that has not been optimized by the optimization algorithm, so the optimization algorithm is extremely important for the portfolio; (2) These four SOTA models fail to utilize data from listed companies or to fully leverage high-order moments, highlighting the importance of constructing the objective function; (3) Among several optimization algorithms, the improved NSGA-III stands out as the most effective.**Remark:** Several classic portfolios, genetic-portfolios, *meta*-heuristic-portfolios, and SOTA portfolios are compared to our proposed portfolios. Evidence suggests that our proposed portfolio strategy yields the highest profits. Figs. 3 and 4 depict the allocation weights and performance metrics of each portfolio.

6. Discussion

Initially, the efficacy of stock classification requires test. Subsequently, to delve deeper into the hybrid system's performance, this section explores various methodologies for evaluating robustness and efficiency, including sensitivity analysis, computational time analysis, and algorithm complexity analysis. Moreover, the potential misestimation of credit risk by the KMV model is also examined.

6.1. Statistical test

In order to test whether the classification results of CNN-BiLSTM are significantly different from those of other models, we conduct two statistical tests based on two metrics: ***Accuracy*** and ***F1-score***. First, we use a statistical framework based on non-parametric Friedman test described by Demsar [47], which aims to test whether there are significant differences between all classification models. The Friedman statistic can be defined as:

$$\tau_{\chi^2} = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4}\right), \tag{27}$$

where r_i denotes the rank of the performance measures of algorithms $i = 1, 2, \ldots, 8$ over 21 stocks, so k and N are equal to 8 and 21. The Friedman statistic obeys a chi-square distribution of $k-1$ degrees of freedom under the null hypothesis, and the null hypothesis in-

Table 6
Metrics of return by comparing with SOTA models.

Group A	SR	IR
P_{12}	1.739	0.077
P_{13}	2.193	0.336
P_{14}	1.849	0.211
P_{15}	1.037	−0.105
P_*	2.572	0.397

Group B	SR	IR
P_{12}	0.274	0.063
P_{13}	0.282	0.066
P_{14}	0.173	0.047
P_{15}	−0.085	−0.021
P_*	0.313	0.098

Group C	SR	IR
P_{12}	0.739	0.106
P_{13}	0.632	0.079
P_{14}	0.481	0.040
P_{15}	0.608	−0.083
P_*	0.898	0.127

