Expert Systems With Applications 255 (2024) 124555

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Two-stage stock portfolio optimization based on AI-powered price prediction and mean-CVaR models

Chia-Hung Wang [a,b,*], Yingping Zeng [a], Jinchen Yuan [a]

[a] College of Computer Science and Mathematics, Fujian University of Technology, No. 69, Xuefu South Road, University Town, Fuzhou City, Fujian Province, 350118, China

[b] Fujian Provincial Key Laboratory of Big Data Mining and Applications, No. 69, Xuefu South Road, University Town, Fuzhou City, Fujian Province, 350118, China

ARTICLE INFO

Keywords:
Artificial intelligence
Portfolio optimization
Mean-Conditional Value-at-Risk
Stock price prediction
Savitzky–Golay filter
Sparrow search algorithm
Long short-term memory network

ABSTRACT

With the advancement of prediction methods in the field of artificial intelligence, accurate price predictions can effectively support financial portfolio selection. This paper proposes an intelligent stock portfolio selection method based on a prediction neural network, incorporating signal processing and hyperparameter optimization techniques. The method is divided into two key stages: stock price prediction and portfolio selection. In the first stage, we apply Savitzky–Golay filtering to denoise price data and reveal its patterns, and optimize the hyperparameters of the long short-term memory network using the sparrow search algorithm to achieve high-precision stock price predictions. In the second stage, we use the mean-Conditional Value-at-Risk (mean-CVaR) model to select the optimal stock allocation, considering factors such as potential returns, prediction accuracy, and growth rate. Numerical comparisons based on multiple public financial datasets demonstrate that the proposed two-stage method significantly outperforms seven benchmark methods. Specifically, on the Shanghai and Shenzhen 300 (CSI 300) Index dataset, the proposed method achieves a determination coefficient of 0.9980 and an accuracy rate of 97.05%. Additionally, its cumulative returns reach 9.38%, 8.63%, and 7.54% at different confidence levels.

1. Introduction

With the progress of artificial intelligence technology, accurately predicting stock market trends and constructing effective investment portfolios have become central focuses in investment research. In the Chinese stock market, the rise of financial technology has propelled the widespread application of artificial intelligence and big data (Yang & Ferrer, 2023). Faced with market dynamics and investment risks, improving the accuracy of predictive models is a significant challenge (Wang, Hu, et al., 2023). Currently, traditional predictive models suffer from low prediction accuracy, resulting in insufficiently precise forecasts.

Nowadays, many studies focus solely on either predicting or optimizing investment portfolios, with limited research directly applying prediction results to portfolio optimization. This limitation significantly discounts the practical application value of prediction results in investment decision-making. Advanced predictive models provide comprehensive guidance to investors, enabling them to navigate stock market fluctuations more flexibly. Simultaneously, asset allocation optimization models aim to determine optimal asset allocation strategies (Yu et al., 2023). Integrating precise stock price predictions with portfolio construction contributes to a comprehensive and systematic approach to addressing market challenges. It provides reliable support for investors and creates more favorable conditions for investment returns.

Stock market data is often regarded as time series, characterized by volatility, nonlinearity, non-stationarity, and high noise (Li et al., 2022; Wu et al., 2017). Researchers employ various methods for analysis, including traditional approaches such as moving average and exponential smoothing (Mohammed et al., 2022), as well as classical methods like fourier transform, wavelet transform (Wu et al., 2021), and empirical mode decomposition (Ali et al., 2023). The kalman filter (Dastgerdi & Mercorelli, 2022) can handle noise in both system models and measurement data but is not suitable for nonlinear and non-Gaussian noise. The Savitzky–Golay (SG) filter (Chen et al., 2022), capable of capturing both local features and global trends, effectively suppresses

* Corresponding author at: College of Computer Science and Mathematics, Fujian University of Technology, No. 69, Xuefu South Road, University Town, Fuzhou City, Fujian Province, 350118, China.

E-mail addresses: 61201506@fjut.edu.cn (C.-H. Wang), ypzeng520@hotmail.com (Y. Zeng), jcyuan520@hotmail.com (J. Yuan).

https://doi.org/10.1016/j.eswa.2024.124555
Received 15 January 2024; Received in revised form 17 June 2024; Accepted 19 June 2024
Available online 22 June 2024

high-frequency noise, making it particularly suitable for complex time series data (Cai et al., 2024; Qin et al., 2023). Therefore, this study will employ the SG filte r for smoothing and denoising stock market data.

The prediction of stock market movements poses challenges, as the Efficient Market Hypothesis (Jensen, 1978) posits that stock markets have fully reflected all publicly available information, making it difficult to attain excess profits based solely on public information. However, artificial intelligence methods have demonstrated significant potential in stock prediction (Oyewola et al., 2024). By integrating historical stock data, sentiment analysis of market emotions, and macroeconomic indicators, AI models offer accurate predictions of stock prices (Yan et al., 2024). The primary focus of research lies in machine learning and deep learning models to capture more intricate nonlinear relationships and underlying patterns (Chen et al., 2023). Classical machine learning models such as Support Vector Machine (SVM) (Kurani et al., 2023), Random Forest (RF) (Yin et al., 2023), k-Nearest Neighbors (KNN) (Balazs et al., 2022), and deep learning models including Convolutional Neural Network (CNN) (Chandar, 2022; Wang, Cai, et al., 2023), Recurrent Neural Network (RNN) (Dudukcu et al., 2023), and Long Short-Term Memory Network (LSTM) (Maarif et al., 2023). Comparative studies have found that deep learning models perform better in processing large-scale data and complex tasks (Wang et al., 2024).

Furthermore, the judicious tuning of model hyperparameters is crucial for prediction accuracy (Yousaf et al., 2022). Traditional neural network prediction models often require significant manual effort to adjust hyperparameter values for optimal training results. To overcome this labor-intensive process, metaheuristic optimization algorithms have gained popularity in recent years, including Particle Swarm Optimization (PSO) (Wang, Zhao, & Tian, 2023), Seagull Optimization Algorithm (SOA) (Jia et al., 2019), Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014) and Sparrow Search Algorithm (SSA) (Fan et al., 2023). The SSA simulates the foraging behavior of sparrows and is highly regarded for its excellent search performance and stability. Compared to other optimization algorithms, SSA possesses superior global search capabilities and convergence properties. Hence, this research leverages SSA to optimize LSTM hyperparameters.

Portfolio optimization aims to balance expected returns and risks. Markowits (1952) proposed a Mean-Variance (MV) model that pursues the maximum expected return while limiting the variance or reduces the variance given the minimum expected return. In the MV model, variance is used to measure the risk in the financial asset market, but it does not explicitly consider the trend of volatility (Chennaf & Amor, 2023). Afterward, Value at Risk (VaR) emerged as an alternative risk metric, estimating the maximum loss within a specified range under normal market conditions. Thus, the mean-VaR model began to be applied in portfolio optimization (Kumar et al., 2022). However, VaR has drawbacks such as non-subadditivity, lack of convexity, and not providing detailed information about tail losses (Ran et al., 2024). To address these limitations, Conditional Value at Risk (CVaR) was subsequently introduced as a consistent risk measurement, adhering to principles such as translation invariance, positive homogeneity, and monotonic additivity (Staino et al., 2023). Following that, more and more researchers applied the mean-CVaR (mCVaR) model to portfolio optimization (Abudurexiti et al., 2023). Nevertheless, the mentioned studies predominantly focus on enhancing and extending the MV model, while neglecting the aspects of subsequent asset selection and allocation. Therefore, this study adopts the mCVaR model for portfolio selection, determining the investment proportions for each asset (Wu et al., 2022).

In summary, a novel approach to portfolio construction is presented, incorporating SG filtering and SSA-optimized LSTM predictions with the mCVaR model. The model comprises two main stages: the construction of the portfolio prediction model and the selection of the portfolio. The overall workflow is depicted in Fig. 1. In the first stage, SG filtering is employed for smoothing and denoising stock price data, uncovering potential trends and patterns. Furthermore, a combined SSA and LSTM hybrid model is employed to forecast stock prices for the forthcoming period. In the second stage, factors such as potential returns, prediction accuracy, and the rate of increase are comprehensively considered to select high-quality stocks. The mCVaR model is then employed to allocate the investment proportions for the portfolio.

This method provides financial practitioners with a powerful decision support tool, enabling them to better understand stock price trends and accurately predict future movements. Furthermore, by optimizing risk management through the mCVaR model, it aims to reduce potential losses and enhance investment returns. Hence, this study is poised to improve the investment decision-making abilities of practitioners and strengthen their portfolio management skills. From a theoretical standpoint, this research integrates artificial intelligence technology with financial models, opening up a new avenue for the advancement of portfolio prediction models. By applying cutting-edge technologies such as big data analytics and machine learning to financial market analysis, this study introduces a novel perspective and framework to the field of finance. This not only fosters innovation and progress in financial theory but also offers robust theoretical backing for practical applications in the financial market.

The primary findings and noteworthy facets of this analysis can be summarized as follows:

- This paper introduces the SG-SSA-LSTM, a composite model for stock price prediction. After benchmark testing, SSA is selected to optimize the LSTM network.
- It combines the hybrid prediction model with the mCVaR model, offering an innovative approach to portfolio construction.
- Tested on historical data from the Shanghai and Shenzhen 300 (CSI 300) index, this method outperforms seven benchmark strategies, showcasing its practical efficacy and superior performance.
- Comparisons at different confidence levels, accounting for transaction costs, further validate the effectiveness and applicability of the proposed approach.

The remainder of the paper is structured as follows: In Section 2, we provide a review of relevant studies. The proposed hybrid prediction neural network will be introduced in Section 3. Next, in Section 4, we present a mCVaR portfolio optimization model for intelligent stock portfolio selection. The details about test datasets, our experimental setup and benchmark strategies employed in the experiments are to be presented in Section 5. In Section 6, we will conduct a series of data analysis of the numerical experiments. Finally, we summarize the main findings of this study and several future research directions in Section 7.

2. Related work

In this segment, we explore prior research concerning the utilization of artificial intelligence techniques in models for predicting stock movements and selecting portfolios. Conventional techniques for stock prediction predominantly utilize time-series models such as autoregressive moving average, autoregressive integrated moving average, and similar methodologies. These models are based on historical data and predict future stock prices by analyzing factors such as trends, seasonality and cyclicality. However, they have limitations in handling nonlinear relationships and external factors. Subsequently, machine learning models such as SVM, RF, KNN, and others emerged. Kumbure et al. (2022) studied the machine learning techniques applied to stock market prediction from 2000 to 2019. This research analyzed the markets and stock indices used in predictions, 2173 unique predictor variables, and various machine learning technique variants. Additionally, the most influential studies were identified through bibliometric analysis. Illa et al. (2022) proposed an integrated pattern matching approach combining RF and SVM for the development and evaluation of stock price predictors to optimize the profitability of trading

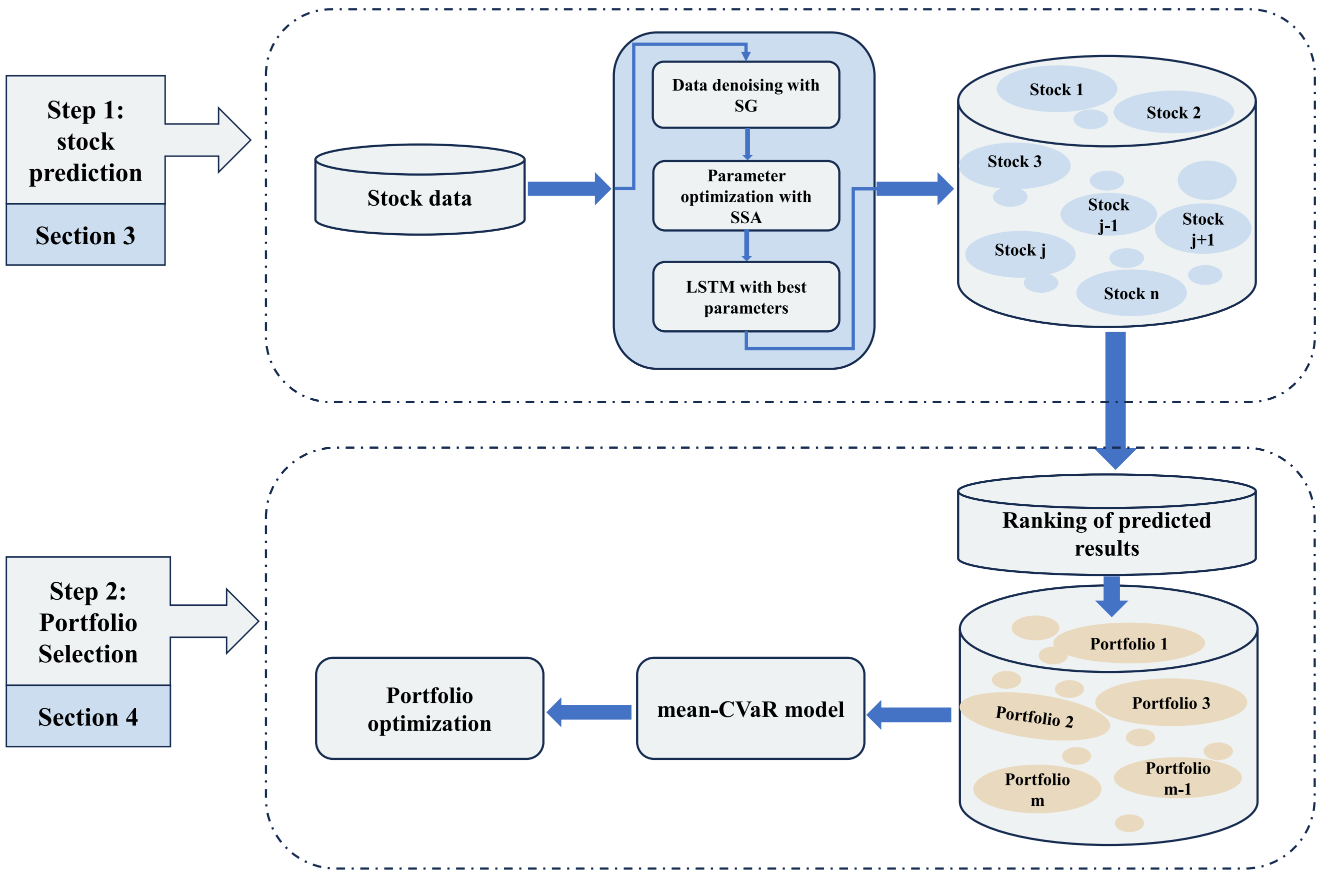

Fig. 1. Overall Process of Two-Stage Portfolio Optimization Model.

strategies.Although this model demonstrates outstanding performance in stock market prediction, it still has some shortcomings in addressing overfitting and sensitivity to data noise.

