ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

A New Genetic Algorithm with LSTM for Optimising Stock Investment Portfolio

Haoping Li[1] *and Xiaojing Zhu*[1,*]

[1] College of Mathematics and Physics, Shanghai University of Electric Power, Shanghai 200120, Shanghai, China

SUMMARY: *In this article, we bring forward a newly-made asset combination optimization method that combines Genetic Algorithms (GA) with Long Short-Term Memory (LSTM) networks for market prediction. LSTM networks are utilized by us to build a model that is based on past market changes, for the forecasting of the future earnings of assets. The forecasted numerical results of these targets help the GA to find out a portfolio which has more fitting allocation. This multi-objective method is able to assign the weights of risk and profit more effectively than one single-objective model. In this place, an adaptive genetic algorithm is put forward. It makes changes to the crossover and mutation operators on the basis of the fitness of the portfolio. Iterative methods can be utilized by people for the optimization of portfolio distribution. In the end, the model's performance is evaluated by means of risk-adjusted profits, steadiness, and out-of-sample operational results. We then carry out a comparison between it and the traditional models which are like mean-variance optimization. We carry out backtesting work, therefore, in order to further make confirmation of the robustness and the generality of this optimized combination of investment. Through making use of LSTM forecast and GA optimization method, an optimal portfolio may be obtained. Therefore, the function and adaptive ability of this novel method are better than what the traditional method has.*

KEYWORDS: *Genetic Algorithm, Multi-objective Portfolio Optimization, LSTM, Market Prediction, Adaptive Portfolio Allocation.*

1 Introduction

A typical problem in finance is how to optimise a stock-investment portfolio to reduce risk and enhance returns simultaneously [1-3]. To maximise the return on investment and reduce risk, a portfolio needs to be optimised; thus, it is necessary for good financial management at present [4, 5]. The first model to determine an ideal allocation among the above is the Markowitz mean-variance optimisation model [6-8], which considers both expected returns and asset covariances. The model assumes that the best A portfolio can be built by allocating different proportions of wealth among a group of assets in a way that reduces the risk (such as variance or standard deviation) of achieving a certain level of return. Although it is a reasonable foundation for portfolio management, it is not very practical due to various assumptions.

Although many simplifying assumptions are used in practice, such as assuming normally distributed returns, which do not reflect the complexity of real life [9-11], they are often taken

*xjzhu2013@shiep.edu.cn
https://doi.org/10.65102/is2026930 Published: 30 April, 2026

as given. Generally, the failure of asset returns to show the extremes observed in financial markets, such as stock market crashes or bubbles, which are far from normal, indicates that asset returns are normally distributed. Other reasons that change in the market at any time include technological innovations, changes in politics, and development of the economy. To reduce the problem of a single-objective trade-off between risk and return, serious issues in tail risk or downside risk that affect the sensitivity of a portfolio under financial stress have been ignored. A portfolio will be in good health during a good economy and will decline relative to others in the event of an economic crisis. Many times, these very high risks have been ignored in the traditional model, and the portfolio performance expectations are excessively optimistic.

Many of the current portfolio optimisation methods are either single-objective or linear in nature; thus, risk is usually considered the standard deviation or volatility of returns, and returns are only taken as the average of past returns. Although the above methods can be used to achieve high returns easily and are relatively simple to understand, they are not all-encompassing in dealing with various types of risks and fail to show a non-linear relationship among assets. A typical mean-variance approach does not consider the low-volatility objective of reduced risk in favour of a higher-volatility portfolio. However, the above assumption does not consider other factors in risk, such as credit risk, liquidity risk and the risk of substantial loss during a financial crisis. These models generally do not consider the risk-return trade-off.

Therefore, they are more sensitive to the time period selected and prone to overfitting. Due to the changes in the economic environment, the previous conditions may no longer be suitable for the company's new plans.

For solving the limited adaptive ability of traditional mean-variance models in changeable market conditions, a GA-LSTM portfolio optimizing framework has been designed by us. Different from the traditional methods which only rely on mean value and variance, this framework has the integration of a broader scope of risk data. It has consideration of downward risk as well as distribution features like skewness and kurtosis. This permits a more effective assessment of investment combination performance when market conditions are in tense states and when fat-tailed income distributions exist. This method brings together market prediction which is based on Long Short-Term Memory (LSTM) networks with optimization that uses the Genetic Algorithm (GA).

The LSTM part carries out the identification of time tendency in past asset prices, hence it produces the prediction of returns for the optimization stage. After that, the GA carries out exploration on portfolio weights among many targets, for instance, the Sharpe ratio, the Sortino ratio, and the maximum drawdown. This enables us to carry out a synchronous consideration of revenue, downward risk, and investment combination stability. For the promotion of the efficiency of the search procedure, the crossover and mutation operations are by the portfolio's fitness adaptively adjusted. This tactic can assist in the retention of diversity and hence prevent the occurrence of premature convergence. This framework has three important contributions that can not be ignored. First of all, it has put deep learning into the asset allocation procedure, hence permitting market predictions to guide the searching work for optimal investment combinations. Secondly, this thing builds a multi-goal optimization framework which provides a more overall evaluation of risk-adjusted incomes when compared with single-goal distribution methods. Thirdly, we have formulated a mechanism of adaptive GA.

This mechanism can promote the convergence velocity and the solution quality by means of crossover and mutation which are driven by fitness. The results of backtesting show that the framework we put forward has better performance than traditional portfolio optimization methods in the aspects of risk-adjusted earnings and stability under different market situations.