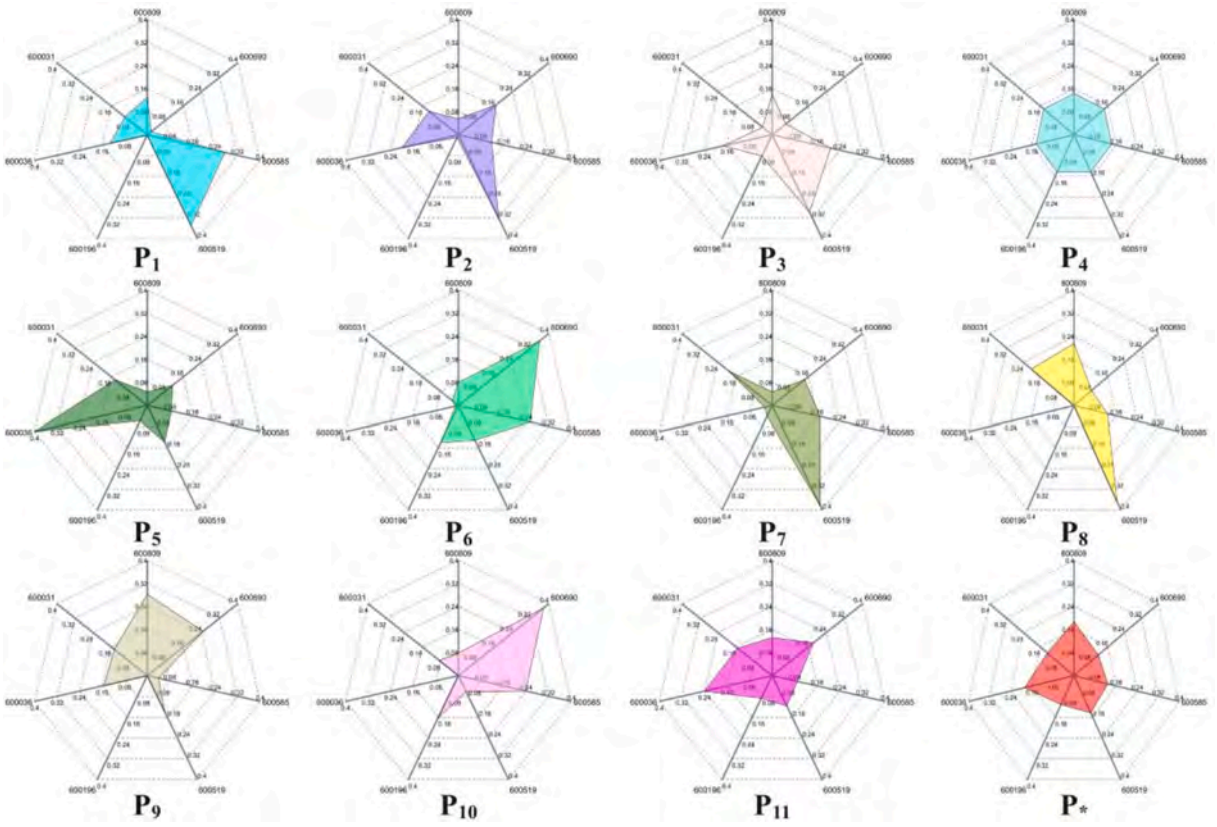

Fig. 3. The weights of each portfolio.

dicates that there is no difference in the results of all algorithms.

The Friedman statistic equals 155.42 (p-value = 0.000) based on ***Accuracy*** ranks and 114.58 (p-value = 0.000) based on ***F1-score*** ranks, which indicates that there are significant differences in terms of ranks. Then, the Wilcoxon signed-ranks test [48] is further used to test whether CNN-BiLSTM differs from the rest of the models. Table 7 presents the p-value of the results of Wilcoxon signed-ranks test; and all p-values [49] are far lower than the significance level 0.1, 0.05, and 0.01. Therefore, the null hypothesis is rejected, which demonstrates that the proposed CNN-BiLSTM and other classification models are significant different. The above can prove the robustness of the proposed CNN-BiLSTM.

6.2. Sensitivity analysis

6.2.1. Parameter sensitivity analysis

In the hybrid system, optimization represents the pivotal process, hence, considerable emphasis was placed on the key parameters of the improved NSGA-III. Furthermore, three hyperparameters were used for sensitivity analysis, including the maximum iterations $\wp$, the population size $\Im$, and the division of reference points n_d. Then, we combined the best values of these hyperparameters to set 200, 300, 400, 500, and 600 for $\wp$, 30, 40, 50, 60, and 70 for $\Im$ and 8, 9, 10, 11, and 12 for n_d. The standard deviation of changes in metrics **SR** and **IR** is shown in Table 8.

As shown in Table 8, most of the values in this table were between 0.01 and 0.05; this indicates that the algorithm was not particularly sensitive to the core parameters. Therefore, it further illustrates the stability and robustness of our proposed system. Specifically, we could also draw the following conclusion: the indicator **IR** was less sensitive than **SR**; n_d was the one that had the least impact on the final result in the three key parameters; the value in Group A was less sensitive than that in Groups B and C.