Therefore, our focus turns to deep learning architectures, encompassing CNN, RNN, LSTM, etc. Jiang (2021) conducted a recent review of deep learning models for stock market prediction, categorizing different data sources, neural network architectures, evaluation metrics, as well as implementation and reproducibility aspects. This study helps researchers understand the latest advancements and facilitates the easy replication of previous studies as baselines. Additionally, it suggests future research directions. This review provides a deeper understanding of the application of deep learning models in the stock market. We also noted that Lu et al. (2021) innovatively introduced CNN, Bidirectional LSTM (BiLSTM), and attention mechanism, constructing the CNN-BiLSTM-AM method. Over 1000 trading days of the Shanghai Composite Index, the study highlights significant improvements in predicting next-day closing stock prices. Bathla et al. (2023) developed an LSTM-based neural network model predicted prices for highly volatile stocks in 2020, outperforming traditional data analysis methods in mean absolute percentage error values. However, these methods did not consider whether the optimization of hyperparameters would lead to more accurate predictive results.

To examine the role of hyperparameter settings in model performance, we investigated the following relevant literature. Zhang et al. (2023) proposed an LSTM network enhanced with features, integrating residue-driven υ support vector regression. Utilizing a genetic algorithm to optimize model parameters, this method ensures not only that the network structure is superior to the pre-set structure but also achieves satisfactory experimental results on real datasets. Mu et al. (2023) utilized the SSA to optimize LSTM hyperparameters. Experimental results demonstrate that this method objectively interprets model parameter settings, leading to improved predictive performance. Liu et al. (2022) introduced an LSTM model optimized with SSA for predicting stock trends. Comparative experiments indicate that this model surpasses traditional stock prediction methods in terms of higher predictive accuracy improved the interpretability of the network model structure and parameters. Test results show that these models have higher prediction accuracy and better stability than single prediction models. However, these methods lack further practical trading and evaluation of the predictive results.

After obtaining accurate stock prices and predicting price movements, we further consider how to allocate shares of assets to achieve outstanding investment returns and effective risk control. Therefore, we refer to the application of optimization models based on accurate predictions in portfolio management. For instance, Ma et al. (2023) employed an LSTM network with autoencoder feature extraction to extract stock return features and combined it with the omega model for portfolio optimization. The experiments demonstrated that the model achieved satisfactory returns. Wang et al. (2020) proposed a combination optimization method based on LSTM networks and the MV model to obtain the optimal average risk-return trade-off. Xie and Wang (2022) constructed a stock portfolio investment strategy based on deep learning neural networks. Using 111 stable stocks as samples, the experimental results revealed optimal levels for five evaluation metrics. However, these methods lack consideration for denoising complex financial data. Therefore, this study specifically focuses on the potential impact of smoothed data on predictive performance.

Du (2022) created an LSTM network with enhanced features tailored for asset selection, then applied the MV model for portfolio construction. The model achieved significant returns on the CSI 300 index and the standard & poor's 500 data. Behera et al. (2023) combined machine learning algorithms with the mean-VaR model. By predicting stock values and optimizing investment portfolios, they verified that the

Table 1
Comparison with the existing literature.

Attribute	Financial market	Data denoising	Hyperparameter optimization	Portfolio selection	Cardinality	Solution approach	Sortino ratio	Stock prediction	Transaction cost	Confidence level
Lu et al. (2021)	SSE	×	×	×	×	CNN-BiLSTM-AM	×	√	×	×
Bathla et al. (2023)	NSE, BSE, NYSE, Dow Jone, Nikkei 225, S&P 500, NASDAQ	×	×	×	×	LSTM	×	√	×	×
Liu et al. (2022)	N225,ADI, 1SS, AAPL, DJI,HIS, IXIC,WTI, S&P 500	×	√	×	×	FEL-vSVR+LSTM	×	√	×	×
Ma et al. (2023)	CSI 100	×	×	√	√	AE-LSTM+omega	×	√	√	×
Wang et al. (2020)	UK 100	×	×	√	√	LSTM+MV	√	√	√	×
Xie and Wang (2022)	CSI 300, SSE 50, CSI 500	×	×	√	×	DL-NN	×	√	×	×
Du (2022)	CSI 300, S&P 500	×	×	√	×	A-LSTM+MV	√	√	×	×
Behera et al. (2023)	BSE,TSE, SSE	×	×	√	√	AdaBoost+mean-VaR	×	√	×	×
Paiva et al. (2019)	Ibovespa	×	×	×	√	SVM+MV	×	√	√	×
Proposed approach	CSI 300	√	√	√	√	SG-SSA-LSTM+mCVaR	√	√	√	√

adaptive boosting-predicted mean-VaR model exhibited optimal performance. Researchers like Paiva et al. (2019) adopted a fusion method based on a machine learning classifier, combining the SVM and MV methods for portfolio selection. The experimental results show that this model provides a potential practical method for stock price prediction. While the aforementioned portfolio models successfully integrate stock prediction and portfolio selection, they do not fully consider investors' risk preferences. Therefore, this study introduces confidence levels to meet investors' varying risk preferences.

After extensive experimental validation, our proposed portfolio prediction model demonstrates significant potential in the 24 individual stocks of the CSI 300 index, exhibiting high-precision predictive performance. Among these stocks, the average fitting accuracy and the accuracy of predicting price movements reached 0.9894 and 0.9366 respectively. This model lays a solid foundation for subsequent integration with portfolio selection models. Considering factors such as portfolio cardinality, confidence levels, and transaction costs, our model, when compared with seven other benchmark models, still achieves the highest cumulative returns. Therefore, our proposed method provides investors with outstanding choices in both stock prediction and portfolio construction.

Compared to existing research on portfolio construction, the approach proposed in this paper is more comprehensive and incorporates richer features. It includes data denoising, hyperparameter optimization, portfolio selection, stock cardinality filtering, transaction costs, and confidence levels, among others. We have conducted a comprehensive comparison of our proposed method with the literature mentioned above, and specific details will be presented in Table 1.

3. A hybrid prediction neural network based on Savitzky–Golay filtering and sparrow search optimization

This section breaks down the proposed predictive methodology. It first introduces how the SG filter smooths the data, followed by a brief overview of the basic structure of the LSTM network. Finally, based on this framework, the hybrid prediction model with the integration of SSA is further elaborated. Additionally, the model parameters and their definitions to be used in Sections 3 and 4 are summarized in Table 2.

3.1. The inclusion of Savitzky–Golay filter

In pursuit of more accurate time series data prediction, smoothing and noise removal are essential. To achieve this objective, this paper employs the SG filter (Zhao et al., 2024). Firstly, a multidimensional time series $P_L = (P_{L1}, P_{L2}, \dots, P_{Ln})$ is obtained, with a one-dimensional time series $P_{Ln} = (p_1, p_2, \dots, p_L)$, of length L. A subsequence of P_{Ln} with a window size of $B = 2a + 1 (a \in Z^+)$ is represented as:

$$\{p_{s-a}, \dots, p_s, \dots, p_{s+a}\}, \quad s \in [a+1, m-a], \tag{1}$$

where P_L represents the fundamental information of the stock market. P_{Ln} denotes the constituent components of the fundamental information (e.g., closing prices). p_L represents the specific daily values of the constituent components. m signifies the number of days for historical observation.

The Kth-order polynomial $F(v)$ defined for fitting the data points within the window is:

$$F(v) = \sum_{i=0}^{K} A_i v^i, \quad v \in [-a, a], \tag{2}$$

where, A_i represents the i-th coefficient of the SG filter.

We employ the method of least squares to minimize the following error:

$$\theta = \sum_{v=-a}^{a} (q(v) - p_{s+v})^2. \tag{3}$$

Subsequently, by computing A_0, we obtain the optimal fitting $q(0)$ for the central point p_s in the window. As the window shifts, each point

Table 2
Definition of mathematical notations used in the proposed method.

Symbol	Implication
P_L	Basic information about the stock market (closing price, opening price, ect)
Q_L	The new sequence obtained after SG smoothing of P_L
T_{origi}	The original input time series
T_{inp}	The new sequence obtained by normalizing T_{origi}
$F(v)$	Fitting a Kth-degree polynomial to the data points within the window
h_t	Predicted stock price normalized values
$\bar{p}_L$	The actual stock prediction value after performing reverse normalization on h_t
pop	The number of sparrow populations
dim	Dimensions of sparrow population
$iter_{max}$	Maximum iterations
X_i	The position of the ith sparrow
$X^t_{i,j}$	Positional information of the ith sparrow in the jth dimension at iteration t
h_1, h_2	Number of hidden layer neurons
dr	Dropout rate of LSTM
bs	Batch-size of LSTM
α	Confidence level
u	Investment weight vector for n assets
R	Expected return rates for n assets
M_0	Maximum tolerable risk level for the investor
r_0	Expected portfolio return of the investor

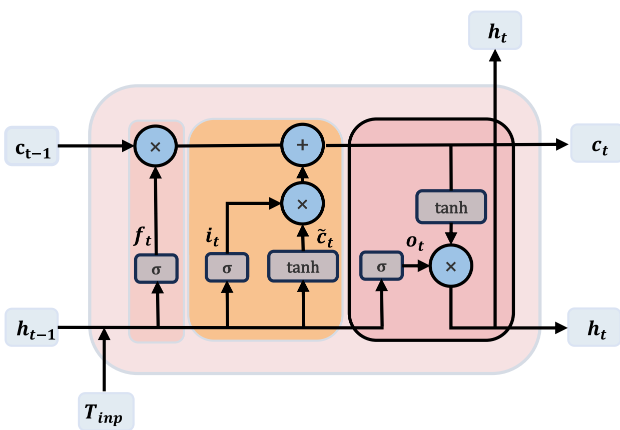

Fig. 2. The unit structure of LSTM.

in P_{Ln} is processed as a new window center point. Ultimately, the new sequence $Q_L = (Q_{L1}, Q_{L2}, \dots, Q_{Ln})$ results from the SG smoothing of P_L, where $Q_{Ln} = (q_1, q_2, \dots, q_L)$.

3.2. The inclusion of long short-term memory

Hochreiter and Schmidhuber introduced LSTM in 1997 (Hochreiter & Schmidhuber, 1997) to tackle the challenges of vanishing and exploding gradient problems encountered by traditional RNNs when processing lengthy sequences. To alleviate the sensitivity of the LSTM network to the scale of input data, the original input time series $T_{origi} = (P_L, Q_L, MA, EMA)$ is first preprocessed, followed by the application of min–max normalization to each item. The normalization equation is as follows:

$$T_{\text{inp}} = \frac{T_{\text{origi}} - \min(T_{\text{origi}})}{\max(T_{\text{origi}}) - \min(T_{\text{origi}})}, \tag{4}$$

where the MA and EMA in the sequence input represent the moving average and exponential moving average, respectively, of the closing prices over different days. The LSTM cell unit structure comprises a forget gate, an input gate, and an output gate, as depicted in Fig. 2.

The forget gate manages the level of memory erasure from the previous cell state, while the input gate governs the assimilation of current information into the cell state. The output gate adjusts how much information from the current cell state will be passed as the final output. The computation equation for the LSTM unit from input to output is as follows:

$$f_t = \sigma(W_f[h_{t-1}, T_{\text{inp}}] + b_f), \tag{5}$$
$$i_t = \sigma(W_i[h_{t-1}, T_{\text{inp}}] + b_i), \tag{6}$$
$$o_t = \sigma(W_o[h_{t-1}, T_{\text{inp}}] + b_o), \tag{7}$$
$$\tilde{c}_t = \tanh(W_c[h_{t-1}, T_{\text{inp}}] + b_c), \tag{8}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{9}$$
$$h_t = o_t \odot \tanh(c_t), \tag{10}$$
$$\bar{p}_L = h_t \cdot (\max(T_{\text{origi}}) - \min(T_{\text{origi}})) + \min(T_{\text{origi}}), \tag{11}$$

where f_t, i_t, o_t and $\tilde{c}_t$ represent the forget gate, input gate, output gate, and current candidate cell state value, respectively. T_{inp} represents the current input information. c_{t-1}, c_t, h_{t-1} and h_t denote the previous memory information, current memory information, previous output, and current output, respectively. W_f, W_i, W_o and W_c represent weight vectors. b_f, b_i, b_o and b_c denote bias vectors. The symbols σ and tanh denote the sigmoid function and hyperbolic tangent function, respectively. $\odot$ indicates the dot product. $\bar{p}_L$ represents the actual stock prediction value after performing reverse normalization on h_t.