2 Related Work

2.1 Traditional Portfolio Optimization

The beginning of modern portfolio optimisation can be traced back to the first study by [12], which established the mean-variance theory and aimed to maximise returns at a certain level of risk. This model assumes normally distributed returns and a static covariance matrix, but its shortcomings in handling non-linear market fluctuations and tail risk have also been widely reported. Kurani and others performed an all-encompassing comparison experiment of Artificial Neural Networks (ANN) and Support Vector Machines (SVM) for stock prediction. The research by the above-mentioned individuals analysed the strengths and deficiencies of both methods for financial forecasting and specifically investigated in which areas each algorithm performed better in terms of performance [13]. Cao and others have used Support Vector Machines (SVM) for many classification and prediction problems in finance. Support Vector Machines (SVM) have been applied in many ways to finance prediction and perform well on complex data [14]. Han and Chen have applied Support Vector Machines (SVM) in combination with financial statement analysis for stock trend forecasting. By pulling in some key financial indicators, they have improved the accuracy of the model [15]. Cao and Tay have also done this and compared SVM with BP neural networks. SVM performed better in identifying the non-linear features of the financial data that often occur [16]. Later, Simian and others used to drive it. Further modify SVR by using an automatic way to optimise the kernel. It resulted in the prediction of the above items [17].

2.2 Multi-Objective Optimization in Finance

Multiple objectives for portfolio optimisation are relatively new; they hope to achieve multiple purposes simultaneously, such as reducing risk and increasing returns. El-Abbasy et al. (2020) have applied this to construct-ion project scheduling in a novel way. They did not think about how long the project would be carried out, how much credit to obtain, and what financing expenses would be incurred. Many experiments have been carried out to study the effectiveness of evolutionary algorithms, and it has been shown that they are suitable for optimisation in financial engineering and construction [18].

Doumpos and Zopounidis (2020) have also studied this problem, but they have done so on a larger scale in terms of the application of multi-objective models in finance and investment. They believe that these tools can help make more informed decisions on all kinds of trade-offs in finance; that is to say, there is no single aim to be pursued.

Coello's work is also older, but it has provided many foundations for others. He discussed the application of evolutionary algorithms in the multi-objective problem of finance and provided several actual cases that have been improved [19].

Fathi and Afshar are applying a Genetic Algorithm to this project schedule. The components of the model are project duration, credit requirements and funds. The same applies to multi-objective optimisation; it is also feasible in practice, and both financial planning and complex project management can be achieved simultaneously [20].

More recently, Alexandre and others (2023) have been studying a large-scale problem in financial systems: how to achieve both efficiency and stability. They used a multi-objective evolutionary algorithm to find an optimum point and found that one can be improved at the expense of a decrease in the other [21].

2.3 Deep Learning-Enhanced Optimization

Deep learning and evolutionary optimisation have begun to be combined more frequently in recent years for financial research. Shi and others (2025) have developed a method for applying deep learning to financial statements and, to reduce risk, have used this approach for that purpose. They applied NLP to deal with the various unstructured and messy text in the financial documents; otherwise, it would be very difficult to process. It can be seen where the risk lies in this way [22].

Huang (2024) has studied how to improve the optimisation of investment portfolios in traditional finance using machine learning. Based on the results above, machine learning can be used to improve the accuracy of forecasts for asset returns and address some limitations of the traditional model in investment management [23].

Liu and Pun attempted to do so differently. A two-step supervised learning setup was used to predict systemic risk; that is to say, it aimed to forecast the magnitude of equity losses for the financial system. By using a return time series, they were able to make more accurate predictions and thus anticipate a financial downturn earlier [24].

Hadad and his team carried out pairs trading with a machine-learning modification. They have optimised the selection and tuning of pairs to improve the sensitivity and, ideally, the profit of the strategy [25].

Wang and others have employed deep reinforcement learning for portfolio management. Dynamically adjust the portfolio weights of their models based on new information and pursue better returns constantly. It is one of those arrangements where the system learns by doing and gets better with use [26].

Recently, Joshi et al. have introduced a hybrid model of LSTM, GRU and CNN. It is a multi-objective portfolio optimisation model that seeks to maximise the sum of return, skewness and entropy, and minimise variance and kurtosis. It has a relatively large scale of application for hybrid models in financial decision-making [27].

3 Methodology

3.1 Model Overview

This research article has designed a portfolio optimization method that combines market prediction on the basis of Long Short-Term Memory (LSTM) networks with a searching for portfolio distributions by use of the Genetic Algorithm (GA). At the beginning stage, the historical market data are utilized to carry out the training of the LSTM model. This model which has completed training then produces signals of anticipated earnings, which are used in the optimization stage. These forecasted returning signals are afterwards put into the GA framework. The genetic algorithm carries out searching of portfolio distribution on the basis of multiple standards, for example, the Sharpe ratio, Sortino ratio, and maximum drawdown. In the process of optimization, candidate portfolios are evaluated according to their degree of adaptation. By means of a selection mechanism, the more beneficial asset combination distributions are kept. Crossover working steps are conducted for combining the features of high-achieving investment combinations. At the same time, the mutation is utilized by people for keeping the diversity and enlarging the searching scope. When the algorithm carries out its advancement via continuous generations, it accurately finds out portfolio allocations which display improved risk-adjusted performance. The concluding investment combination is assessed on the basis of risk-modified revenue, steadiness, and out-of-sample operational result. We carry out backtesting work in order to make a contrast between the GA-optimized portfolio and the traditional mean-variance method under many different kinds of market situations. This comparison helps us carry out the evaluation on the robustness and the practical application scope of the GA-optimized asset combination.

3.2 Market Prediction using LSTM

Long Short-Term Memory (LSTM) networks undertake the task to carry out prediction on market fluctuations on the basis of historical price sequences. As one kind of gate-controlled Recurrent Neural Network (RNN) structure, LSTM reduces the problem that gradient disappears, and hence it can find long-term connections in sequence-type financial data. Because asset yields have dependence on time, therefore historical price data are firstly processed into yield sequences. After that, these return sequences are put into the LSTM model for the prediction of the next period's returns, this step has key importance for portfolio optimization.