6.2.2. Data sensitivity analysis

Sensitivity analysis, leveraging varying data sets, elucidates the potential fluctuation in final outcomes in response to data alterations. In this study, the three datasets of Groups A, B, and C in this research were quite different, representing three trends in which prices were likely to rise, fall, or remain unchanged in the future, so we could not directly compare the three datasets horizontally. To address this, a longitudinal comparison methodology was employed for data sensitivity analysis. Specifically, in the previously constructed portfolio optimization, data from Jan. 4, 2016 to Mar. 31, 2022 were adopted, and the training set and test set were divided. The training set was used for determining the weight of each stock, and the test set was used for outputting the metrics. Furthermore,

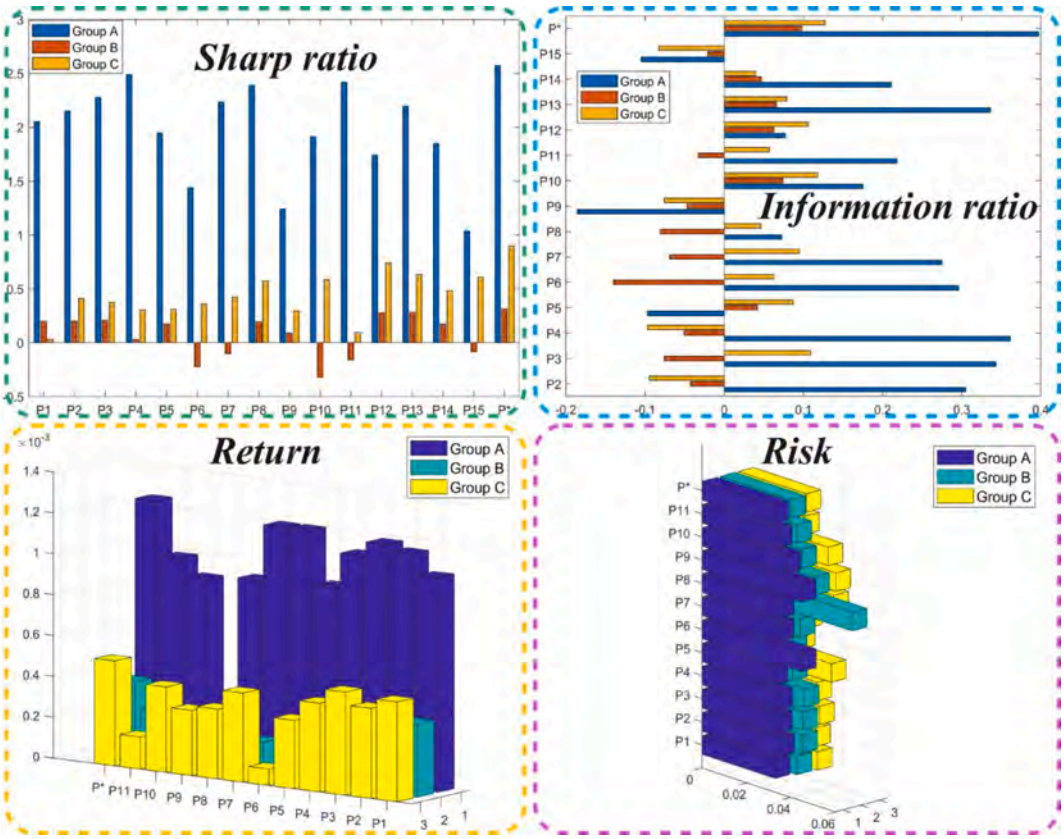

Fig. 4. The metrics of each portfolio.

the data from Mar. 31, 2022 to Mar. 31, 2023 were used for validating the results in this study as an entirely new test set. Appendix K records the SR and IR for all portfolio models in the new test set.

Appendix K Delineates the Sharpe and information ratios across models, aligning with Section 5's experimental outcomes. It is evident that P_* excels within Group A, showcasing superior metric values, thus affirming its robustness even under the situation of new data. While certain models in groups B and C exhibited enhanced SR or IR compared to P_*, considering the stocks to be invested had to belong to Group A, we did not have to pay too much attention to the metrics of groups B and C. Even so, no model could achieve both **SR** and **IR** ahead of P_*. A remarkable observation was the consistently negative IR in models P_4, P_5, and P_{15} across all datasets, underscoring their inferiority to the M−V portfolio. Consequently, when subjected to a new data set, the proposed portfolio model P_* was still found to be optimal

6.3. Computational time analysis

In addition to focusing on the effectiveness of the model, computational time is also a critical metric, reflecting the efficiency of models. Although the models' performance has been previously elaborated, a comparative analysis of their running times is essential. Table 9 records the computational time of each model.