3.3. The inclusion of sparrow search algorithm

The SSA is a metaheuristic algorithm inspired by the foraging and anti-predatory behaviors of sparrows. It is characterized by strong optimization capabilities and fast convergence. The sparrow population is divided into discoverers and joiners, including scouts (a certain proportion of individuals selected from the population). Assuming a population size of pop, a sparrow flock conducts foraging activities in a dim-dimensional space, with the position of the ith sparrow represented as $X_i (i = 1, 2, \dots, pop)$. The algorithm operates based on the following principles:

(1) Updated positions for the discoverers. The duty of the discoverers is to find the direction and location of food, guiding the joiners to move toward the food.

$$X^{t+1}_{i,j} = \begin{cases} X^t_{i,j} \cdot \exp\left(-\frac{i}{h \cdot \text{iter\_max}}\right), & G < ST \\ X^t_{i,j} + QL, & G \geq ST, \end{cases} \tag{12}$$

where $X^t_{i,j}$ represents the individual position information of the ith sparrow in the jth dimension. $iter_{max}$ (a constant) denotes the maximum iteration times, h is a random number within the range of (0,1]. Q follows a normal distribution, and L is a 1×d matrix. G is the warning

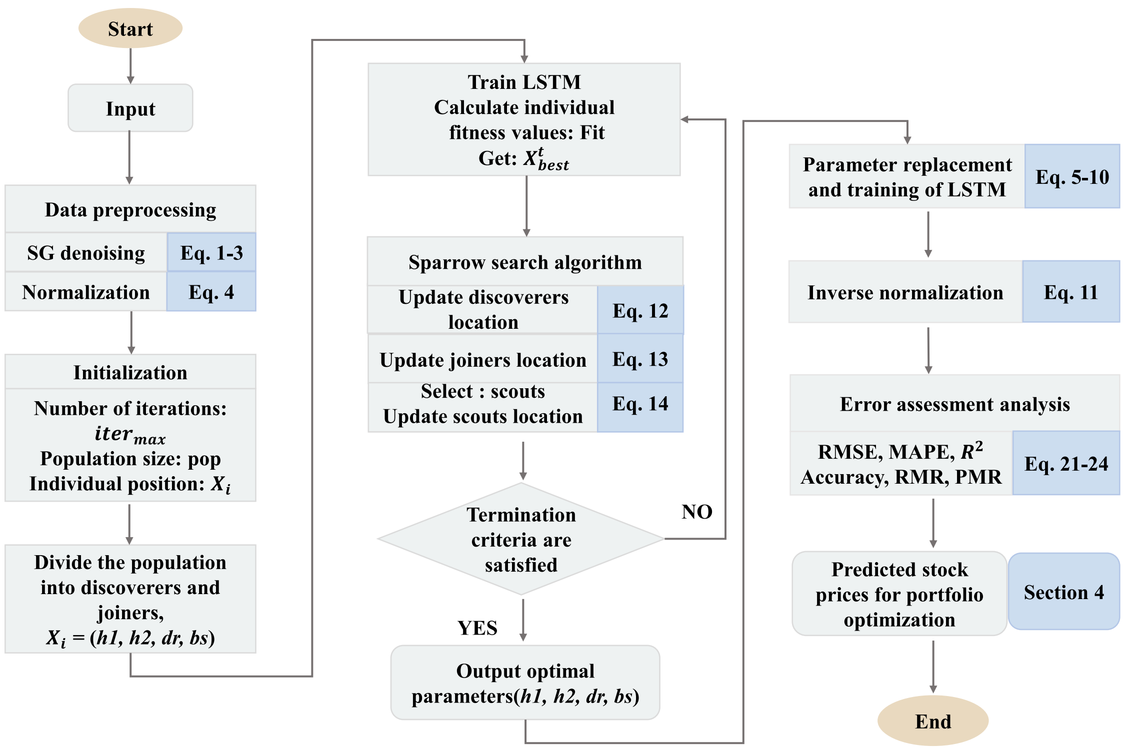

Fig. 3. The flowchart of the proposed SG-LSTM network optimized via SSA.

signal value emitted by the discoverer predator's individual, and ST is the pre-set safety threshold.

(2) The update of the joiner's position. The responsibility of the joiner is to closely monitor the actions of the discoverer and compete for food resources to increase the success rate of predation.

$$X^{t+1}_{i,j} = \begin{cases} Q \cdot \exp\left(\frac{X^t_{\text{worst}} - X^t_{i,j}}{i^2}\right), & i > \frac{t}{2} \\ X^{t+1}_p + |X^{t+1}_{i,j} - X^{t+1}_p| \cdot L \cdot A^+, & \text{otherwise}, \end{cases} \tag{13}$$

where X^t_{worst} represents the current worst value in generation t, X^{t+1}_p denotes the position of the joiner in generation t+1. A is a 1×d matrix, where each element is randomly assigned as either 1 or −1, and $A^+ = A^T(AA^T)^{-1}$.

(3) The update of the scout's position. The responsibility of the scout is to monitor and identify risks and hazards in the environment, accounting for 10% to 20% of the population.

$$X^{t+1}_{i,j} = \begin{cases} X^t_{\text{best}} + \beta |X^t_{i,j} - X^t_{\text{best}}|, & F_i > F_g \\ X^{t+1}_{i,j} + k|X^t_{i,j} - X^t_{\text{worst}}|/(F_i - F_{\text{worst}}) + \epsilon, & F_i = F_g, \end{cases} \tag{14}$$

where X^t_{best} represents the current global optimum value at iteration t. β is the step-size control coefficient, following a normal distribution. k is a randomly generated number within the interval [−1, 1]. F_i, F_{worst} and F_g correspond to the current fitness value, the global worst fitness value, and the global best fitness value, respectively. ϵ is a small constant.

3.4. Hyperparameter optimization for SG-LSTM network via SSA algorithm

In this paper, we utilize SSA to optimize the hyperparameters of the SG-LSTM network, including the number of neurons in the hidden layers (h_1, h_2), the dropout rate (dr), and the batch size (bs). The process of optimizing SG-LSTM using SSA is illustrated in Fig. 3 and will be elaborated in detail in the following steps.

Step 1: Smoothing the multi-dimensional time series P_L based on Eqs. (1) to (3) yields Q_L.

Step 2: Normalizing the original input time series T_{origi} based on Eq. (4) yields T_{inp}.

Step 3: Initialize basic parameters such as the sparrow population size (pop), maximum number of iterations (Max_{iter}), and initial position X_i.

Step 4: The sparrow population is categorized into discoverers, followers, and scouts. In the SSA algorithm, the sparrow X_i is represented as (h_1, h_2, dr, bs), and the optimization ranges for the parameters are determined.

Step 5: Compute the fitness of sparrow individuals to obtain the current optimal value X^t_{best}, where the fitness function is based on the mean squared error between actual and predicted values, expressed as follows:

$$Fit = \frac{1}{n}\sum_{t=1}^{n}(T_{\text{inp}} - h_t)^2. \tag{15}$$

Step 6: Eqs. (12) to (14) are employed for updating the positions of the discoverer, joiner and scout, respectively.

Step 7: If the termination condition is satisfied or the optimal value is found, assign the optimal value X_i to LSTM for training and output the predicted value. Otherwise, return to step (5) and continue until the termination condition is met.

Step 8: Perform inverse normalization on the output results to obtain the actual stock price prediction value $\bar{p}_L$.

Step 10: Evaluate and analyze the error between $\bar{p}_L$ and P_L based on Eqs. (21) to (24).

Step 10: Utilize the predictive analysis results for portfolio optimization, with details to be further elaborated in Section 4.

4. The mCVaR portfolio optimization model for intelligent stock selection

The MV model proposed by Markowits (1952) laid the foundation for portfolio selection. This model strikes a balance between the historical average returns and variance of assets, aiming for a trade-off between expected return and risk. However, the MV model considers only the volatility of asset returns when measuring risk, neglecting the tail risks of return distributions (i.e., potential losses that may occur in extreme situations). To comprehensively consider the risk characteristics of the portfolio, this paper adopts the CVaR method within the MV framework, replacing variance with CVaR. The introduction of the mCVaR model better reflects the uncertainty of risks and the impact of extreme events.

4.1. Risk measurement tool

Risk management plays a crucial role in financial investments, aiding investors in evaluating potential losses and formulating risk management strategies. This section introduces two primary risk measurement tools, namely VaR and CVaR (Bodnar et al., 2022).

4.1.1. Definition of VaR

VaR refers to the maximum potential loss that an investment instrument may suffer at a specific confidence level $\alpha \in (0, 1)$ over a holding period. Its calculation equation is as follows:

$$\mathrm{VaR}_\alpha(D) = \min\{x \mid F_D(e) \geq \alpha\}, \tag{16}$$

where D is the distribution function of random losses, $F_D(e)$ denotes the cumulative distribution function of the random variable D being less than or equal to a certain value e. $\mathrm{VaR}_\alpha(D)$ represents the VaR at a given confidence level α. It means the potential maximum loss of the investment instrument.

4.1.2. Definition of CVaR

CVaR is the expected loss of a portfolio when the losses exceed a certain given VaR value at a specific confidence level and holding period. Its calculation equation is as follows:

$$\begin{aligned}\mathrm{CVaR}_\alpha &= \mathrm{VaR}_\alpha + E[f(u,v) - \mathrm{VaR}_\alpha \mid f(u,v) > \mathrm{VaR}_\alpha] \\ &= E[f(u,v) \mid f(u,v) > \mathrm{VaR}_\alpha],\end{aligned} \tag{17}$$

where $u = (u_1, u_2, \dots, u_n)^\top$ represents the investment weight vector of n assets. $v = (v_1, v_2, \dots, v_n)^\top$, represents the random column vector of the returns of n assets in the portfolio. $f(u, v)$ is a random vector that represents the loss function of the portfolio.

4.2. mCVaR model

The purpose of the mCVaR model (Abudurexiti et al., 2023) is to find the optimal weight allocation of the portfolio at a given confidence level. It is expressed in a typical multi-objective optimization equation:

Model 1: Using CVaR as a risk measure, we seek to maximize the expected return of the portfolio under the constraint of the given maximum risk M_0 that can be borne. In addition, the sum of the investment weights of n assets is 1 and short selling is not allowed.

$$\begin{aligned}&\text{Maximize } E(u^\top R) \\ &\text{subject to } \begin{cases}\mathrm{CVaR}_\alpha(u^\top R) \leq M_0 \\ u^\top I = 1, \\ u \geq 0\end{cases}\end{aligned} \tag{18}$$

where $R = (r_1, r_2, \dots, r_n)^\top$, r_i represents the expected return rate of the ith asset, M_0 is the upper limit of the risk level that the investor can bear.

Model 2: Using CVaR as a risk measure, we seek to minimize the risk under the constraint of the given portfolio expected return r_0. In addition, the sum of the investment weights of n assets is 1 and short selling is not allowed.

$$\begin{aligned}&\text{Minimize } \mathrm{CVaR}_\alpha(u^\top R) \\ &\text{subject to } \begin{cases}E(u^\top R) \geq r_0 \\ u^\top I = 1, \\ u \geq 0\end{cases}\end{aligned} \tag{19}$$

where r_0 denotes the anticipated return rate of the investor's portfolio, and I is an $n \times 1$ unit matrix.

From the mathematical expression given by the definition of CVaR, observing this, the asset's loss function can be expressed as:

$$f(u, R) = -u^\top R. \tag{20}$$

This section implements a portfolio optimization method based on Monte Carlo simulation. First, high-quality individual stocks are selected based on the results of the first-stage prediction. Next, the monthly returns of individual stocks are calculated using the historical stock closing prices P_{Ln}. Subsequently, iterative random generation of portfolio weights is adopted, and the expected returns and CVaR values are calculated using Monte Carlo simulation and Eqs. (16) to (20). Finally, the optimal weight combination is determined by maximizing the Sortino ratio to more effectively control adverse risks and improve investment returns.

5. The experimental setup and benchmark strategies on test datasets

5.1. Experimental setup

This section is divided into two parts. The first section outlines the dataset and its source, with the second part offering an in-depth explanation of the specific data preprocessing procedures employing the SG filter. The hardware environment used in the experiments was Intel(R) Core (TM) i7-11800H @ 2.30 GHz, and the programming environment utilized Python 3.8, the TensorFlow framework, and the Keras API. Additionally, the parameter configurations for the hybrid prediction model can be found in Table 3.

5.1.1. Test datasets

The CSI 300 Index (Xie & Wang, 2022) is a significant stock index in the Chinese A-share market, aiming to reflect the overall performance of 300 representative large-cap stocks from both the Shanghai and Shenzhen stock exchanges. Consequently, this research randomly chose 24 stocks from the CSI 300 Index as potential assets. All data were sourced through the open-source financial data interface library AKShare, (https://www.akshare.xyz/data/stock/stock.html), which is Python-based. covering the period from June 3, 2013, to June 6, 2023. This dataset offers ample opportunities for individual investors to choose stocks prior to constructing a portfolio. The selected stock codes and price trends are shown in Table 4 and Fig. 4, respectively.

5.1.2. Data preprocessing

In the data preprocessing part, a set of data is extracted from each stock data every month (21 trading days) to build a new data set. In the experiment, the first 70% of the input data is used as the training set, 10% as the verification set, and the last 20% as the test set. After many experiments, this paper sets the number of days m for history review to 5. As the fluctuation range of each input feature varies, preprocessing is required prior to training. Specifically, the input sequence P_L is denoised by the SG filter. Secondly, the smoothed data is normalized by maximum and minimum values.

The SG filter (Chen et al., 2022) has two key parameters: the Window Length (WL) and the value of K. It is essential to ensure that: $(WL = 2c+1, c \in \mathbb{Z}^+) \cap (K < WL)$. A smaller WL leads to a curve that closely resembles the original, while a larger WL results in a smoother curve. The value of K determines the degree of the K-order polynomial fit to the data points within the window. A higher K yields a more refined fit but may lead to a straightening of high-frequency curves under larger WL. Hence, the choice of parameters is crucial. For the sake of fairness, all parameters of LSTM are uniformly set. By using the method of controlling variables, experiments were conducted to select the parameters of SG, with the experimental results shown in Fig. 5. We can note that when the value of K is approximately half of WL, the fitting of stock prices ($R\_squared$) and the accuracy of prediction (Acc) are outstanding. Finally, this study adopts the optimal experimental values, setting WL to 3 and K to 1.

Table 3
The parameter setting of the hybrid prediction model framework.