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}} \quad (1)$$

where r_t is the return of the asset at time t, and P_t is the price of the asset at time t.

The dataset is then divided into training and testing sets, a model is trained using the training set, and the accuracy on the test set is assessed. We normalise the data to have a similar range for the input features and thus improve the convergence and performance of the LSTM model. Min-Max normalization is a common method of scaling data to the range [0, 1], as shown below:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (2)$$

The standardized return sequences were put into sliding windows by us. Every window contains the newest portfolio income returns, and the income return of the next period is used as the target for prediction work. After that step, a LSTM network was undergone training. This network's goal was to carry out detection of non-linear time connections inside the return sequence by way of its memory units. These unit modules have stored precious historical data, have incorporated present information, and have produced concealed states for one-step-forward prediction works. In the training stage, the predicted earnings were compared against the real earnings. The objective was to make the average square error of the forecasted outcomes become as small as possible. For renewing the model's parameters and strengthening convergence when we meet noisy financial time-series data, Adam and RMSprop optimization algorithms have been utilized by us. After the training had been finished, the forecasting performance was measured through the comparison between the predicted portfolio returns and the actual real-world returns in the test set.

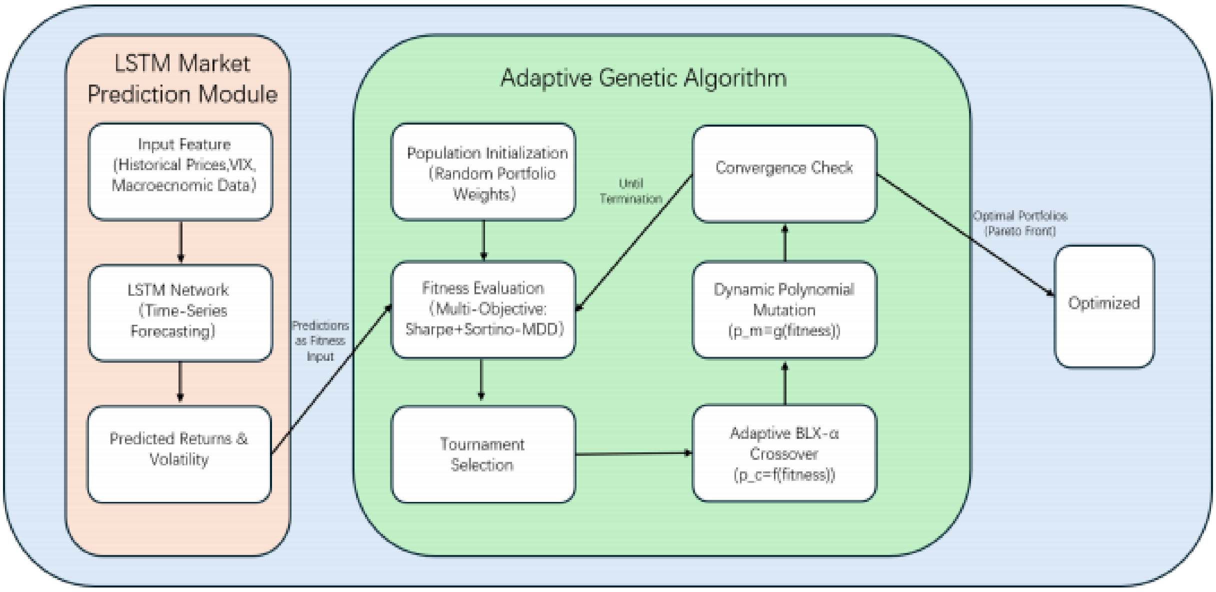

Figure 1: The LSTM–GA framework of portfolio optimization is got up to promote investment strategies. Within this framework, the Long Short-Term Memory (LSTM) model is utilized by us to carry out forecast work for multi-variable return data. At the same time, the self-adjusting Genetic Algorithm (GA) has a very important function in the evolving process of portfolio weight values. It realizes this thing by a sequence of operations, which include fitness-guided intersection, variation, and feedback mechanisms.

LSTM model has been trained and the predicted returns $\hat{r}$ The predicted outcomes are the expected future results for all the assets in this portfolio. Use the predicted returns as the objective function in a Genetic Algorithm to optimise the proportion of assets for a diversified portfolio. Use the predicted market environment as the initial conditions for a genetic algorithm, then optimise the portfolio under the new conditions in this algorithm.

3.3 Multi-Objective Optimization Framework

The multi-objective optimisation framework that will be employed in the portfolio optimisation method is introduced below. Not all the above methods have been employed so far; therefore, several performance indicators will be optimised simultaneously instead of only maximising returns or reducing risks. The first two indices of the model are the Sharpe ratio and the Sortino ratio; the third is maximum drawdown. By the above, we hope to construct a relatively balanced portfolio that is good for risk and return simultaneously.

A typical indicator for a low-risk-adjusted return is a low Sharpe ratio. It is the proportion of the excess return of a portfolio (that is, the return after deducting the risk-free rate) to the risk (standard deviation). Sharpe's Ratio Formula is shown above:

$$S = \frac{R_p - R_f}{\sigma_p} \tag{3}$$

R_p is the expected return of the portfolio, R_f is the risk-free rate, and σ_p is the standard deviation of the portfolio's returns.

The Sharpe ratio shows how much excess return per unit of risk a portfolio has. A relatively high Sharpe ratio indicates that the portfolio can achieve higher returns at the same level of risk.

Sortino ratio is close to the Sharpe ratio, but it only considers downside risk, or negative returns; therefore, it excludes total volatility. This index has a small change with fluctuations in the general volatility of returns and is thus less suitable for investors focused on capital losses. Sortino Ratio is as follows:

$$\text{Sortino Ratio} = \frac{R_p - R_f}{\sigma_d} \tag{4}$$

Rp is the expected return of the portfolio, R_f is the risk-free rate, and σ_d is the downside deviation, which is the standard deviation of negative returns.