As shown in the table above, there is no doubt that P_4 demonstrated the shortest execution time, but it had the worst performance. Afterward, the running time of the three portfolios P_1, P_2, and P_3 were found to be relatively short, and $P_1 < P_2 < P_3$, primarily due to their objective functions comprising fewer than five objectives. In particular, we observed that the computational time grew longer as the number of objectives increased; this conforms to the pattern of the optimization algorithm. In the improved NSGA-III, the complexity increased exponentially as the number of objectives increased by one. The complexity issue is discussed in detail in Section 6.4. According to Tables 3 and 4, the performance of P_7, P_8, and P_{11} is close to P_*, and their running times are not much different. Specifically, the running times of these portfolios are sorted as follows: $P_{11} > P_* > P_7 > P_8$. The short running time of P_7 is attributed to the absence of chaotic mapping during population initialization; which, in comparison to P_*, results in a shorter overall running time. P_8 and P_{11} both employing swarm-based optimization algorithms, exhibit distinct operational efficiencies, which is mainly because of the single-objective of P_8 and the multi-objective of P_{11}. Therefore, taking all these factors into consideration, we believe that the computational time of the portfolio of P_* is reasonable.

Table 7
The Wilcoxon signed-ranks test.

Algorithm role	Algorithm	Metric	
		Accuracy	***F1-score***
Control	Convolutional Neural Network-Bi-directional Long Short-Term Memory (CNN-BiLSTM)	–	–
Benchmarks	Support Vector Classification (SVC)	4.0151*** (0.000)	4.0151*** (0.000)
	K-Nearest Neighbor (KNN)	4.0160*** (0.000)	4.0151*** (0.000)
	Naive Bayes (NB)	4.0148*** (0.000)	4.0151*** (0.000)
	Decision Tree (DT)	4.0148*** (0.000)	4.0151*** (0.000)
	Back Propagation Neural Network (BPNN)	4.0169*** (0.000)	4.0151*** (0.000)
	Long Short-Term Memory (LSTM)	4.0151*** (0.000)	4.0151*** (0.000)
	Bi-directional Long Short-Term Memory (BiLSTM)	3.9801*** (0.000)	3.9804*** (0.000)
	Convolutional Neural Network (CNN)	4.0160*** (0.000)	4.0151*** (0.000)
	Transformer based on self-attention (Transformer_1)	4.0148*** (0.000)	4.0151*** (0.000)
	Transformer based on encoding (Transformer_2)	4.0152*** (0.000)	3.9804*** (0.000)
	Generative Adversarial Network (GAN)	4.0160*** (0.000)	3.9803*** (0.000)
	Graph Neural Network (GNN)	3.9802*** (0.000)	4.0151*** (0.000)

Note: *** indicates significance on 99% level. The *p*-value for Wilcoxon signed-ranks test is shown between brackets.

Table 8
The sensitivity analysis of improved NSGA-III in hybrid system.

Hyperparameters	Dataset	SR	IR
$\wp$	Group A	**0.0423**	**0.0204**
	Group B	0.0579	0.0273
	Group C	0.0503	0.0249
$\Im$	Group A	**0.0359**	**0.0199**
	Group B	0.0376	0.0245
	Group C	0.0418	0.0270
n_d	Group A	0.0267	**0.0095**
	Group B	**0.0234**	0.0130
	Group C	0.0304	0.0147

Table 9
The results of computational time of all models (s).

Model	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
Group A	81.8	124	205	**1.35**	462	486	500	319
Group B	75.9	137	214	**1.27**	489	503	495	304
Group C	86.4	119	232	**1.16**	443	489	496	337

Model	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_*
Group A	423	541	673	389	531	156	189	507
Group B	405	516	589	376	507	176	201	496
Group C	411	537	634	393	525	147	193	512

6.4. Algorithm complexity analysis

The efficiency of a system's algorithms dictates its computational time, with high algorithmic complexity often leading to reduced efficiency and stability. This paper delves into the time complexity of improved NSGA-III, which is the core of portfolio optimization. Through the examination of its time complexity, this research aims to promote its practical application and ensure its viability over time.