Parameter	Range	Values	Parameter	Range	Values
SG (WL)	$2c + 1(n \in {}^{+})$	3	Optimization algorithm	–	Adam
SG (K)	(0, WL]	1	Activation function	–	Tanh
Input_size	–	(5, 14)	Hidden layers	–	2
SSA-Hidden nodes	[8, 128]	(107, 79)	Loss function	–	MSE
SSA-Dropout rate	[0.01, 0.1]	0.001	Learning rate	[0.001, 0.01]	0.001
SSA-Batch_size	[2, 64]	3	history review (m)	[2, 20]	5

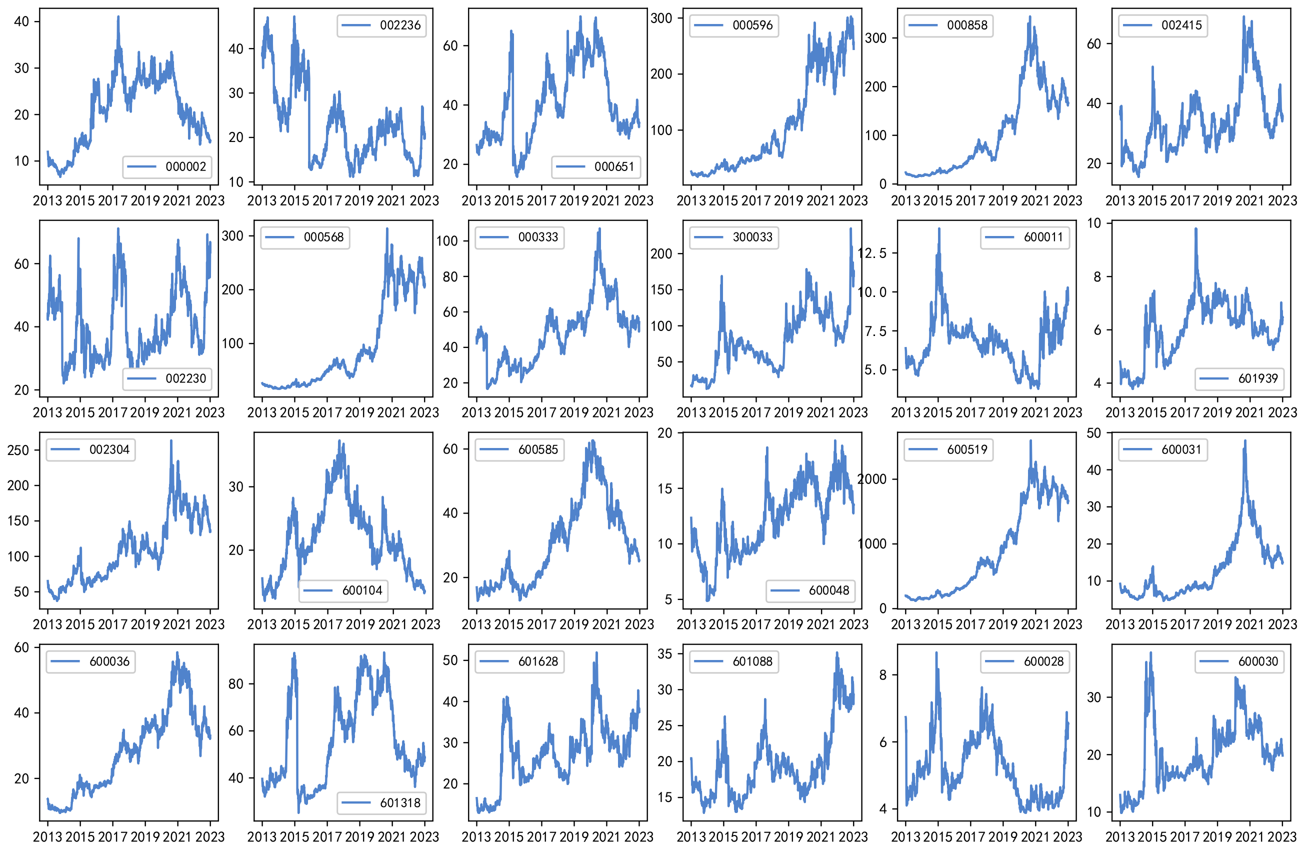

Fig. 4. Daily closing price of selected stocks.

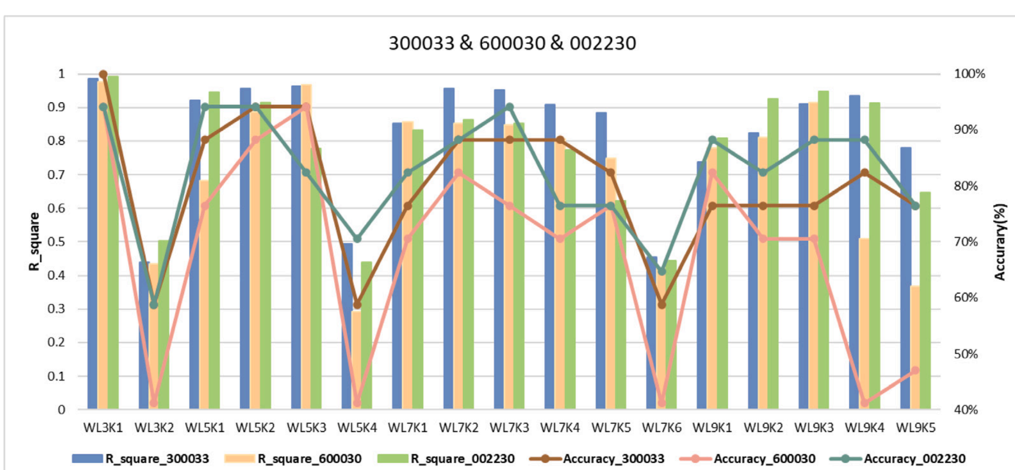

Fig. 5. Impact of parameter combinations on stock price fitting and accuracy.

Table 4
Selected stocks' tickers in the studied portfolio pool.

000002	002 236	000 651	000 596	000 858	002 415
002230	000 568	000 333	300 033	600 011	601 939
002304	600 104	600 585	600 048	600 519	600 031
600036	601 318	601 628	601 088	600028	600 030

5.2. The experimental design of our hybrid SG-SSA-LSTM+mCVaR model

Individual investors in stock investments typically face two major challenges. Firstly, there is a need to select suitable assets, and secondly, there is a requirement to appropriately allocate the investment proportions of the chosen stocks. Therefore, the primary focus of this study is to select stocks with high-precision predictions and high potential returns for portfolio construction from a predictive perspective. This process involves two key stages: portfolio prediction (predicting future stock prices) and portfolio selection (determining the optimal investment portfolio). The comprehensive flowchart is depicted in Fig. 1.

(1) Stock prediction: SG filter is used for denoising stock data. LSTM is employed to predict the next period's stock prices. Additionally, during the training process, the SSA is utilized for optimizing the hyperparameters of LSTM. This study employs mean squared error as the fitness function. The primary hyperparameters such as $h1$, $h2$, dr, bs, are updated during the minimization fitness process, as shown in Fig. 2, until the mean squared error meets the requirements. After acquiring the hyperparameters, the integrated prediction model (SG-SSA-LSTM) is optimally utilized to predict stock prices for the upcoming period.

(2) Portfolio selection: This stage is dedicated to establishing the allocation percentages for each asset. Firstly, the portfolio prediction results are preliminarily screened based on potential returns, prediction accuracy, and upward rates, selecting the top K assets in overall ranking. Subsequently, using the mCVaR model, asset allocation is performed on the basis of the preliminary screening results to construct an optimal investment portfolio with unequal proportions. The specific procedure involves generating multiple investment portfolios with different weights using the Monte Carlo method, then calculating the average rate of return and CVaR value for each portfolio under each weight. This process iterates 2,000,000 times (Wang et al., 2020). Utilizing outcomes from the mCVaR model, the optimal investment portfolio is chosen. To maintain a balanced approach between returns and risks, the Sortino ratio is utilized for informed decision-making, and funds are allocated to the investment portfolio with the highest Sortino ratio.

5.3. Benchmark strategies

In addition to the mCVaR model, it is also worthwhile to consider the equal weight (1/N) portfolio model (Du, 2022). This study gradually decomposes the portfolio model into single models and combines it with either the mCVaR model or the 1/N model to explore its effects in the experiment and for comparison. The following is the benchmark model based on SG-SSA-LSTM+mCVaR model.

(1) Alternative model: SG-SSA-LSTM+1/N

The architecture of this model resembles the SG-SSA-LSTM+mCVaR structure, aiming to confirm the efficacy of the mCVaR model under identical asset selection criteria. It utilizes the same portfolio prediction model for asset selection, differing only in the uniform distribution of asset proportions. In the initial stage, the portfolio prediction model forecasts asset prices for the upcoming period. Subsequently, a preliminary screening identifies the top K assets in the overall ranking, which are then allocated in equal proportions.

(2) Alternative model: SG-LSTM+mCVaR or 1/N

The objective of this model is to explore the influence of hyperparameter optimization on the construction of the optimal investment portfolio. In the initial stage, the prediction model without hyperparameter optimization is employed to predict stock prices. Subsequently, the top K stocks in the overall ranking are chosen in the second stage, and the proportions of various assets in the investment portfolio are determined using either the mCVaR or 1/N method.

(3) Alternative model: LSTM+mCVaR or 1/N

The purpose of this model is to explore whether SG's data smoothing and denoising affect the prediction performance. To ensure fairness, in the first stage, the stock prices are forecasted using hyperparameters optimized through the SSA. Subsequently, in the second stage, the top K stocks in overall ranking are selected, and the optimal investment portfolio is determined using the mCVaR or 1/N method.

(4) Alternative model: Random+mCVaR or 1/N

These alternative models aim to affirm the essentiality of employing machine learning for stock prediction. In contrast to the previously mentioned benchmark model, the Random + mCVaR or 1/N model relies on random selection. To elaborate, a specific number of assets are chosen randomly from all available samples, and subsequently, the investment portfolio is determined using the MV or 1/N method.

Table 5
Description of benchmark functions.

Name	Function	Range	Global optimum
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	[−100, 100]	0
Ellipsoidal	$f_2(x) = \sum_{i=1}^{D} c_i x_i^2$	[−100, 100]	0
Bent_cigar	$f_3(x) = x_1^2 + \sum_{i=2}^{D} x_i^2$	[−100, 100]	0
Discus	$f_4(x) = x_1^2 + c\sum_{i=2}^{D} x_i^2$	[−100, 100]	0
Alpine	$f_5(x) = \sum_{i=1}^{D} \lvert x_i \cdot \sin(x_i) + 0.1x_i \rvert$	[−10, 10]	0
Schwefel	$f_6(x) = -\sum_{i=1}^{D} x_i \sin\sqrt{\lvert x_i \rvert}$	[−500, 500]	−12569.487

6. Data analysis and discussions of numerical results

6.1. Effectiveness of the introduced SSA

In this section, two unimodal functions and four multimodal functions were chosen to evaluate the performance of SSA. Refer to Table 5 for comprehensive explanations of the benchmark functions. In the subsequent experiments, we compared SSA with GWO (Ahmed et al., 2023), SOA (Jia et al., 2019), Dung Beetle Optimizer (DBO) (Xue & Shen, 2023), and PSO (Wang, Zhao, & Tian, 2023). For specific parameters, please refer to Table 6.

The performance metrics of different algorithms are presented in Table 7, indicating that the SSA algorithm demonstrates significant advantages in various aspects, including maximum, mean, and standard deviation. Furthermore, in most cases, the SSA algorithm's minimum value also surpasses other algorithms. However, it is essential to highlight that the computation time for the SSA algorithm is slightly longer in comparison to the GWO, SOA, and PSO algorithms. To provide a clearer representation of the performance of different algorithms, we illustrate their convergence curves in Fig. 6. The graph unmistakably illustrates the SSA algorithm's superior performance in both convergence speed and accuracy compared to other algorithms. Overall, the SSA algorithm demonstrates outstanding performance in terms of convergence precision, stability, and robustness, surpassing GWO, SOA, DBO, and PSO algorithms by a significant margin.

6.2. Stock price prediction results

Effective forecasting holds significant value in achieving investment returns and risk management. We compare the proposed SG-SSA-LSTM with other composite models, including SG-GWO-LSTM (LSTM hyperparameters optimized by GWO), SG-SOA-LSTM (LSTM hyperparameters optimized by SOA), SG-PSO-LSTM (LSTM hyperparameters optimized by PSO), and SG-LSTM, as well as the single model LSTM. To examine the accuracy of the stock prediction methods (Behera et al.,

Table 6
Configurations of various algorithm parameters.

Parameters	Population size	Dimension of function	Maximum iterations
	60	30	200
SSA	Constant $C_1 \in [0, 1]$	Constant $C_2 \in [0, 1]$	
GWO	Random number $a \in [0, 2]$	Coefficient vector $A \in [-a, a]$	Coefficient vector $C \in [0, 2]$
SOA	$fc = 3$	$u = 1$	$v = 1$
DBO	$b = 0.3$	$P\_percent = 0.2$	
PSO	Inertia constant $\omega = 0.8$	Cognitive constant C_1=1.5	Social constant C_2=1.5

Table 7
Analyzing benchmark function comparisons.