Sortino ratio focuses on downside risk and is more suitable for assessing the risk-adjusted return of assets that seek to avoid losses.

The highest amount of decline from the highest point to the bottom of a series of prices is called the maximum drawdown (MDD). It is the maximum loss an investor would have suffered if they had invested at the highest point and then sold.

at the bottom end. The formula for the largest peak-to-trough is as follows:

$$MDD = \frac{max(P_t) - min(P_t)}{max(P_t)} \tag{5}$$

Pt is the portfolio value at time t, $max(P_t)$ is the maximum portfolio value in this period, and $min(P_t)$ is the minimum portfolio value in this period.

A smaller maximum drawdown shows that the risk of large losses for a portfolio is relatively low and it is stable.

Traditional portfolio optimisation has only one objective function for evaluating the performance of a portfolio. In a multi-objective way, we want to improve several indicators at the same time. A single objective function can be constructed by combining the above indicators through Pareto-front optimisation.

Multi-objective optimisation has been used to establish the objective function as a vector.

$$F = [f_1, f_2, f_3] \tag{6}$$

where f_1 is the Sharpe ratio, f_2 is the Sortino ratio, and f_3 is the negative of the maximum drawdown (as we wish to minimise drawdown).

The objective of the optimisation is to find a Pareto-optimal set of solutions, where an increase in one objective cannot be achieved at the expense of another. A solution x^* is Pareto optimal if there is no other solution x such that:

$$\begin{gathered} \forall i \in \{1,2,3\}, f_i(x) \leq \mathrm{f}_i(x^*)\ \exists j \in \{1,2,3\} \\ such\ that\ f_j(x) < \mathrm{f}_j(x^*) \end{gathered} \tag{7}$$

Therefore, it cannot be said that one of these portfolios is superior to all others across all these goals. The set of all Pareto-optimal solutions constitutes the Pareto front and shows how to trade off between different goals.

Pareto-front optimisation can be used to select among multiple objectives. For example, a portfolio with a high Sharpe ratio may have a relatively high return, but it is also more volatile; a portfolio with a low Sharpe ratio may have a lower risk but also a lower return. The Pareto front can be used to plot and select a portfolio that is optimal for balancing the above conflicting purposes.

A Genetic Algorithm (GA) can be used to find a Pareto-optimal solution. In each generation of the GA, we keep a diverse population of portfolio allocations and select the best-performing portfolios according to the multi-objective criteria. The GA converges to a

Pareto front after some time, and a set of good risk-return trade-off portfolios is obtained for decision-makers.

The multi-objective optimisation problem can be formulated as follows:

$$minF(x) = [f_1(x), f_2(x), f_3(x)] \tag{8}$$

where x is the vector of portfolio weights (the decision variables). Find the portfolio allocation x that minimizes the objective vector F(x) subject to the following portfolio constraints:

$$\sum_{i=1}^{N} x_i = 1, x_i \geq 0 \forall i \tag{9}$$

x_i is the weight of asset i in the portfolio, and N is the number of assets in the portfolio. The constraints ensure that the portfolio is fully allocated (the sum of weights equals 1) and that the weights are non-negative; that is to say, only long-only positions in each asset are allowed.

A multi-objective optimisation framework can be used to assess the all-around risk and return of a portfolio better. The Sharpe ratio, Sortino ratio and maximum drawdown are all commonly used measures in the optimisation of a portfolio to consider both the risk and return of the portfolio. Pareto-front optimisation is used to show the trade-off among various goals and presents a group of Pareto-optimal solutions that the decision-maker can select from in terms of investment decisions.

3.4 Adaptive Genetic Algorithm Operations

The self-adjusting heredity algorithm (GA) through selection, crossover, mutation produces the potential investment combinations. When compared with GAs that have fixed parameters, it makes modification to crossover and mutation probabilities on the basis of the population's fitness. This method has already hold a balance between exploration and exploitation, therefore it can reduce the possibility of early convergence. Crossover is that it merges two parent portfolios to produce offspring, hence it maintains diversity. A fixed crossing probability can bring about either overmuch or inadequate searching of the solution space, the proposed method defines a generation-specific $P_c(t)$ based on the average population fitness $F_{avg}(t)$.

$$P_c(t) = P_{\omega+} + (P_{\omega\_} - P_{\omega\_}) \cdot \frac{P_{\omega_{-1}}(t)}{P_{\omega_{-1}}(t)} \tag{10}$$

Pmin and Pmax are the minimum and maximum crossover probabilities, Favg(t) is the average fitness of the population at generation t, and Fmax(t) is the maximum fitness of the population at generation t.

When the population is relatively homogeneous (that is, the mean fitness is high), the algorithm will be more likely to explore locally for a longer time and thus find a good solution more easily. On the other hand, when the population is heterogeneous (that is, the fitness variation is large), the algorithm will increase the probability of crossover and combination to explore.

A random perturbation is added to the existing solution, and then a new area in the solution space is searched. Mutation helps avoid early convergence to a local optimum and keeps the diversity of the population. Pm is the probability of a mutation. Traditional GAs have a fixed mutation probability, which may be either too high in terms of randomness (and thus slow convergence) or too low (and thus stagnant search).

We put forward an adaptive mutation probability Pm(t) that changes according to the fitness of the population:

$$P_m(t) = P_m + (P_m - P_m) \cdot (1 - \frac{F_{m_1}(t)}{F_{m \times n}(t)}) \tag{11}$$

where Pmin is the minimum mutation probability, Pmax is the maximum mutation probability, Favg(t) is the average fitness of the population at generation t, and Fmax(t) is the maximum fitness of the population at generation t.