For a detailed analysis, the NSGA-III algorithm was segmented into four distinct modules: Module 1 (Generation), Module 2 (Normalize), Module 3 (Associate), and Module 4 (Niching).

In Module 1, the nondominated sorting of a population of size $2N$ with M-dimensional vectors required the complexity of $O(N \cdot \log^{M-2} \cdot N)$. In Module 2, it took the cost of $O(M \cdot N)$ to identify ideal points, $O(M \cdot N)$ to perform objectives translation, $O(M^2 \cdot N)$ to identify extreme points, and $O(M^3)$ to determine interceptions to find the inverse matrix of a matrix of size $M \times M$; finally, we needed $O(N)$ complexity for at least $2N$ population regularization. In Module 3, all operations that were associated with up to $2N$ group members with H reference points required $O(M \cdot N \cdot H)$ computations. In Module 4, $O(H)$ comparisons were required when niching was performed, but $O(L)$ complexity was required if a worst-case scenario occurred. In the niching algorithm, the calculation was performed at most L times, so it required $O(L^2)$ or $O(LH)$ costs of computation. In all of our analysis, we assumed that N was greater than or equal to H. When using the tent chaotic mapping algorithm to improve NSGA-III, we considered the complexity of $O(a \cdot M)$, where a is the number of chaotic iterations.

Based on the above-mentioned calculations and analyses, the worst-case complexity of improved NSGA-III was found to be $O(N^2 \cdot \log^{M-2} \cdot N)$ or $O(N^2 \cdot M)$, whichever was larger.

6.5. KMV mis-estimate credit Risk?

The KMV method, a popular tool for estimating credit risk, may produce inaccurate measurements of credit, resulting in underestimations during bull markets and overestimations during bear markets. Therefore, this chapter aimed to investigate whether our proposed portfolio model could outperform other comparison models in addressing this issue. To achieve this goal, we partitioned the dataset into two subgroups corresponding to bear and bull markets, respectively, and conducted experiments on each subset. According to the movement of the Shanghai Composite Index, we defined the stock market from Jan. 4, 2020 to Feb. 26, 2021 as a bull market and the market from Oct. 29, 2021 to Mar. 31, 2022 as a bear market. The evaluation focused on two key performance indicators: the SR and IR, with detailed comparisons presented in Appendix L.

Based on the findings presented here, it can be inferred that P_* (the proposed portfolio model) outperformed other models in both the bear and bull markets. It is commonly acknowledged that most models exhibit superior performance during bull markets in comparison to bear markets. Nonetheless, P_* and some benchmark models demonstrated the capacity to generate considerable positive returns even in a bear market. This is particularly noteworthy, as the returns exceeded those provided by the risk-free interest rate and the M−V portfolio model. In conclusion, the possible defects of the KMV had little impact on the results of this study.

7. Conclusion

Portfolio management represents a critical aspect of investment decision-making in the real world, with investors universally seeking portfolios that offer high returns at low risk levels. This study introduced a hybrid system that encompasses both stock selection and portfolio optimization to address these investment challenges.

- **Stock Selection:** The study innovatively integrated a convolutional neural network with a bi-directional recurrent neural network to develop a CNN-BiLSTM model for trend classification prediction. Numerical experiments demonstrated that CNN-BiLSTM outperformed numerous traditional classification models and neural network models, showcasing its efficacy in accurately predicting stock trends.
- **Portfolio Optimization:** A novel five-objective optimization problem was formulated, incorporating mean, variance, skewness, kurtosis, and distance-to-default as objectives. The improved NSGA-III algorithm was applied to solve this complex problem. Compared with traditional, genetic, and heuristic portfolios, the proposed portfolio achieved higher returns and lower risks, indicating its superior investment performance.
- **Robustness and Efficiency:** Through sensitivity analysis and computational time analysis, the proposed portfolio was shown to exhibit greater robustness and efficiency than competing portfolio strategies, enhancing its appeal to investors.