Functions	Algorithms	Maximum	Minimum	Mean	Std	CPU Time (s)
Sphere	**SSA**	**1.16e−165**	**0.00000**	**3.87e−167**	**0.00000**	1.25
	GWO	5.27e−80	0.00000	1.75e−81	9.46e−81	1.10
	SOA	2.99e−82	1.83907	9.99e−84	5.37e−83	1.14
	DBO	11 608.1	1073.38	3291.81	3683.16	3.44
	PSO	17.9667	2.47112	5.86176	2.69689	**0.24**
Ellipsoidal	**SSA**	**1.37e−112**	**0.00000**	**4.56e−114**	**2.46e−113**	1.32
	GWO	6.05e−83	0.00000	2.01e−84	1.08e−83	1.17
	SOA	2.22e−82	1.42e−103	1.14e−83	4.43e−83	1.19
	DBO	353 480	17 725.2	91 562.2	80 092.0	3.68
	PSO	260.539	28.6857	79.3322	49.1777	**0.27**
Bent_cigar	**SSA**	**6.16e−143**	**0.00000**	**2.53e−144**	**1.12e−143**	1.27
	GWO	4.68e−131	0.00000	1.56e−132	8.4e−132	1.12
	SOA	8.13e−85	4.83e−107	3.08e−86	1.46e−85	1.15
	DBO	11 916.1	919.480	4356.61	4240.56	3.47
	PSO	12.1413	1.90917	5.66940	2.23037	**0.25**
Discus	**SSA**	**8.11e−114**	**0.00000**	**2.7e−115**	**1.45e−114**	1.27
	GWO	6.89e−95	0.00000	2.29e−96	1.23e−95	1.12
	SOA	1.84e−80	4.74e−106	7.65e−82	3.35e−81	1.15
	DBO	113 074	42 809.2	76 829.2	20 584.7	3.47
	PSO	155.401	50.2637	94.0945	28.8410	**0.25**
Alpine	**SSA**	**1.5e−59**	**0.00000**	**5.14e−61**	**2.77e−60**	1.27
	GWO	1.24e−50	0.00000	4.14e−52	2.23e−51	1.12
	SOA	2.19e−40	3.57e−47	2.58e−41	5.47e−41	1.14
	DBO	51.7357	8.46543	29.5634	8.14562	3.49
	PSO	14.0874	5.87398	9.01908	1.94959	**0.26**
Schwefel	**SSA**	**−11633.8**	−12562.1	**−12396.6**	**230.451**	1.23
	GWO	−5866.97	−12546.1	−8024.51	1830.41	1.08
	SOA	−9868.59	**−12569.4**	−12389.2	516.975	1.07
	DBO	−4688.76	−7038.55	−5757.22	686.229	3.37
	PSO	−374.496	−9128.11	−6085.33	1742.30	**0.26**

2023), this paper employs six metrics: Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), coefficient of determination (R^2), Accuracy, Real Monthly Returns (RMR), and Predicted Monthly Returns (PMR). RMSE, MAPE and R^2 typically indicate the fit of the stock prices, while Accuracy, RMR, and PMR reflect the volatility of the stock prices. The following are the definitions for these metrics:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}, \tag{21}$$

$$\mathrm{MAPE} = \frac{(100\%)}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|, \tag{22}$$

$$R^2 = \frac{1 - \sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{n}(y_t - \bar{y}_t)^2}, \tag{23}$$

$$\mathrm{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP}, \tag{24}$$

where $\hat{y}_t$, $\bar{y}_t$, and y_t represent the predicted price, actual price mean, and actual price, respectively. n is the total number of samples. True Positive (TP) signifies the number of samples accurately predicted as the positive class by the model. True Negative (TN) denotes the number of samples accurately predicted as the negative class. False Positive (FP) represents the number of samples the model incorrectly predicted as the positive class when they are negative, and False Negative (FN) indicates the number of samples the model erroneously predicted as the negative class when they are positive. Additionally, training time serves as a metric for evaluating the complexity of various prediction models.

The average values (AVG) in Tables 8 to 10 indicate that the SG-SSA-LSTM model exhibits the smallest error metrics among all predictive models. In comparison to SG-GWO-LSTM and SG-SOA-LSTM, there is a noticeable reduction in errors for RMSE, MAPE, RMR, and PMR. The R^2 and Accuracy metrics also show improvements, highlighting the effectiveness of SSA-based optimization.

Compared to SSA-LSTM, SG-SSA-LSTM shows reductions of 11.0744, 0.0698, and 0.0087 in RMSE, MAPE, RMR, and PMR, respectively. The improvements in R^2 and Accuracy are 0.7754 and 0.3820. Meanwhile, compared to LSTM, SG-LSTM exhibits reductions of 11.5643, 0.0736, and 0.0056 in RMSE, MAPE, RMR, and PMR, respectively. The enhancements in R^2 and Accuracy are 0.9137 and 0.3993. This indicates that using SG for denoising stock data is crucial.

Compared to the LSTM single model, the SG-SSA-LSTM model shows significant improvements in error metrics, with reductions of 12.8531, 0.0852, and 0.0178 in RMSE, MAPE, RMR, and PMR, respectively. The enhancements in R^2 and Accuracy are 0.9482 and 0.4248. Other hybrid models, such as SG-GWO-LSTM, SG-SOA-LSTM, SG-LSTM, and SSA-LSTM, also exhibit varying degrees of improvement across six different indicators. This confirms that hybrid models have higher predictive accuracy than single models. However, it is worth noting that hybrid models (SG-SSA-LSTM, SG-GWO-LSTM, SG-SOA-LSTM, SG-LSTM, and SSA-LSTM) require more computational time as they utilize swarm intelligence algorithms for hyperparameter optimization, while single models (LSTM) use manually set hyperparameters.

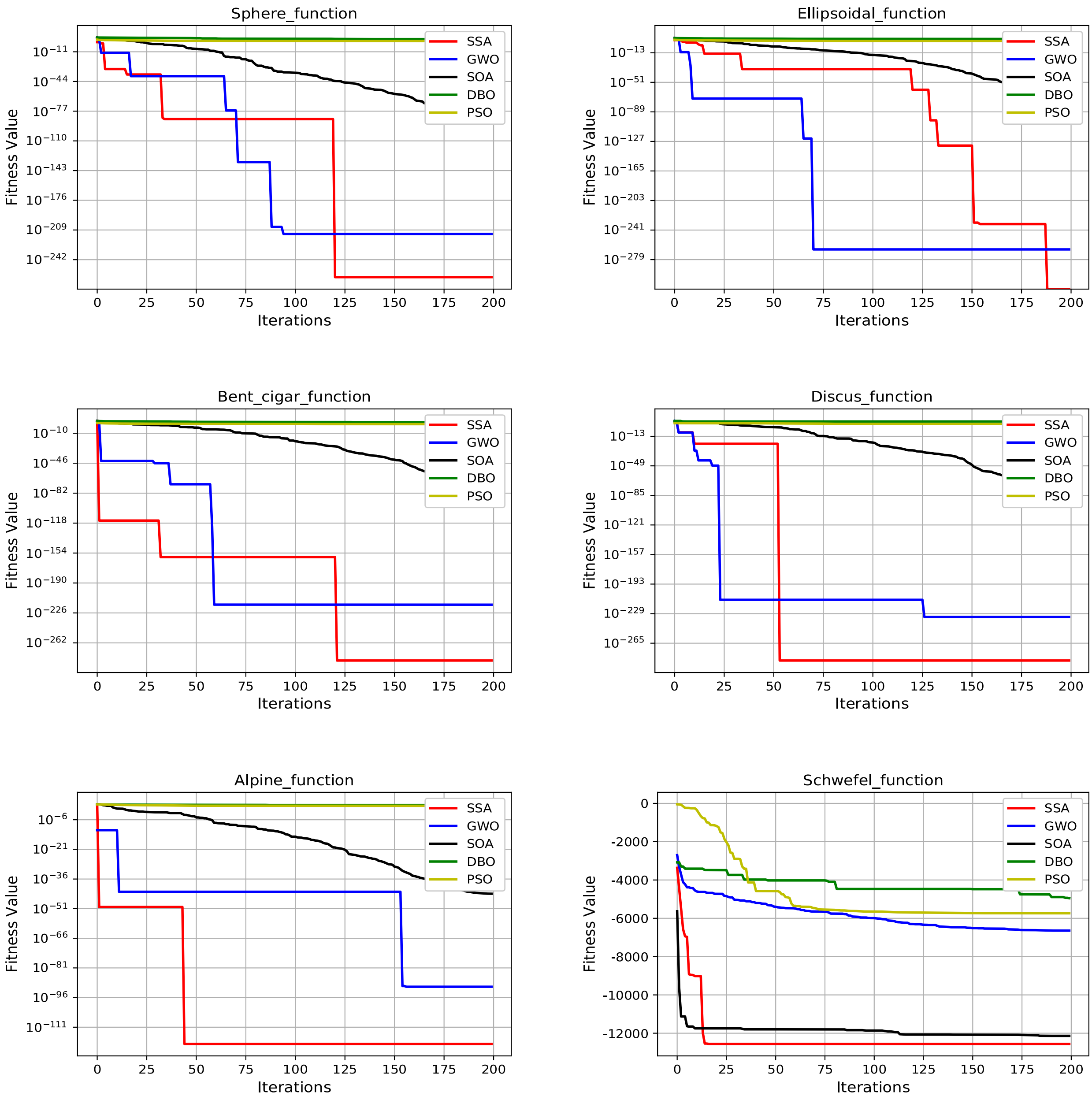

Fig. 6. Convergence performance of different algorithms.

6.3. Optimal portfolio selection

The objective of this section is to ascertain the optimal size for the investment portfolio. We investigated portfolios of different scales to find the optimal choice. Considering that individual investors find it challenging to manage a large number of stocks effectively, much research has focused on constructing portfolios with fewer stocks. A study by Paiva et al. (2019) suggests that an investment portfolio comprising an average of 7 different assets might be the most suitable choice, while Wang et al. (2020) argue that portfolios with 10 different assets exhibit better performance. Given these differences, we selected different stock portfolios with cardinalities k ranging from 6 to 10 for our study. We evaluated the performance of these portfolios using metrics such as annualized average return, annualized CVaR, annualized Sortino ratio, and the ratio of annualized average return to CVaR. The Sortino ratio is employed to assess the performance of assets or portfolios relative to downside risk, while the return-to-risk ratio considers the balance between returns and risks comprehensively. It is important to note that the risk-free asset's yield was set at 0.03 based on data from Chinese treasury bond rates over the past decade.

Fig. 7 depicts the annualized performance of different models across various cardinalities (k = 6, 7, 8, 9, 10). The findings suggest that SG-SSA-LSTM+mCVaR surpasses alternative models in mean return, Sortino ratio, and return-to-risk ratio (Return/Risk). Across k values of 6, 7, 8, 9, and 10, SG-SSA-LSTM+mCVaR consistently demonstrates the highest mean return. Specifically, when k is 7, the mean return of SG-SSA-LSTM+mCVaR is 3.1%, compared to 2.7%, 1.2%, 1.4%, 0.8%, 0.79%, 0.59%, and 0.66% for SG-LSTM+mCVaR, SG-SSA-LSTM+1/N, SG-LSTM+1/N, LSTM+mCVaR, LSTM+1/N, Random+mCVaR, and Random+1/N, respectively. As a risk measure, the annualized CVaR demonstrates that SG-SSA-LSTM+mCVaR exhibits relatively lower risk compared to other hybrid models (SG-SSA-LSTM+1/N, SG-LSTM+mCVaR, SG-LSTM+1/N) for k=6, 7, and 8. Nevertheless, it is crucial to emphasize that individual models usually

Table 8
Comparison of SG-SSA-LSTM and SG-GWO-LSTM.

Stock	SG-SSA-LSTM							SG-GWO-LSTM						
	RMSE	MAPE	R^2	Acc	RMR	PMR	Time (s)	RMSE	MAPE	R^2	Acc	RMR	PMR	Time (s)
000002	0.2467	0.0132	0.9837	0.9375	0.0285	0.0256	**325.73**	0.4462	0.0237	0.9467	0.8750	0.0285	0.0307	729.02
600031	0.2639	0.0123	0.9871	**0.9705**	−0.0085	−0.0101	418.61	0.6762	0.0280	0.9153	**0.9705**	−0.0085	−0.0111	**262.51**
600104	0.1334	0.0063	0.9958	0.8800	0.0063	−0.0017	822.64	0.1363	0.0074	0.9956	0.9411	0.0063	0.0007	689.50
600030	0.1735	0.0073	0.9938	0.9411	0.0565	0.0599	613.75	0.1777	0.0068	0.9935	0.9411	0.0565	0.0684	580.72
601628	0.3976	0.0106	0.9906	0.9411	−0.0174	−0.0195	469.79	0.3881	0.0105	0.9911	0.9411	**−0.0174**	**−0.0176**	710.35
601318	0.3842	0.0065	0.9893	0.9411	0.0057	0.0043	682.22	0.6313	0.0108	0.9711	0.8823	0.0057	−0.0012	729.39
600028	**0.0364**	**0.0048**	0.9970	0.9411	0.0976	0.1296	646.64	0.0567	0.0108	0.9927	0.9411	0.0976	0.1359	985.40
601088	0.3862	0.0115	0.9781	0.9411	0.0266	0.0097	611.75	0.4410	0.0123	0.9714	0.9411	0.0266	0.0071	677.12
000858	1.8669	0.0080	0.9926	**0.9705**	−0.0750	−0.0856	618.20	3.6009	0.0155	0.9726	0.9411	−0.0750	−0.0645	515.08
300033	2.8333	0.0133	0.9912	**0.9705**	0.6583	0.6121	664.59	3.1273	0.0212	0.9893	**0.9705**	0.6583	0.6562	671.93
600036	0.6105	0.0120	0.9918	0.9444	0.0352	0.0479	642.10	1.2458	0.0269	0.9660	0.8888	0.0352	0.0452	518.48
000568	3.3145	0.0126	0.9729	0.9411	−0.0234	−0.0055	672.88	4.3749	0.0160	0.9528	0.9411	−0.0234	−0.0014	583.29
002304	1.3974	0.0073	0.9890	**0.9705**	−0.0830	−0.0781	531.94	2.7614	0.0134	0.9572	0.8823	−0.0830	−0.0758	364.31
000333	0.5366	0.0085	0.9955	**0.9705**	0.0375	0.0427	781.07	0.8711	0.0124	0.9883	0.9375	0.0375	0.0441	791.04
600048	0.1592	0.0081	0.9837	0.9412	−0.0167	−0.0134	898.67	0.2651	0.0121	0.9549	0.9406	−0.0167	−0.0084	560.30
601939	0.0324	0.0049	0.9903	0.7778	**0.0657**	**0.0641**	554.64	**0.0411**	**0.0053**	0.9844	0.7778	0.0657	0.0625	618.73
600011	0.0931	0.0091	0.9883	**0.9705**	−0.0179	−0.0030	593.15	0.1430	0.0144	0.9725	0.9412	−0.0179	0.0000	814.13
000596	3.3911	0.0118	0.9892	0.8235	0.0119	0.0142	495.81	5.6224	0.0195	0.9702	0.9412	0.0119	0.0234	450.06
002236	0.2584	0.0089	0.9956	0.9412	0.0632	0.1041	802.19	0.5240	0.0197	0.9818	0.9412	0.0632	0.1709	566.25
000651	0.3674	0.0080	0.9828	**0.9705**	0.0733	0.0596	581.41	0.4595	0.0096	0.9731	**0.9705**	0.0733	0.0577	609.51
002415	0.5947	0.0126	0.9916	0.9412	−0.1134	−0.0776	644.14	0.7470	0.0139	0.9868	0.9412	−0.1134	−0.0644	485.78
600585	0.2411	0.0063	**0.9980**	0.9412	−0.0432	−0.0544	599.38	0.3250	0.0086	**0.9964**	0.9412	−0.0432	−0.0548	838.17
002230	0.4859	0.0093	0.9968	**0.9705**	0.1700	0.1524	569.73	0.6632	0.0131	0.9940	**0.9705**	0.0656	0.1574	481.16
600519	15.729	0.0063	0.9828	0.9412	−0.0159	−0.0381	813.61	25.138	0.0110	0.9561	0.9412	−0.0159	−0.0270	561.47
AVG	**1.4139**	**0.0091**	**0.9894**	**0.9366**	**0.0384**	**0.0391**	**627.27**	**2.2026**	**0.0142**	**0.9739**	**0.9292**	**0.0340**	**0.0472**	**616.40**

Table 9
Comparison of SG-SOA-LSTM and SG-LSTM.