As the population becomes more uniform (i.e., as the average fitness increases), the mutation probability is reduced to allow for fine-grained tuning and exploitation of existing solutions. At the same time, if there are more different individuals in the population (that is to say, a larger range of fitness values), the mutation rate will be relatively high to help explore new areas.

One of the first problems to be solved by genetic algorithms is how to avoid losing good solutions that have already been found in previous iterations. Therefore, we use an elitism approach to ensure that the top-performing individuals of the current generation are passed on to the next generation unchanged. Therefore, the algorithm can keep the good solutions and increase the speed of convergence.

Let x* be the optimal portfolio of the current generation. The strategy of elitism directly copies x* to the next generation:

$$x_{elite}(t+1) = x^*(t) \tag{12}$$

where $x_{elite}(t+1)$ is the elite individual of the next generation, and $x^*(t)$ is the best individual in the current generation. Elitism is a simple technique that preserves the best-so-far solutions generated by randomness. Otherwise, the good portfolios may not be included in selection, crossover or mutation. We use Nelite to set how many of the top ones we will keep.

Our algorithm does not use fixed crossover and mutation rates. We will not adjust them dynamically. If the people are still scattered all over the place, we'll increase crossover and try new things. When the contents are too similar, we will be reduced in quantity. Mutation also follows the same logic: more for greater diversity, less when it has become stable.

Thus, we can continue to explore new fields and keep the successful ones as well. It is faster, but it may also fail to avoid a poor local optimum early on. In short, it helps all the portfolio search functions work normally without much effort.

3.5 Incorporating Deep Learning for Market Prediction

The standardized return sequences were put into sliding windows by us. Every window contains the newest portfolio income returns, and the income return of the next period is used as the target for prediction work. After that step, a LSTM network was undergone training. This network's goal was to carry out detection of non-linear time connections inside the return sequence by way of its memory units. These unit modules have stored precious historical data, have incorporated present information, and have produced concealed states for one-step-forward prediction works. In the training stage, the predicted earnings were compared against the real earnings. The objective was to make the average square error of the forecasted outcomes become as small as possible. For renewing the model's parameters and strengthening convergence when we meet noisy financial time-series data, Adam and RMSprop optimization algorithms have been utilized by us. After the training had been finished, the forecasting performance was measured through the comparison between the predicted portfolio returns and the actual real-world returns in the test set.

$$R_p = \sum_{i=1}^{N} x_i \cdot \hat{r}_{i,t+1} \tag{13}$$

where x_i is the weight of asset i in the portfolio, $\hat{r}_{i,t+1}$ is the predicted return for asset i, and N is the number of assets in the portfolio.

Add the predicted return to the fitness function of the genetic algorithm and evaluate the multiple objectives of the portfolio. The objective function is a vector containing the Sharpe ratio S, Sortino ratio Sortino, and negative maximum drawdown -MDD, as we wish to maximise returns and minimise risks.

$$F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})] \tag{14}$$

where: $f_1(\mathbf{x})$ = Sharpe ratio, $f_2(\mathbf{x})$ = Sortino ratio,$f_3(\mathbf{x})$ = —Maximum drawdown.

The genetic algorithm finds trade-off solutions to the two goals of maximising and minimising at the Pareto front.

-imitation. Predicted returns i,t+1 determine the evolution of GA.

Portfolio allocation that can meet the demand for high returns at a reasonable risk level; this portfolio optimisation problem is shown in Algorithm 1.

LSTM predicts the future market environment and adjusts the asset allocation in the optimisation of the genetic algorithm. Most traditional portfolio models assume that the expected return will not change significantly, or worse, they use the average of past data. That will not be convenient in the future. Therefore, in addition to history, a real-time prediction from an LSTM model will be used. This provides a future-looking objective for the optimisation process.

We pass the LSTM-predicted returns directly to the genetic algorithm. In this way, when GA is searching for portfolios, it will not be based on old figures but will adjust according to what is likely to occur. It will help the algorithm find a set of weights that maximise the return-to-risk ratio by considering the scale of the losses on both sides of the distribution.

The predicted values are added to the penalty term of the loss function. So GA is using forward-looking data for all of its portfolio assessments now. It will be more adaptable and less likely to be mistaken for the real thing. Deep learning can incorporate context, and GA can explore all kinds of options; thus, together they can help us build portfolios that are more responsive to changes in the market rather than those based on past performance.

4 Experimental Analysis

4.1 Dataset

We will use the yfinance Python API to retrieve the main equity data, and then acquire the daily price and volume information of the S&P 500.

Algorithm 1 Dynamic Multi-objective GA-LSTM Algorithm

Input: Historical prices P, Sequence length L, Hidden layer dimension H, Population size Psize, Number of iterations, Adaptive probability range

Output: Optimal asset weight vector w*, performance metrics

1: Compute the return on assets R

2. Divide the training set into Rtrain and the test set Rtest.

3: Use Min-Max scaling:$R_{norm} = \frac{R - \min(R)}{\max(R) - \min(R)}$

4: Build the LSTM input sequence

5. Initialize the LSTM model
6: Train LSTM and Predict Future Returns
7. Initial Population
8: for each t in [1, G] do
9: for each w in population do
10. Calculate expected return, Sharpe ratio, Sortino ratio, maximum drawdown and fitness.
11: end for
12: Compute the mean fitness and maximum fitness of the population in generation
13. Calculate (16) and (17).
14: Execute Tournament Selection
15. Execute BLX-α Crossover Operation
16: Execute mutation operation
17. Update the Population
18: End for
19: Select the comprehensive optimal solution w*
20: Evaluate Other Indices
21: return Outputs

Stocks from 2000 to 2023. To keep the list short, we will only include names with trading volumes of over 500,000 shares in the last day. Financial stocks (SIC 6000-6999) are also excluded to avoid the typical sector concentration they tend to cause.