The proposed hybrid system stands out for its ability to mitigate investment risks while enhancing returns, offering a promising framework for future research and practical applications. It can be applied not only to the stocks in SSE 50 but also to a broader range of stocks. However, the system has certain limitations that suggest avenues for future research:

- **Predictive Inputs:** The current model primarily focuses on stock-related data, overlooking potentially influential external factors such as market sentiment and public opinion, which could enhance the accuracy of stock trend predictions.

- **Optimization Efficiency:** The efficiency of the portfolio optimization algorithm presents opportunities for improvement, suggesting that future efforts could focus on either selecting more effective optimization algorithms or significantly enhancing existing ones.

CRediT authorship contribution statement

Mengzheng Lv: Visualization, Writing – original draft, Software. **Jianzhou Wang:** Funding acquisition, Supervision, Visualization. **Shuai Wang:** Formal analysis, Methodology. **Jialu Gao:** Visualization. **Honggang Guo:** Software.

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.

Acknowledgements

This work was supported by the Major Program of National Fund of Philosophy and Social Science of China (grant number 17ZDA093).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ins.2024.120549.

References

[1] A. Dezhkam, M.T. Manzuri, Forecasting stock market for an efficient portfolio by combining XGBoost and Hilbert-Huang transform, Eng. Appl. Artif. Intel. (2023), https://doi.org/10.1016/j.engappai.2022.105626.
[2] Y. Chen, X. Zhao, J. Hao, A novel MOPSO-SODE algorithm for solving three-objective SR-ES-TR portfolio optimization problem, Expert Syst. Appl. (2023), https://doi.org/10.1016/j.eswa.2023.120742.
[3] H. Markowitz, Portfolio selection Harry Markowitz, J. Financ. (1952).
[4] D. Maringer, P. Parpas, Global optimization of higher order moments in portfolio selection, J. Glob. Optim. (2009), https://doi.org/10.1007/s10898-007-9224-3.
[5] L. Liu, A new foundation for the mean-variance analysis, Eur. J. Oper. Res. 158 (2004) 229–242, https://doi.org/10.1016/S0377-2217(03)00301-1.
[6] P.A. Samuelson, The fundamental approximation theorem of portfolio analysis in terms of means, variances and higher moments, Rev. Econ. Stud. (1970), https://doi.org/10.2307/2296483.
[7] F. Zhen, J. Chen, A closed-form mean–variance–skewness portfolio strategy, Financ. Res. Lett. (2022), https://doi.org/10.1016/j.frl.2022.102933.
[8] I. Abid, C. Urom, J. Peillex, M. Karmani, G. Ndubuisi, PGP for portfolio optimization: application to ESG index family, Ann. Oper. Res. (2023), https://doi.org/10.1007/s10479-023-05460-w.
[9] X. Li, B. Li, T. Jin, P. Zheng, Uncertain random portfolio optimization with non-dominated sorting genetic algorithm-II and optimal solution criterion, Artif. Intell. Rev. (2023), https://doi.org/10.1007/s10462-022-10388-x.