Stock	SG-SOA-LSTM							SG-LSTM						
	RMSE	MAPE	R^2	Acc	RMR	PMR	Time (s)	RMSE	MAPE	R^2	Acc	RMR	PMR	Time (s)
000002	0.2717	0.0130	0.9802	0.8750	0.0285	0.0304	757.20	0.5850	0.0315	0.9084	0.8750	0.0285	0.0216	22.65
600031	0.3849	0.0156	0.9725	**0.9705**	−0.0085	−0.0148	824.29	0.6240	0.0303	0.9279	**0.9705**	−0.0085	−0.0091	24.18
600104	0.1725	0.0093	0.9929	0.9411	0.0063	0.0083	1064.71	0.4362	0.0236	0.9551	0.8823	0.0063	−0.0039	29.67
600030	0.2107	0.0080	0.9909	0.9411	0.0565	0.0661	1180.79	0.5419	0.0232	0.9398	0.9411	0.0565	0.0779	29.54
601628	0.4510	0.0121	0.9880	**0.9705**	−0.0174	−0.0150	1187.49	0.6798	0.0176	0.9728	**0.9705**	−0.0174	−0.0064	32.13
601318	0.7461	0.0130	0.9596	**0.9705**	0.0057	0.0028	1038.53	0.8619	0.0149	0.9462	**0.9705**	0.0057	0.0063	32.32
600028	0.0394	0.0056	**0.9965**	0.9411	0.0976	0.1277	1334.25	0.1137	0.0169	0.9708	0.8235	0.0976	0.1907	28.29
601088	0.5985	0.0159	0.9474	0.9411	0.0266	0.0252	937.43	0.7619	0.0223	0.9147	0.9411	0.0266	0.0208	**16.50**
000858	2.1098	0.0095	0.9906	**0.9705**	−0.0750	−0.0882	594.93	4.3348	0.0193	0.9603	**0.9705**	−0.0750	−0.0862	30.18
300033	2.9121	0.0143	0.9907	**0.9705**	0.6583	0.5981	1304.29	3.6701	0.0329	0.9852	**0.9705**	0.6583	0.6194	24.95
600036	1.0535	0.0238	0.9757	0.8333	0.0352	0.0403	943.90	1.8189	0.0345	0.9276	0.8333	0.0352	0.0345	32.62
000568	3.6887	0.0143	0.9664	0.9411	−0.0234	0.0008	572.62	5.6629	0.0234	0.9210	0.9411	−0.0234	−0.0022	27.75
002304	1.5462	0.0075	0.9866	0.8823	**−0.0830**	**−0.0823**	**422.27**	3.5759	0.0169	0.9283	0.7647	−0.0830	−0.0784	22.28
000333	1.0050	0.0159	0.9844	0.8750	0.0375	0.0476	1209.62	1.3622	0.0196	0.9714	0.9375	0.0375	0.0404	38.97
600048	0.2427	0.0098	0.9622	**0.9705**	−0.0167	−0.0087	636.34	0.2892	0.0150	0.9463	0.9412	−0.0167	−0.010	35.33
601939	**0.0366**	**0.0050**	0.9877	0.7778	0.0657	0.0615	966.32	0.0519	**0.0077**	0.9829	0.8333	0.0657	0.0649	33.09
600011	0.1277	0.0137	0.9781	**0.9705**	−0.0179	−0.0043	1019.88	0.1611	0.0161	0.9652	0.9485	−0.0179	−0.0018	38.75
000596	3.8971	0.0134	0.9857	**0.9705**	0.0119	0.0323	1113.98	5.8120	0.0207	0.9682	0.9412	0.0119	0.0370	40.32
002236	0.5201	0.0264	0.9821	0.8824	0.0632	0.1564	1453.13	0.8330	0.0317	0.9541	0.8824	0.0632	0.2686	35.63
000651	0.3852	0.0093	0.9811	**0.9705**	0.0733	0.0625	1066.85	0.4679	0.0097	0.9721	0.9412	0.0733	0.0650	35.06
002415	0.6339	0.0127	0.9905	0.9412	−0.1134	−0.0851	910.64	0.8179	0.0166	0.9842	0.8824	−0.1134	−0.0803	28.86
600585	0.3374	0.0087	0.9962	0.8824	−0.0432	−0.0490	1113.34	**0.0103**	0.0245	**0.9914**	**0.9705**	−0.0432	−0.0593	36.03
002230	0.5179	0.0094	0.9963	0.9412	0.1700	0.1547	976.72	0.8265	0.0152	0.9907	**0.9705**	0.1700	0.1528	37.64
600519	18.681	0.0088	0.9758	0.8824	−0.0159	−0.0276	773.24	30.566	0.0136	0.9351	0.7647	−0.0159	−0.0290	35.38
AVG	**1.6904**	**0.0122**	**0.9815**	**0.9255**	**0.0384**	**0.0433**	**975.11**	**2.7027**	**0.0207**	**0.9549**	**0.9111**	**0.0384**	**0.0513**	**31.17**

come with reduced risk, as higher returns are frequently associated with elevated risk. Considering the Sortino ratio, it is evident that SG-SSA-LSTM+mCVaR reaches its peak at 2.22 for k=7, with SG-LSTM+mCVaR following closely at 1.75. In terms of the Return/Risk, SG-SSA-LSTM+mCVaR delivers strong results. For instance, when k=7, the SG-SSA-LSTM+mCVaR model achieves the highest Return/Risk of 2.42, followed by SG-LSTM+mCVaR (1.93), LSTM+mCVaR (1.68), LSTM+1/N (1.33), SG-SSA-LSTM+1/N (1.14), SG-LSTM+1/N (1.13), Random+mCVaR (1.11), and Random+1/N (1.06).

In summary, most models perform well in terms of average returns, CVaR, Sortino ratio, and return-to-risk ratio, particularly at the $k = 7$ level. Therefore, consistent with the findings of Paiva et al. (2019), this study selects the portfolio with a cardinality of 7 for further analysis.

6.4. Visualization of model performance

In this section, we first discuss the impact of confidence levels on individual investor decisions. Subsequently, we present the cumulative returns of the model at different confidence levels, including scenarios with and without considering transaction costs. Finally, we gauge the effectiveness and supremacy of the SG-SSA-LSTM+mCVaR model.

6.4.1. Different levels of confidence

The confidence level reflects the credibility or accuracy of the interval range of estimated population parameters obtained through data analysis. Different confidence levels have a certain impact on portfolio construction and risk management. In the theory of CVaR risk

Table 10
Comparison of SSA-LSTM and LSTM.

Stock	SSA-LSTM							LSTM						
	RMSE	MAPE	R^2	Acc	RMR	PMR	Time (s)	RMSE	MAPE	R^2	Acc	RMR	PMR	Time (s)
000002	2.4395	0.1021	0.6084	0.7058	0.0285	−0.0216	290.65	2.6163	0.1343	−0.8313	0.3750	0.0285	−0.0713	16.76
600031	2.5021	0.0998	−0.0279	**0.7058**	−0.0085	0.0091	380.18	2.5703	0.1119	−0.2230	**0.7058**	**−0.0085**	**−0.0070**	16.35
600104	1.3250	0.0936	0.4551	0.5882	0.0063	−0.0239	795.67	1.7295	0.1021	0.2946	0.4117	0.0063	−0.0446	13.20
600030	1.7749	0.0532	0.3398	0.3705	0.0565	−0.0779	580.54	1.8281	0.0700	0.3159	0.2941	0.0565	−0.0470	17.37
601628	2.4350	0.0476	0.6728	0.3705	−0.0174	0.0064	430.13	2.8772	0.0691	0.5128	0.3529	−0.0174	−0.0002	14.71
601318	4.2381	0.0549	−0.1462	0.4117	**0.0057**	**0.0093**	651.32	4.7283	0.0794	−0.6184	0.5294	0.0057	−0.0348	16.26
600028	**0.2514**	0.0269	0.7708	0.7058	0.0976	−0.1907	624.29	0.2832	0.0345	**0.8193**	0.5882	0.0976	0.2499	14.76
601088	1.2110	0.0923	−0.5147	0.4375	0.0266	−0.0208	580.50	3.4274	0.1069	−0.7240	**0.7058**	0.0266	0.0516	**11.63**
000858	29.875	0.1193	0.2603	0.7058	−0.0750	0.0862	591.18	30.402	0.1426	−0.9503	**0.7058**	−0.075	−0.0417	15.02
300033	19.951	0.1290	−0.2852	0.4117	0.6583	0.4194	632.95	22.816	0.1411	0.4313	0.5294	0.6583	0.0253	15.32
600036	4.2380	0.0945	0.5276	0.3529	0.0352	0.0145	611.62	5.6565	0.1192	0.3007	0.3333	0.0352	−0.0433	14.40
000568	23.705	0.0734	−0.3210	0.6111	−0.0234	−0.0022	640.75	24.902	0.0971	−0.5267	0.5882	−0.0234	−0.0026	24.76
002304	17.225	0.0869	0.1283	0.5294	−0.0830	0.0784	500.28	18.111	0.0979	−0.8381	0.4117	−0.0830	0.0087	12.75
000333	7.0121	0.0996	0.3714	0.5294	0.0375	−0.0404	751.97	7.0606	0.1002	0.2327	0.4375	0.0375	0.0318	15.56
600048	0.9942	0.0350	0.3563	0.4706	−0.0167	0.0600	862.33	1.1047	0.0585	0.2175	0.4706	−0.0167	−0.0320	22.10
601939	0.3014	**0.0177**	0.7829	0.7058	0.0657	−0.0349	529.09	**0.2208**	**0.0290**	0.5511	0.6111	0.0657	0.0544	31.07
600011	1.2405	0.1061	−0.9052	0.5294	−0.0179	0.0018	563.75	1.2091	0.1255	−0.9661	0.4705	−0.0179	0.0747	14.18
000596	22.435	0.0907	0.6682	0.7058	0.0119	0.0370	475.32	27.975	0.1003	0.2631	0.4118	0.0119	−0.0071	27.41
002236	1.2630	0.0817	0.4841	0.7058	0.0632	0.2686	782.63	2.8611	0.1071	0.4589	**0.7058**	0.0632	0.4224	35.81
000651	2.2221	0.0697	0.2721	0.4117	0.0733	−0.0650	564.06	3.5482	0.0822	−0.6046	0.3529	0.0733	0.0785	12.78
002415	3.5200	0.0866	**0.7842**	0.5239	−0.1134	0.0803	624.86	4.1530	0.0909	0.5928	0.5294	−0.1134	0.1543	22.48
600585	1.1023	0.0645	−0.3014	0.6451	−0.0432	−0.0593	579.03	2.5226	0.0717	0.7858	0.5882	−0.0432	−0.0182	35.72
002230	4.2280	0.1052	0.5907	0.5882	0.1700	0.3228	549.64	6.8457	0.1136	0.3593	0.5882	0.1700	0.0958	13.24
600519	144.23	0.0636	−0.4351	0.5882	−0.0159	0.0290	795.38	162.97	0.0796	−0.8438	0.5882	−0.0159	−0.4200	32.36
AVG	**12.4883**	**0.0789**	**0.1121**	**0.5546**	**0.0384**	**0.0290**	**599.51**	**14.267**	**0.0943**	**−0.0412**	**0.5118**	**0.0384**	**0.0199**	**19.41**

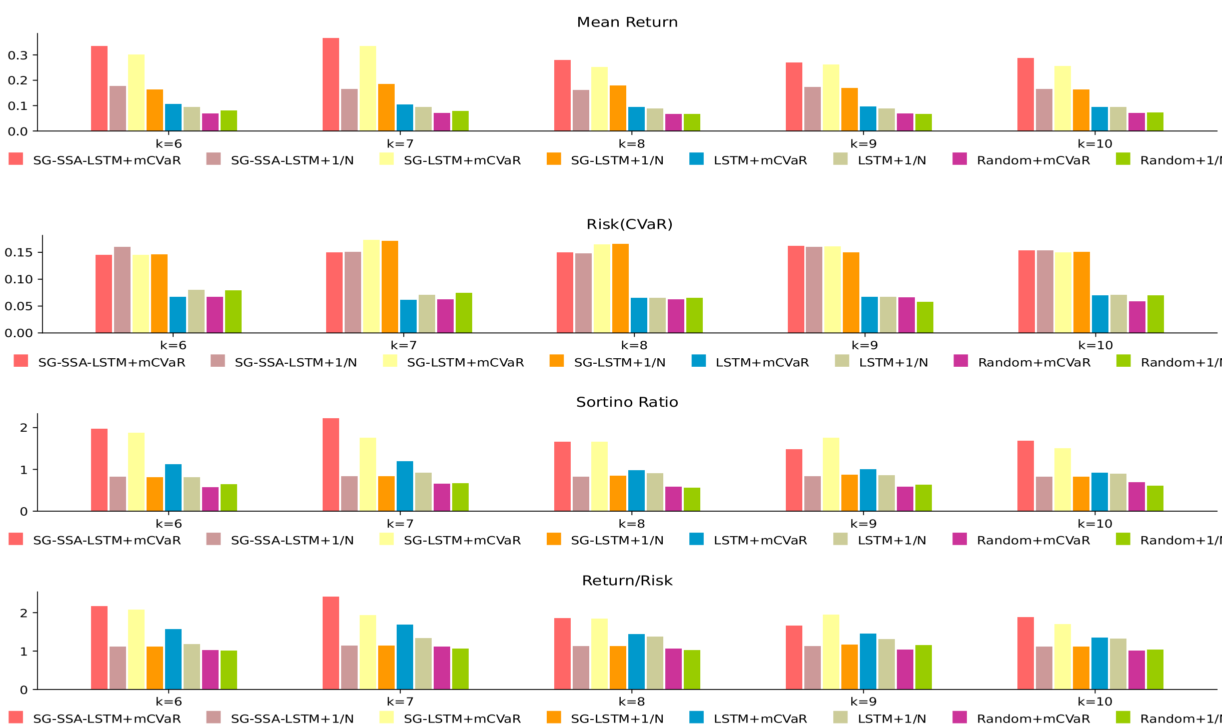

Fig. 7. Annualized performance of portfolios of different sizes.

measurement, selecting an appropriate confidence level is crucial for optimizing the investment portfolio (Ran et al., 2024). In general, investors typically determine the confidence level based on their personal risk preferences when making investment decisions. Investors with a high risk preference may choose a lower confidence level (0.90), while those with a moderate risk preference may opt for a moderate confidence level (0.95). Conversely, individuals with a low risk preference might select a higher confidence level (0.99). In this study, assuming a positive expected return, the mCVaR model is employed to compute the optimal portfolio weights and corresponding CVaR values across confidence levels of 0.90, 0.95, and 0.99.