The dataset covers the day-to-day price fluctuations and trading volumes well enough for our purposes of short-term prediction and long-term portfolio management. We are only using the daily level here, no minutes. The prices and volumes of the data are not normalised, and only the adjusted close price is normalised. Most of the signal will also be there.

YFinance is the main source, but we have also verified that the data it provides matches that from Alpha Vantage. They do; any gaps are less than 0.5%, so they are suitable for our purposes. In short, this arrangement will give us a relatively stable and small set of data, and we will avoid many problems on the vendor's end. This dataset can be used to build a model that connects micro-level asset pricing signals with changes in the broader market to perform in-depth anomaly detection in the market and optimise portfolios.

4.2 Experimental Setup

We have carried out the implementation of this framework by the use of Python 3.9. We have utilized PyTorch 2.0 to carry out the training work of the Long Short-Term Memory (LSTM) model, and thus we have used DEAP 1.3.3 for the optimization work of the Genetic Algorithm (GA). The LSTM model is constituted by three layers with gate structures. These layer structures respectively contain 128, 64, 32 hidden units. We have carried out the application of a Tanh activation function. We have the input windows which are set as a 60-day period, the batch size is 64, and 80 percent of the data are assigned for the training work. The procedure of training can be prolonged to reach as many as 300 epochs. We have utilized the AdamW optimizer, which has a learning rate of 0.001 and a weight decay of 1e - 4. The early stopping mechanism was thus activated after 15 successive verification epochs had not displayed any enhancement. In the genetic algorithm, every generation gives birth to 500 investment combinations, and the algorithm has the operation of 200 cycles. The speed of mutation underwent a decrease from 0.20 to 0.05. The BLX-alpha crossover point we have set is 0.7, and its alpha numerical value is 0.3. The fitness function that we use has combined Sharpe ratio, maximum drawdown, and Sortino ratio, the weight values are 0.35, 0.25, and 0.40 respectively. For the backtesting work, the walk-forward verification method was utilized, and the rebalancing operation was conducted on every three month basis. Our training work obtained support from NVIDIA A100 GPUs which use CUDA 11.8 mixed precision. The hyperparameter experiments in number 100 were carried out by using Optuna 3.2.0. We carried out the econometric tests by the utilization of HC3 standard errors together with Bartlett kernel smoothing method.

4.3 Experimental Results

Table 1 makes it clear that the GA-LSTM model gives the strongest whole portfolio working effect when it is compared with the GA, LSTM-only, Mean-Variance, Min-Variance, Equal-Weight, and Risk-Parity models. It has obtained the biggest annualized callback, which reaches 9.53%, and the biggest Sharpe ratio, which is located at 1.83. These numerical results exceed the corresponding performance of the traditional GA model (7.14% return and 1.21 Sharpe ratio) and the Mean-Variance model (6.51% return and 1.19 Sharpe ratio). The biggest fall degree of the GA-LSTM model is 12.3 percent, this is near the number of the conservative allocation models, and obviously lower than that of the traditional GA model (18.9%) and the Mean-Variance model (14.0%). Although its volatility is 13.3%, this is not the lowest value, the higher income therefore offsets this risk position. The GA-LSTM model also records the best risk-revenue ratio, which is 0.714, and possesses a Conditional Value-at-Risk (CVaR) of -22.5%. This therefore indicates that it possesses stronger tail-risk controlling capability when compared with the traditional GA and Mean-Variance models. Nevertheless, the Min-Variance, Equal-Weight, and Risk-Parity models still hold more conservative properties when we talk about the problem of extreme-loss exposure. Figure 2 has proven that the same tendency exists in the rolling 12-month performance. The GA-LSTM model can maintain higher accumulated earnings and stronger resistance ability during the 2020 COVID-19 market decline and the 2022 market pressure period. The research results indicate that combining LSTM forecast with GA distribution promotes portfolio earnings, risk-adjusted achievement, and the capability to get used to market fallings when put together with static or only-GA optimization.

4.4 Ablation Study Analysis

Figure 3 gives the experiment results of cutting out irrelevant parts for the GA-LSTM portfolio model. This all-round model combines the LSTM-based profit prediction together with the GA-based asset combination optimization. In this course, the GA brings into consideration the Sharpe ratio, Sortino ratio, and maximum drawdown for reaching a balance. It obtains the most excellent comprehensive performance, having an annualized earning rate of 9.53%, a Sharpe ratio of 1.83, and a relatively low maximum drawdown of - 0.123. After the LSTM module is taken away, the model returns to an allocation method which depends on historical average values or other basic income estimations. Under the condition that forward-looking market indicators do not exist, the annualized return has a decrease to 7.04%, the Sharpe ratio has a decline to 1.17, and the maximum drawdown has an expansion to - 0.157. This very clearly shows that market forecast has a direct function in increasing earnings and managing downward risks. The model that has no multi-objective optimization removes the combined risk-return restriction and puts focus on optimizing one single objective. This conduct makes the balance between earning, fluctuation and capital decrease become weaker. Therefore, Figure 3 has the verification that both LSTM forecasting and multi-objective GA optimization are essential components for the entire model to reach its performance level.