[10] P. Crosbie, Modeling default risk - modeling methodology, Moodýs KMV. (2003).
[11] J. Wang, H. Zhang, H. Luo, Research on the construction of stock portfolios based on multiobjective water cycle algorithm and KMV algorithm, Appl. Soft Comput. 115 (2022) 108186, https://doi.org/10.1016/j.asoc.2021.108186.
[12] M. Brandtner, W. Kürsten, R. Rischau, Beyond expected utility: Subjective risk aversion and optimal portfolio choice under convex shortfall risk measures, Eur. J. Oper. Res. (2020), https://doi.org/10.1016/j.ejor.2020.02.040.
[13] M. Kaucic, F. Piccotto, G. Sbaiz, G. Valentinuz, A hybrid level-based learning swarm algorithm with mutation operator for solving large-scale cardinality-constrained portfolio optimization problems, Inf. Sci. 634 (2023) 321–339, https://doi.org/10.1016/j.ins.2023.03.115.
[14] R. Bedoui, R. Benkraiem, K. Guesmi, I. Kedidi, Portfolio optimization through hybrid deep learning and genetic algorithms vine Copula-GARCH-EVT-CVaR model, Technol. Forecast. Soc. Chang. 197 (2023) 122887, https://doi.org/10.1016/j.techfore.2023.122887.
[15] J. Wang, S. Wang, M. Lv, H. Jiang, Forecasting VaR and ES by using deep quantile regression, GANs-based scenario generation, and heterogeneous market hypothesis, Fin. Innov. 10 (2024), https://doi.org/10.1186/s40854-023-00564-5.
[16] Z.X. Zhang, W.N. Chen, X.M. Hu, A knowledge-based constructive estimation of distribution algorithm for bi-objective portfolio optimization with cardinality constraints [Formula presented], Appl. Soft Comput. (2023), https://doi.org/10.1016/j.asoc.2023.110652.
[17] B. Li, X. Li, K.L. Teo, P. Zheng, A new uncertain random portfolio optimization model for complex systems with downside risks and diversification, Chaos Solitons Fractals (2022), https://doi.org/10.1016/j.chaos.2022.112213.
[18] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput. 18 (2014) 602–622, https://doi.org/10.1109/TEVC.2013.2281534.
[19] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I : Solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (2014) 577–601, https://doi.org/10.1109/TEVC.2013.2281535.
[20] M. Dhaini, N. Mansour, Squirrel search algorithm for portfolio optimization, Expert Syst. Appl. (2021), https://doi.org/10.1016/j.eswa.2021.114968.
[21] A. Corberán-Vallet, E. Vercher, J.V. Segura, J.D. Bermúdez, A new approach to portfolio selection based on forecasting, Expert Syst. Appl. 215 (2023) 119370, https://doi.org/10.1016/j.eswa.2022.119370.
[22] K. Vasantha Lakshmi, K.N. Udaya Kumara, A novel randomized weighted fuzzy AHP by using modified normalization with the TOPSIS for optimal stock portfolio selection model integrated with an effective sensitive analysis, Expert Syst. Appl. 243 (2024) 122770, https://doi.org/10.1016/j.eswa.2023.122770.