6.4.2. Model performance without transaction costs

To better demonstrate the superiority of the proposed SG-SSA-LSTM+mCVaR, we further computed the expected returns based on the obtained optimal weights. Fig. 8 illustrates the cumulative returns of various models without considering transaction costs at different confidence levels. The cumulative returns of SG-SSA-LSTM+mCVaR outperform the benchmarks. For instance, at a confidence level of 0.95, the cumulative return of SG-SSA-LSTM+mCVaR is 9.38, while SG-SSA-LSTM+1/N achieves 2.73, SG-LSTM+mCVaR achieves 6.70, SG-LSTM+1/N achieves 3.94, LSTM+mCVaR achieves 2.24, LSTM+1/N achieves 0.82, Random+mCVaR achieves 1.15, and Random+1/N

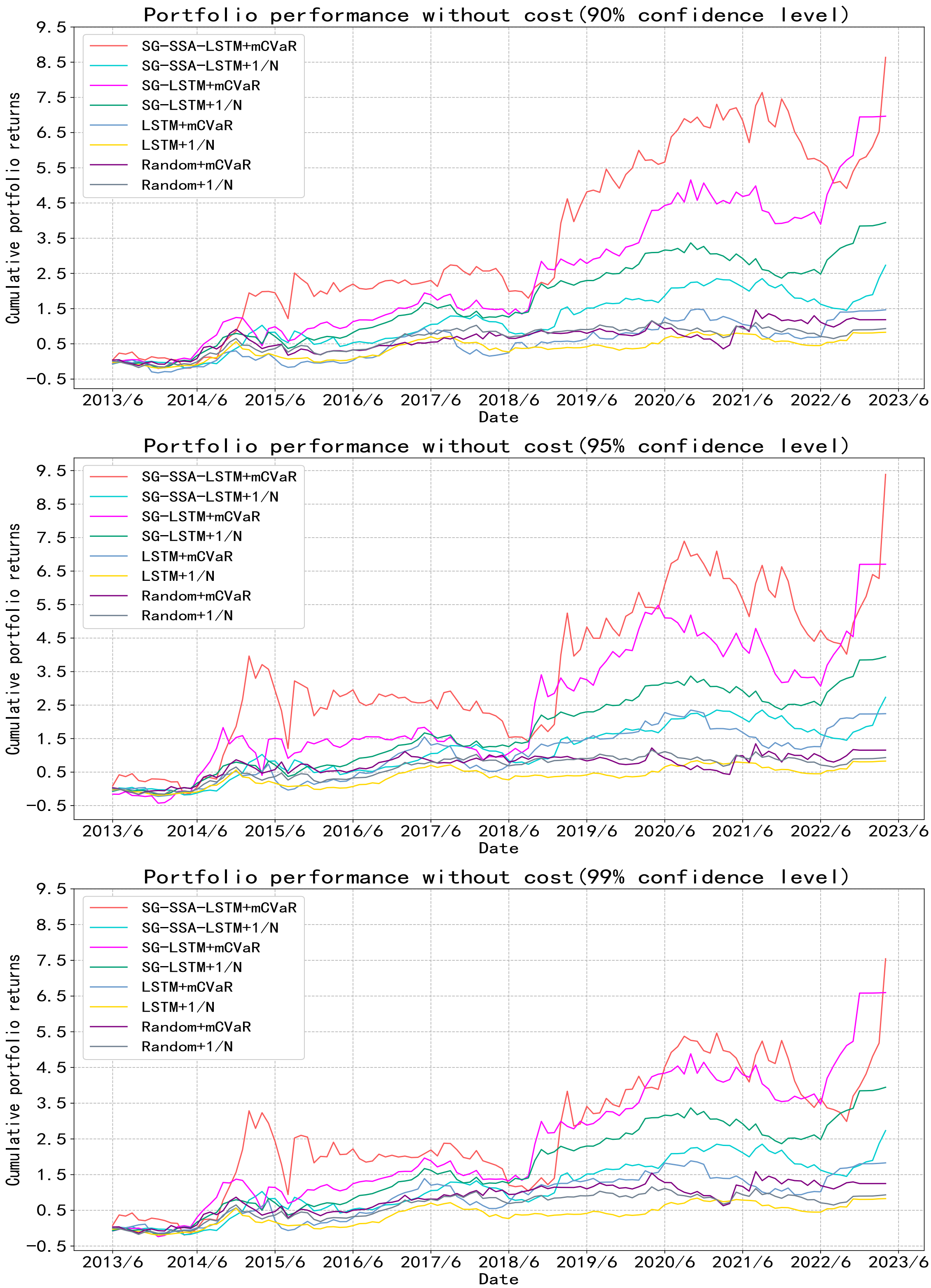

Fig. 8. Cumulative returns without factoring in transaction costs.

achieves 0.93. Compared to the 0.95 confidence level, the cumulative returns of the SG-SSA-LSTM+mCVaR model slightly decrease to 8.63 and 7.54 at confidence levels of 0.9 and 0.99, respectively. Simultaneously, other benchmark models also exhibit varying degrees of decline in cumulative returns. These findings underscore the superior performance of the SG-SSA-LSTM+mCVaR model in comparison to alternative models at different confidence levels. Meanwhile, the performance of each model is particularly prominent at the 0.95 confidence level.

6.4.3. Model performance with transaction costs

After factoring in transaction costs of 0.5% and 1%, it is evident from Figs. 9 and 10 that the cumulative returns of all models exhibit

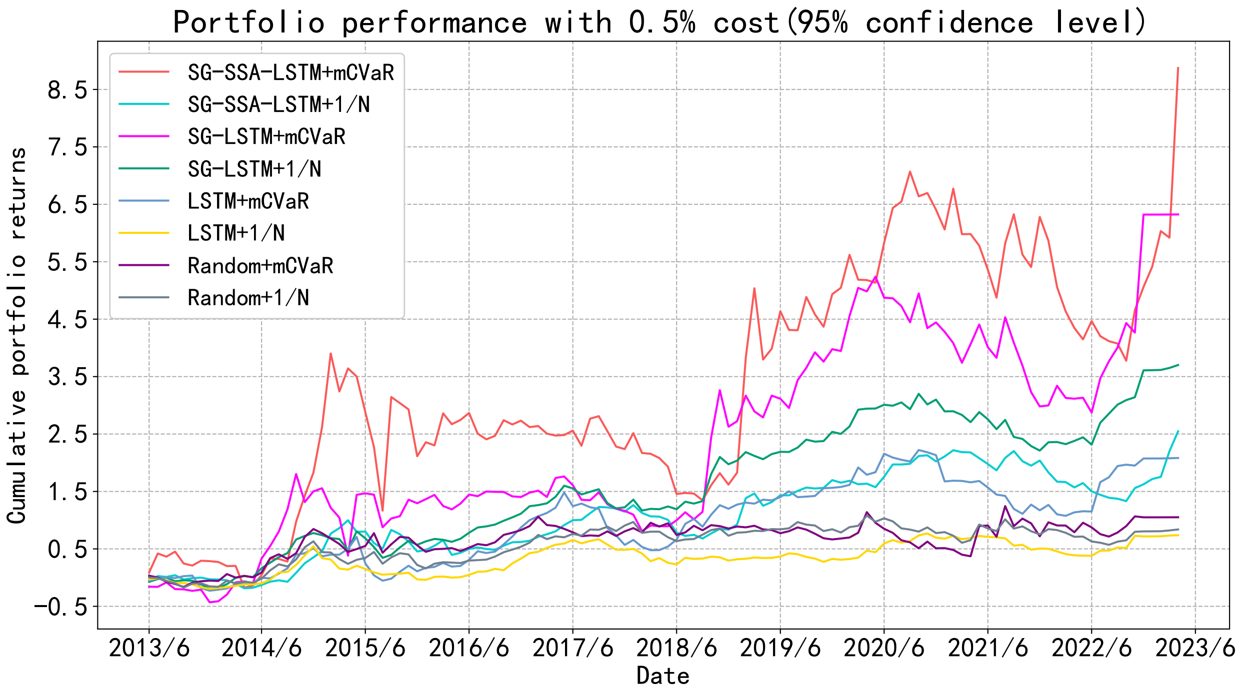

Fig. 9. Cumulative returns considering transaction costs (0.5%).

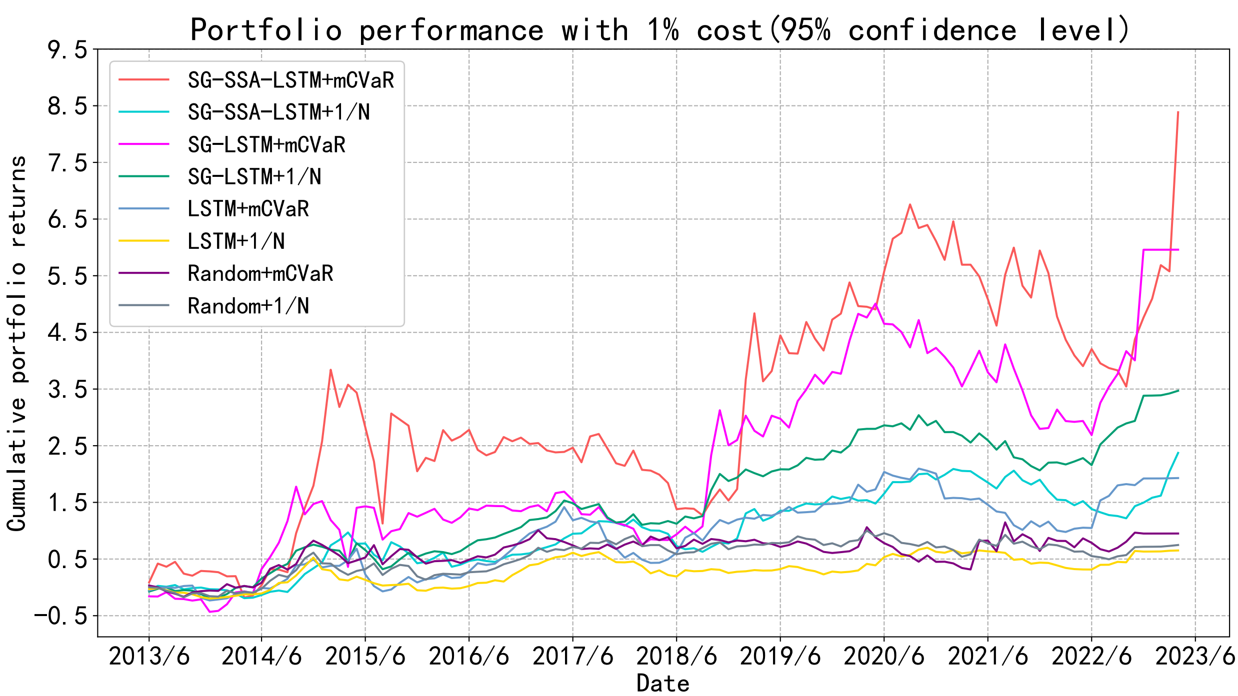

Fig. 10. Cumulative returns considering transaction costs (1%).

a declining trend. However, the cumulative returns of the SG-SSA-LSTM+mCVaR model consistently remain at the highest level. Figs. 9 and 10 display the cumulative return charts at the 0.95 confidence level. Taking the example of a 0.5% transaction cost and a 0.95 confidence level, the cumulative return rate of the SG-SSA-LSTM+mCVaR model reached 8.87. However, when the transaction costs were 0.5% and the confidence levels were 0.9 and 0.99, the cumulative return rates of the SG-SSA-LSTM+mCVaR model decreased to 8.16 and 7.12, respectively. In comparison, the performance of other benchmark models at a 0.5% transaction cost and a 0.95 confidence level is as follows: SG-SSA-LSTM+1/N at 2.54, SG-LSTM+mCVaR at 6.32, SG-LSTM+1/N at 3.70, LSTM+mCVaR at 2.08, LSTM+1/N at 0.73, Random+mCVaR at 1.05, and Random+1/N at 0.83. Meanwhile, when the transaction cost was 1% and the confidence level was 0.95, the cumulative return rate of the SG-SSA-LSTM+mCVaR model was 8.38. However, when the transaction costs were 1% and the confidence levels were 0.9 and 0.99, the cumulative return rates of the SG-SSA-LSTM+mCVaR model were 7.71 and 6.71, respectively. Furthermore, at a 1% transaction cost and a 0.95 confidence level, the cumulative return rates were as follows: SG-SSA-LSTM+1/N at 2.37, SG-LSTM+mCVaR at 5.95, SG-LSTM+1/N at 3.47, LSTM+mCVaR at 1.92, LSTM+1/N at 0.65, Random+mCVaR at 0.95, and Random+1/N at 0.74.