Table 1: Performance Metrics of Various Portfolio Optimisation Models

Model	Annualized Return	Sharpe Ratio	Max Drawdown	Volatility	Risk-Return Ratio	CVaR (95%)
GA-LSTM	0.0953	1.83	-0.123	0.133	0.714	-0.225
Conventional GA	0.0714	1.21	-0.189	0.124	0.574	-0.264
LSTM-only	0.0774	0.82	-0.155	0.145	0.366	-0.241
Mean-Variance	0.0651	1.19	-0.140	0.138	0.471	-0.285
Min-Variance	0.0453	0.63	-0.115	0.112	0.402	-0.200
Equal-Weight	0.0556	0.75	-0.130	0.120	0.540	-0.215
Risk-Parity	0.0501	0.71	-0.125	0.118	0.558	-0.210

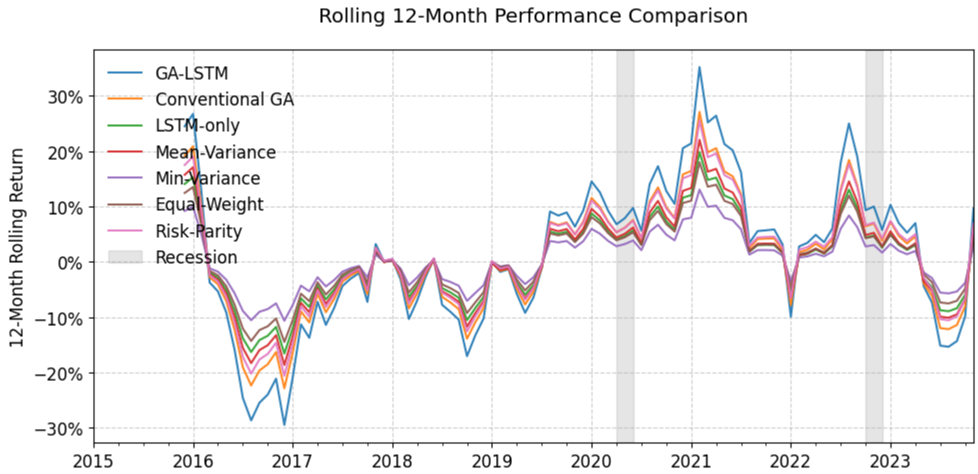

Figure 2: This analysis makes comparison on the accumulated income in a moving 12-month cycle for six combination tactics that cover the years from 2015 to 2023. In these kinds of strategies, the GA-LSTM method shows the topmost effect, particularly in the depression stages which are confirmed by the NBER.

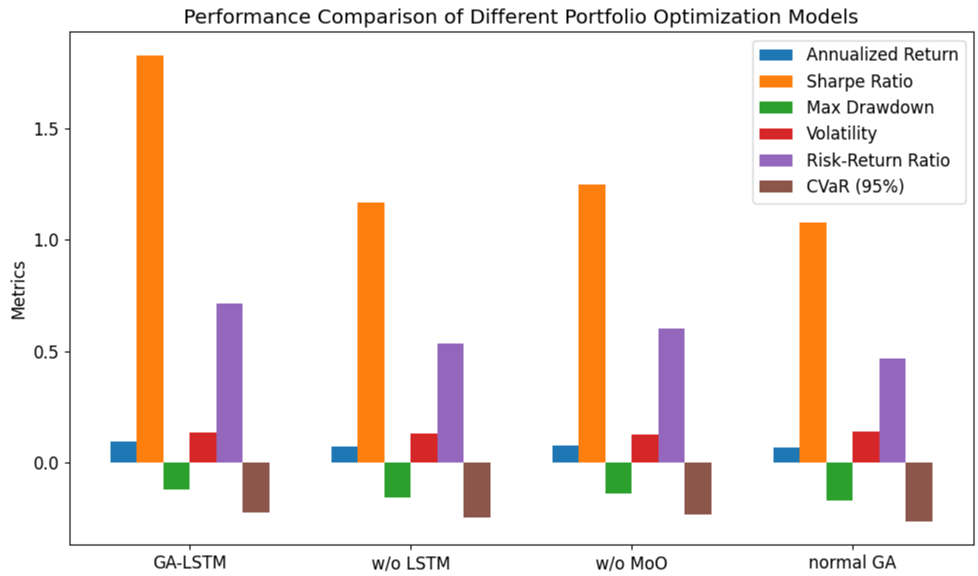

Figure 3: Performance comparison of GA-LSTM and its ablation variants. The models include the full GA-LSTM, w/o LSTM, w/o MoO, and normal GA, evaluated by annualised return, Sharpe ratio, maximum drawdown, volatility, risk–return ratio, and 95% CVaR.

The variant which has no MoO obtains better performance than the standard genetic algorithm (GA) and that which has no the long short - term memory (LSTM). This instrument obtains an annualized earning rate of 7.45% and a Sharpe proportion of 1.25. But, its most big retreat degree is located at -0.140, which is more high than that of the whole GA-LSTM model. This hence indicates that depending only on prediction of return is not sufficient when goals of risk and return are not together limited. The standard GA model shows the most bad performance. This thing possesses an annualized profit return of 6.51 percent, a Sharpe ratio of 1.08, and the most great maximum drawdown that equals minus 0.173. This result shows that its control to the downward risk is not effective when the market situation is not good. Figure 3 has proven that the complete GA-LSTM model obtains benefits from both LSTM prediction and multi-objective GA optimization. Getting rid of the LSTM thus damages the capability for adapting market allocations, hence taking away multi-objective optimization reduces the risk-control capability. Because the standard GA model does not have these two elements at the same time, therefore it displays the most unfavorable balance between earnings and drawdowns.

5 Conclusion

We are exploring a novel approach for portfolio optimisation in this paper by combining a Genetic Algorithm (GA) with LSTM-based market prediction and multi-objective tuning. The idea is relatively simple: LSTM predicts the movement of the market to some extent, and GA optimizes the weights in this prediction to maximise the Sharpe Ratio, Sortino Ratio, and minimise drawdowns simultaneously. The above arrangement is relatively flexible and can better handle market noise compared with the former method.

The result was not favourable. Both GA-LSTM and the old methods have shown certain strengths, but GA-LSTM is better at dealing with risks and fluctuations. We also conducted ablation studies, and yes, both the LSTM and the multi-objective parts are performing reasonably well here. Remove one and decrease performance.

However, it is not ideal. We are still based on old data, so if something strange appears in the market that is not in the history, the model will not recognise it. The whole setup will be relatively heavy on computing power, and if we need to expand or run it in real time, this will be even worse.