[23] H. Peng, K. Dong, J. Yang, Stock Price movement prediction based on relation type guided graph convolutional network, Eng. Appl. Artif. Intel. (2023), https://doi.org/10.1016/j.engappai.2023.106948.
[24] X. Ma, X. Li, W. Feng, L. Fang, C. Zhang, Dynamic graph construction via motif detection for stock prediction, Inf. Process. Manag. (2023), https://doi.org/10.1016/j.ipm.2023.103480.
[25] Y. Han, J. Kim, D. Enke, A machine learning trading system for the stock market based on N-period min-max labeling using XGBoost, Expert Syst. Appl. (2023), https://doi.org/10.1016/j.eswa.2022.118581.
[26] W. Ye, J. Yang, P. Chen, Short-term stock price trend prediction with imaging high frequency limit order book data, Int. J. Forecast. (2023), https://doi.org/10.1016/j.ijforecast.2023.10.008.
[27] M. Lu, X. Xu, TRNN: An efficient time-series recurrent neural network for stock price prediction, Inf. Sci. 657 (2024) 119951, https://doi.org/10.1016/j.ins.2023.119951.
[28] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE (1998), https://doi.org/10.1109/5.726791.
[29] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. (1997), https://doi.org/10.1109/78.650093.
[30] S. Manaster, G. Koehler, The calculation of implied variances from the black-scholes model: A note, J. Financ. (1982), https://doi.org/10.2307/2327127.
[31] R. Matthews, On the derivation of a "chaotic" encryption algorithm, Cryptologia (1989), https://doi.org/10.1080/0161-118991863745.
[32] Y. Wang, D. Gao, L. Yu, W. Lei, M. Feiszli, M.Z. Shou, GEB+: A benchmark for generic event boundary captioning, grounding and text-based retrieval, (2022).
[33] J. Wang, M. Lv, Z. Li, B. Zeng, Multivariate selection-combination short-term wind speed forecasting system based on convolution-recurrent network and multi-objective chameleon swarm algorithm, Expert Syst. Appl. (2023), https://doi.org/10.1016/j.eswa.2022.119129.
[34] A. Homaifar, C.X. Qi, S.H. Lai, Constrained optimization via genetic algorithms, Simulation 62 (1994) 242–254, https://doi.org/10.1177/003754979406200405.
[35] W. Sun, X. Zhang, M. Li, Y. Wang, Interpretable high-stakes decision support system for credit default forecasting, Technol. Forecast. Soc. Chang. 196 (2023) 122825, https://doi.org/10.1016/j.techfore.2023.122825.
[36] P. Jiang, Z. Liu, M.Z. Abedin, J. Wang, W. Yang, Q. Dong, Profit-driven weighted classifier with interpretable ability for customer churn prediction, Omega (United Kingdom). 125 (2024) 103034, https://doi.org/10.1016/j.omega.2024.103034.
[37] J. Gao, J. Wang, Y. Zhou, M. Lv, D. Wei, Enhancing investment performance of Black-Litterman model with AI hybrid system: Can it be done? Expert Syst. Appl. (2024).
[38] T.K. Lee, S.Y. Sohn, Alpha-factor integrated risk parity portfolio strategy in global equity fund of funds, Int. Rev. Financ. Anal. 88 (2023) 102654, https://doi.org/10.1016/j.irfa.2023.102654.
[39] N. Vafai, D. Rakowski, The sources of portfolio volatility and mutual fund performance, Int. Rev. Financ. Anal. 91 (2024) 102985, https://doi.org/10.1016/j.irfa.2023.102985.
[40] N.S. Vaswani, N. Ashish, J. Parmar, L. Uszkoreit, A.N. Jones, Ł.K. Gomez, Illia polosukhin, attention is all you need, Adv. Neural Inf. Proces. Syst. (2017).
[41] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Proces. Syst. (2014), https://doi.org/10.1007/978-3-658-40442-0_9.
[42] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. (2009), https://doi.org/10.1109/TNN.2008.2005605.
[43] Q. Yang, W.N. Chen, J. Da Deng, Y. Li, T. Gu, J. Zhang, A level-based learning swarm optimizer for large-scale optimization, IEEE Trans. Evol. Comput. 22 (2018) 578–594, https://doi.org/10.1109/TEVC.2017.2743016.
[44] M.K. Mehlawat, P. Gupta, A.Z. Khan, Portfolio optimization using higher moments in an uncertain random environment, Inf. Sci. (2021), https://doi.org/10.1016/j.ins.2021.03.019.
[45] Y. Song, G. Zhao, B. Zhang, H. Chen, W. Deng, W. Deng, Engineering applications of artificial intelligence an enhanced distributed differential evolution algorithm for portfolio optimization problems, Eng. Appl. Artif. Intel. 121 (2023) 106004, https://doi.org/10.1016/j.engappai.2023.106004.
[46] J. Behera, A.K. Pasayat, H. Behera, P. Kumar, Prediction based mean-value-at-risk portfolio optimization using machine learning regression algorithms for multi-national stock markets, Eng. Appl. Artif. Intel. 120 (2023) 105843, https://doi.org/10.1016/j.engappai.2023.105843.
[47] J. Demśar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. (2006).
[48] K. Krishnamoorthy, Wilcoxon signed-rank test, Handb. Stat. Distribut. Appl. (2020), https://doi.org/10.1201/9781420011371-34.
[49] K.W. De Bock, A. De Caigny, Spline-rule ensemble classifiers with structured sparsity regularization for interpretable customer churn modeling, Decis. Support Syst. 150 (2021) 113523, https://doi.org/10.1016/j.dss.2021.113523.