7. Conclusions

This paper proposes an two-Stage Stock Portfolio Optimization Based on Artificial Intelligence-Empowered Price Prediction and Mean-Conditional-Value-at-Risk Models. The LSTM combination prediction method based on SG filter and SSA optimization is used, and then combined with the mCVaR model for portfolio selection. After empirical analysis, we draw the following conclusions:

Firstly, in the initial phase, we applied the SG filter to denoise the stock price data. To optimize the hyperparameters of the neural network, the SSA was introduced. Experimental results demonstrated that the SSA algorithm exhibited the fastest convergence speed and highest accuracy when compared with GWO, SOA, DBO and PSO.

Secondly, SG-SSA-LSTM is developed to predict the stock price of candidate assets in the next period (month) and is used for portfolio selection. Furthermore, the results of SG-SSA-LSTM are compared with SG-GWO-LSTM, SG-SOA-LSTM, SG-LSTM, SSA-LSTM and LSTM. The research found that: (1) The SG-SSA-LSTM model attained an R^2 of 0.9980 and an accuracy of 0.9705, contributing superior asset inputs for portfolio selection. (2) The hyperparameter optimization within the combined model exhibits greater prediction accuracy than that of individual models.

Ultimately, the prediction outcomes are comprehensively evaluated based on potential return, prediction accuracy, and growth rate. The top K high-quality stocks are then chosen to construct the optimal investment portfolio. The research findings indicate that: (1) For individual investors, holding 7 different assets is a relatively suitable option. (2) In terms of average return, Sortino ratio, and performance risk, alternative portfolio models like SG-SSA-LSTM +mCVaR outperform Random+mCVaR and Random+1/N models, affirming the essentiality of integrating stock prediction into portfolio selection. (3) The model based on mCVaR performs better than the model based on 1/N. Under three different confidence levels, the cumulative returns of the combined prediction model and the mCVaR model reached 9.38, 8.63 and 7.54 respectively. This result is significantly better than seven benchmark models and effectively reduces investment risks. Even considering transaction fees of 0.5% and 1%, the model still achieved satisfactory returns of 8.87 and 8.38.

To sum up, the SG-SSA-LSTM+mCVaR model proposed in this study significantly outperforms benchmark methods in terms of average return, Sortino ratio, and performance risk. The mCVaR model is advantageous over traditional VaR and CvaR models as it more comprehensively considers losses in extreme scenarios, leading to more efficient asset allocation. The fundamental reasons for this advantage include: (1) It accurately captures future stock market characteristics. (2) The improved SG-SSA-LSTM model significantly enhances prediction accuracy, making the results more reliable. (3) The introduction of the mCVaR model optimizes portfolio allocation, effectively reducing risk, especially in handling extreme risk events and diversifying investment risk.

While this study offers valuable insights, there are still some limitations that require attention for further refinement and extension. Initially, we focused solely on a subset of asset data from the CSI 300 Index. However, considering the diversity in political environments and economic conditions across different regions, it is imperative to apply the SG-SSA-LSTM+mCVaR model to other capital markets to validate its universality and effectiveness. Additionally, in predicting stock returns, we only considered stock data as input features. Existing research indicates that multifaceted factors such as news, investor sentiment, economic conditions, and financial information can also impact short-term stock prices. Therefore, it is necessary to incorporate these factors into the input features to comprehensively assess their influence on stock price predictions.

In future research, integrating news, investor sentiment, and economic conditions into stock return prediction models will be a key focus to enhance their comprehensiveness and accuracy. The primary research directions include effective extraction of news information and its combination with market sentiment impacts, along with leveraging technical methods for the automatic analysis of economic data to construct investor sentiment indicators. Moreover, this model will be applied to FinTech and AI financial investment, with specific attention to its applications in automated investment decision-making and risk management. The impact on practitioners, such as improved decision-making efficiency, reduced human error, and optimized investment strategies, will also be explored.

CRediT authorship contribution statement

Chia-Hung Wang: Conceptualization, Preparation, Investigation, Formal analysis, Funding acquisition, Supervision, Writing – review & editing. **Yingping Zeng:** Methodology, Investigation, Data curation, Formal analysis, Visualization, Validation, Writing – original draft. **Jinchen Yuan:** Methodology, Preparation, Data curation, Validation, Visualization, Software, Writing – original draft.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This research was supported in part by the Fujian Provincial Department of Science and Technology, China, under Grant Number 2021J011070, and in part by the research fund from Fujian University of Technology, China under Grant GY-Z18148.

References

Abudurexiti, N., He, K., Hu, D., Rachev, S. T., Sayit, H., & Sun, R. (2023). Portfolio analysis with mean-cvar and mean-cvar-skewness criteria based on mean–variance mixture models. *Annals of Operations Research*, *336*, 945–966.

Ahmed, H. U., Mostafa, R. R., Mohammed, A., Sihag, P., & Qadir, A. (2023). Support vector regression (SVR) and grey wolf optimization (GWO) to predict the compressive strength of GGBFS-based geopolymer concrete. *Neural Computing and Applications*, *35*(3), 2909–2926.

Ali, M., Khan, D. M., Alshanbari, H. M., & El-Bagoury, A. A.-A. H. (2023). Prediction of complex stock market data using an improved hybrid EMD-LSTM model. *Applied Sciences*, *13*(3), 1429.

Balazs, A., Liski, E., Tuominen, S., & Kangas, A. (2022). Comparison of neural networks and k-nearest neighbors methods in forest stand variable estimation using airborne laser data. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, *4*, Article 100012.

Bathla, G., Rani, R., & Aggarwal, H. (2023). Stocks of year 2020: prediction of high variations in stock prices using LSTM. *Multimedia Tools and Applications*, *82*(7), 9727–9743.

Behera, J., Pasayat, A. K., Behera, H., & Kumar, P. (2023). Prediction based mean-value-at-risk portfolio optimization using machine learning regression algorithms for multi-national stock markets. *Engineering Applications of Artificial Intelligence*, *120*, Article 105843.

Bodnar, T., Lindholm, M., Niklasson, V., & Thorsén, E. (2022). Bayesian portfolio selection using VaR and cvar. *Applied Mathematics and Computation*, *427*, Article 127120.

Cai, J., Wang, C.-H., & Hu, K. (2024). LCDFormer: Long-term correlations dual-graph transformer for traffic forecasting. *Expert Systems with Applications*, *249*, Article 123721.

Chandar, S. K. (2022). Convolutional neural network for stock trading using technical indicators. *Automated Software Engineering*, *29*, 1–14.

Chen, J., Wen, Y., Nanehkaran, Y., Suzauddola, M., Chen, W., & Zhang, D. (2023). Machine learning techniques for stock price prediction and graphic signal recognition. *Engineering Applications of Artificial Intelligence*, *121*, Article 106038.

Chen, L., Zhang, W., & Ye, H. (2022). Accurate workload prediction for edge data centers: Savitzky-golay filter, CNN and BiLSTM with attention mechanism. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *52*(11), 13027–13042.

Chennaf, S., & Amor, J. B. (2023). Entropic value at risk to find the optimal uncertain random portfolio. *Soft Computing*, *27*, 15185–15197.

Dastgerdi, A. K., & Mercorelli, P. (2022). Investigating the effect of noise elimination on LSTM models for financial markets prediction using Kalman filter and wavelet transform. *WSEAS Transactions on Business and Economics*, *19*, 432–441.

Du, J. (2022). Mean–variance portfolio optimization with deep learning based-forecasts for cointegrated stocks. *Expert Systems with Applications*, *201*, Article 117005.

Dudukcu, H. V., Taskiran, M., Taskiran, Z. G. C., & Yildirim, T. (2023). Temporal convolutional networks with RNN approach for chaotic time series prediction. *Applied Soft Computing*, *133*, Article 109945.

Fan, Y., Sui, T., Peng, K., Sang, Y., & Huang, F. (2023). Study on load monitoring and demand side management strategy based on elman neural network optimized by sparrow search algorithm. *Circuit World*, *49*(1), 56–66.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Illa, P. K., Parvathala, B., & Sharma, A. K. (2022). Stock price prediction methodology using random forest algorithm and support vector machine. *Materials Today: Proceedings*, *56*, 1776–1782.

Jensen, M. C. (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics*, *6*(2/3), 95–101.

Jia, H., Xing, Z., & Song, W. (2019). A new hybrid seagull optimization algorithm for feature selection. *IEEE Access*, *7*, 49614–49631.

Jiang, W. (2021). Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, *184*, Article 115537.

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

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, *197*, Article 116659.

Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science*, *10*(1), 183–208.

Li, S., Wu, J., Jiang, X., & Xu, K. (2022). Chart GCN: Learning chart information with a graph convolutional network for stock movement prediction. *Knowledge-Based Systems*, *248*, Article 108842.

Liu, F., Qin, P., You, J., & Fu, Y. (2022). Sparrow search algorithm-optimized long short-term memory model for stock trend prediction. *Computational Intelligence and Neuroscience*, *2022*(1), Article 3680419.

Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, *33*, 4741–4753.

Ma, Y., Wang, W., & Ma, Q. (2023). A novel prediction based portfolio optimization model using deep learning. *Computers & Industrial Engineering*, *177*, Article 109023.

Maarif, M. R., Saleh, A. R., Habibi, M., Fitriyani, N. L., & Syafrudin, M. (2023). Energy usage forecasting model based on long short-term memory (LSTM) and explainable artificial intelligence (EAI). *Information*, *14*(5), 265.

Markowits, H. M. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 71–91.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Mohammed, A. R., Hassan, K. S., & Abdel-Aal, M. A. (2022). Moving average smoothing for gregory-Newton interpolation: A novel approach for short-term demand forecasting. *IFAC-PapersOnLine*, *55*(10), 749–754.

Mu, G., Gao, N., Wang, Y., & Dai, L. (2023). A stock price prediction model based on investor sentiment and optimized deep learning. *IEEE Access*, *11*, 51353–51367.

Oyewola, D. O., Akinwunmi, S. A., & Omotehinwa, T. O. (2024). Deep LSTM and LSTM-attention Q-learning based reinforcement learning in oil and gas sector prediction. *Knowledge-Based Systems*, *284*, Article 111290.

Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, *115*, 635–655.

Qin, D., Peng, Z., & Wu, L. (2023). Deep attention fuzzy cognitive maps for interpretable multivariate time series prediction. *Knowledge-Based Systems*, *275*, Article 110700.

Ran, X., Zhang, J., Zhu, W., Liu, K., & Liu, Y. (2024). A model of correlated interval–probabilistic conditional value-at-risk and optimal dispatch with spatial correlation of multiple wind power generations. *International Journal of Electrical Power & Energy Systems*, *155*, Article 109500.

Staino, A., Russo, E., Costabile, M., & Leccadito, A. (2023). Minimum capital requirement and portfolio allocation for non-life insurance: a semiparametric model with conditional value-at-risk (CVaR) constraint. *Computational Management Science*, *20*(1), 12.

Wang, C.-H., Cai, J., Ye, Q., Suo, Y., Lin, S., & Yuan, J. (2023). A two-stage convolution network algorithm for predicting traffic speed based on multi-feature attention mechanisms. *Journal of Intelligent & Fuzzy Systems*, *45*(3), 5181–5196.

Wang, J., Hu, Y., Jiang, T.-X., Tan, J., & Li, Q. (2023). Essential tensor learning for multimodal information-driven stock movement prediction. *Knowledge-Based Systems*, *262*, Article 110262.

Wang, W., Li, W., Zhang, N., & Liu, K. (2020). Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications*, *143*, Article 113042.

Wang, C.-H., Yuan, J., Zeng, Y., & Lin, S. (2024). A deep learning integrated framework for predicting stock index price and fluctuation via singular spectrum analysis and particle swarm optimization. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *54*, 1770–1797.

Wang, C.-H., Zhao, Q., & Tian, R. (2023). Short-term wind power prediction based on a hybrid Markov-based PSO-BP neural network. *Energies*, *16*(11), 4282.

Wu, Q., Liu, X., Qin, J., Zhou, L., Mardani, A., & Deveci, M. (2022). An integrated generalized TODIM model for portfolio selection based on financial performance of firms. *Knowledge-Based Systems*, *249*, Article 108794.

Wu, M.-E., Wang, C.-H., & Chung, W.-H. (2017). Using trading mechanisms to investigate large futures data and their implications to market trends. *Soft Computing*, *21*, 2821–2834.

Wu, D., Wang, X., & Wu, S. (2021). A hybrid method based on extreme learning machine and wavelet transform denoising for stock prediction. *Entropy*, *23*(4), 440.

Xie, Z., & Wang, Y. (2022). Exploration of stock portfolio investment construction using deep learning neural network. *Computational Intelligence and Neuroscience*, *2022*(1), Article 7957097.

Xue, J., & Shen, B. (2023). Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *Journal of Supercomputing*, *79*(7), 7305–7336.

Yan, Y., Zhang, C., Li, X., & Zhang, B. (2024). A framework for stock selection via concept-oriented attention representation in hypergraph neural network. *Knowledge-Based Systems*, *284*, Article 111326.

Yang, H., & Ferrer (2023). Explosive behavior in the Chinese stock market: A sectoral analysis. *Pacific-Basin Finance Journal*, *81*, Article 102104.

Yin, L., Li, B., Li, P., & Zhang, R. (2023). Research on stock trend prediction method based on optimized random forest. *CAAI Transactions on Intelligence Technology*, *8*(1), 274–284.

Yousaf, I., Anwar, F., Imtiaz, S., Almadhor, A. S., Ishmanov, F., & Kim, S. W. (2022). An optimized hyperparameter of convolutional neural network algorithm for bug severity prediction in alzheimer's-based IoT system. *Computational Intelligence and Neuroscience*, *2022*(1), Article 7210928.

Yu, S., Wang, H., & Dong, C. (2023). Learning risk preferences from investment portfolios using inverse optimization. *Research in International Business and Finance*, *64*, Article 101879.

Zhang, Y., Song, Y., & Wei, G. (2023). A feature-enhanced long short-term memory network combined with residual-driven ν support vector regression for financial market prediction. *Engineering Applications of Artificial Intelligence*, *118*, Article 105663.

Zhao, J., Yang, X., & Zhang, C. (2024). Vehicle trajectory reconstruction for intersections: an integrated wavelet transform and savitzky-golay filter approach. *Transportmetrica A: Transport Science*, *20*(2), Article 2163207.