Next, we will add other kinds of assets, such as cryptocurrencies, and see if they are suitable here. We would like to reduce the computation cost and explore reinforcement learning to see if we can have more intelligent decisions on the fly. There may be some room to try using other kinds of deep-learning models to see if they can achieve higher accuracy in predictions.

Funding

This work was supported by National Natural Science Foundation of China (Grant No.12271342).

Aboutthe Author

Haoping Li was born in Leshan, Sichuan, P.R. China, in 2000. He received the bachelor's degree from Shanghai University of Electric Power, P.R. China. Now, he studies in College of Mathematics and Physics, Shanghai University of Electric Power. His research interests include matrix optimization, manifold optimization and machine learning.

Xiaojing Zhu was born in Shanghai, P.R. China. He received the doctor's degree from Tongji University, P.R. China. Now, he works in College of Mathematics and Physics, Shanghai University of Electric Power. His research interest include optimization on manifolds, matrix and tensor optimization.

References

[1] Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2014). 60 years of portfolio optimization: Practical challenges and current trends. European Journal of Operational Research, 234(2), 356-371.

[2] Maringer, D. G. (2005). Portfolio management with heuristic optimization (Vol. 8). Springer Science & Business Media.

[3] Lintner, J. (1975). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. In Stochastic optimization models in finance (pp. 131-155). Elsevier.

[4] Dahl, H., Meeraus, A., & Zenios, S. A. (1993). Some financial optimization models: I. Risk management. Financial Optimization, 3-36.

[5] Fabozzi, F. J., Kolm, P. N., Pachamanova, D. A., & Focardi, S. M. (2007). Robust portfolio optimization and management. John Wiley & Sons.

[6] Fahmy, H. (2020). Mean-variance-time: An extension of Markowitz's mean-variance portfolio theory. Journal of Economics and Business, 109, 105888.

[7] Chaweewanchon, A., & Chaysiri, R. (2022). Markowitz mean-variance portfolio optimization with predictive stock selection using machine learning. International Journal of Financial Studies, 10(3), 64.

[8] Gökgoz, F., & Atmaca, M. E. (2012). Financial optimisation in the Turkish Electricity Market: Markowitz's Mean-Variance Approach. Renewable and Sustainable Energy Reviews, 2020, 132, 109871.

[9] Rezaei Pandari, A., Azar, A. and Rayati Shavazi, A. (2012). Genetic algorithms for Portfolio Selection Problems with Non-Linear Objectives. African Journal of Business Management, 6(20), 6209.

[10] Zandi, G., Torabi, R., Mohammad, M. A., & Jia, L. (2021). Research on stock portfolio based on time series prediction and multi-objective optimization.

[11] Banihashemi, S., Moayedi Azarpour, A., & Navvabpour, H. (2016). Portfolio optimization by mean-value at risk framework. Applied Mathematics & Information Sciences, 10(5), 1935-1948.

[12] Markowitz, H. M. (1999). The early history of portfolio theory: 1600-1960. Financial Analysts Journal, 55(4), 5-16.

[13] 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.

[14] Cao, B., Zhan, D., & Wu, X. (2009). Application of SVM in financial research. In 2009 International Joint Conference on Computational Sciences and Optimization (Vol. 2, pp. 507-511). IEEE.

[15] Han, S., & Chen, R.-C. (2007). Using SVM with financial statement analysis for prediction of stocks. Communications of the IIMA, 7(4), 8.

[16] Cao, L., & Tay, F. E. H. (2001). Financial forecasting using support vector machines. Neural Computing & Applications, 10, 184-192.

[17] Simian, D., Stoica, F., & Bărbulescu, A. (2020). Automatic optimized support vector regression for financial data prediction. Neural Computing and Applications, 32, 2383-2396.

[18] El-Abbasy, M. S., Elazouni, A., & Zayed, T. (2020). Finance-based scheduling multi-objective optimization: Benchmarking of evolutionary algorithms. Automation in Construction, 120, 103392.

[19] Coello Coello, C. A. (2006). Evolutionary Multi-objective Optimisation and its Application in Finance. In Handbook of Research on Nature-Inspired Computing for Economy and Management. Idea Group Publishing.

[20] Fathi, H., & Afshar, A. (2010). GA-based multi-objective optimization of finance-based construction project scheduling. KSCE Journal of Civil Engineering, 14(5), 627-638.

[21] Alexandre, M., Michalak, K., Silva, T. C., & Rodrigues, F. A. (2023). Efficiency-stability trade-off in financial systems: A multi-objective optimization approach. Physica A: Statistical Mechanics and its Applications, 629, 129213.

[22] Shi, X., Zhang, Y., Yu, M., & Zhang, L. (2025). Deep learning for enhanced risk management: A novel approach to analyzing financial reports. PeerJ Computer Science, 11, e2661.

[23] Huang, S. (2024). Advancing portfolio optimization: The convergence of machine learning and traditional financial models. Applied and Computational Engineering, 57, 206-211.

[24] Liu, R., & Pun, C. S. (2022). Machine-learning-enhanced systemic risk measure: A two-step supervised learning approach. Journal of Banking & Finance, 136, 106416.

[25] Hadad, E., Hodarkar, S., Lemeneh, B., & Shasha, D. (2024). Machine learning-enhanced pairs trading. Forecasting, 6(2), 434-455.

[26] Wang, J., Hai, M., & Li, H. (2024). A deep reinforcement learning model for portfolio management based on weight adjustment. Procedia Computer Science, 242, 356-363.

[27] Joshi, N. K., & Dhodiya, J. M. (2025). Intelligent many-objective portfolio optimization

using hybrid deep learning and evolutionary algorithm approach for advanced decision-making. Computers & Industrial Engineering, 205, 111159.