ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Convolutional Neural Networks and Deep Learning in Quantitative Financial Investments

Wangqi Shen[1,*] *and Wenxiao Zhuo*[2]

[1] School of Economics and Management, Wuhu Institute of Technology, Wuhu, Anhui, 241000, China
[2] School of Food and Bioengineering, Wuhu Institute of Technology, Wuhu, Anhui, 241000, China

SUMMARY: *With the increasing complexity of financial markets, traditional analytical tools are overwhelmed by the amount of information available and the ever-changing markets. Quantitative investment utilizes mathematical models and statistical methods to organize past data, which to a certain extent can grasp the timing of trading and avoid investors' sentimentality and blindness. Convolutional neural networks were initially widely used in the field of computer vision due to their powerful image feature extraction and recognition capabilities, and have gradually gained popularity among financial researchers. The complex nonlinear and potential relationships hidden in financial market sequences are difficult to be described by traditional linear models, and the convolutional neural network can automatically mine the local and global information embedded in the stock price, volume and technical indicators through its multilevelized feature extraction capability. In this paper, we constructed a portfolio strategy based on the combination of convolutional neural network and long and short-term memory network, used multi-dimensional financial market data as input features to train the model, and evaluated the model based on historical backtesting and out-of-sample testing. The model developed in this paper achieves an out-of-sample prediction accuracy of 75.3% for CSI 300 stocks, an annualized return of 26.8%, a Sharpe ratio of 1.84, and an information ratio of 1.28, which are all better than the results of traditional random forest and support vector machine methods; the results of this paper show that deep learning can be widely used in the field of quantitative investment, and to a certain extent, improve portfolio stability and return.*

KEYWORDS: *quantitative investment; convolutional neural network; deep learning; long and short-term memory network; financial time series*

1 Introduction

With the arrival of the era of big data and artificial intelligence, deep learning has gradually become the mainstream of the times. In the field of financial investment, intelligent investment advisors, fund stock selection, commodity trading advisor strategies and high-frequency trading have all utilized deep learning methods, becoming a new trend in quantitative investment. Fang et al. created a deep learning hybrid model to predict the prices of 50 exchange-traded fund options and combined it with a delta hedging strategy to control the risk, which is a deep learning quantitative strategy with higher returns, smaller retracements, and better risk tolerance

*Swqkai1234@163.com
https://doi.org/10.65102/is2026243
Published: 30 April, 2026

than the traditional approach [1]. Yao et al. compare the performance of deep learning and traditional time series methods in quantitative trading of gold and bitcoin, and the results show that the deep learning model identifies strategies with a fitting accuracy of over 73%, with significantly better returns than the traditional strategies, and remains stable under changes in transaction costs [2]. Li et al. fused news text and technical factors to design a deep learning quantitative model to predict stock prices by integrating semantic information from news headlines and content, and achieved annualized returns and maximum retracement rates of 32.06% and 5.14%, respectively, in the A-share market [3]. Based on A-share prices, financial and technical indicators, Li and Hong combine machine learning and deep learning to predict returns and construct two strategies of maximizing returns and risk balance to provide quantitative investment decision references for investors with different risk preferences [4].

Quantitative financial investment relies on statistical and econometric methods to establish appropriate strategies and to obtain investment profits by means of computerized automated (semi-automated) trading, and is a product of the combination of statistics, computers and finance disciplines [5]. Early quantitative investment strategies focused on methods such as linear regression models, but it is difficult to handle and capture multimodal data and nonlinear features in financial markets [6-8]. Convolutional neural network (CNN) is a deep learning model with two features of local connectivity and weight sharing in simulation time, which is suitable to be used for processing data with lattice-like topology, such as images, videos, and sounds [9]. CNN has the advantages of feature extraction and pattern recognition, which automatically captures the hidden patterns in financial data, and is more helpful for quantitative investment.

CNN methods have a wide range of applications in the field of quantitative financial investment. Zhang et al. constructed a hybrid CNN model combining attentional memory and channel weighting to effectively capture intrinsic noise and coin associations for cryptocurrency price prediction [10]. Wu et al. developed a new ACNN model with historical stock price and derivative data, constructed as a sequential array input to a convolutional neural network, which significantly improved the accuracy of stock price movement prediction and verified its potential for practical application in Taiwan and U.S. stock markets [11]. Memiş et al. achieved optimal performance in both binary and multi-class sentiment analysis by characterizing word embeddings and constructing neural network models such as CNNs for sentiment classification based on Twitter financial texts [12]. Zolotareva proposes a trend recognition framework based on expert image annotation with three synergistic CNN sub-models to locate the trend start and end points and determine the direction for long-term trends in the stock market [13]. Abid and Saqlain established a deep assessment model based on CNN and edge computing for quantifying the multidimensional risk of China's international trade investment with an accuracy of 90.38% [14]. Chen et al. applied deep CNN to quantitative trading and designed a trading strategy constructed based on CSI 300 time-series data output with a Sharpe ratio of up to 2.204 [15]. Huang et al. proposed a two-stage portfolio optimization framework fusing a CNN-based stock trend prediction model with a global minimum variance model, which enhanced cumulative returns by about 35% and effectively improved key performance indicators such as the Sharpe ratio [16]. Maratkhan et al. integrated three deep learning prediction frameworks for CNNs and achieved up to 36.70% annual returns in the Dow Jones 30 test by constructing clustered images of technical indicators and optimizing stop-loss strategies, respectively [17]. Luo et al. innovatively applied Faster R-CNN to efficiently recognize and predict multi-featured image patterns such as W-bottom and entanglement pivot, which significantly improves the accuracy of pattern recognition for parsing K-chart patterns, thus realizing quantitative financial investment analysis [18].

This paper adopts the research process of "theoretical analysis - empirical modeling - strategy validation" to study the method of deep learning technology for quantitative trading. The CNN-LSTM multi-factor stock picking strategy is proposed, which extracts the nonlinear relationship between multiple financial factors in convolutional neural network, and extracts the memory characteristics of financial time series in LSTM; in the testing stage of the investment strategy, historical performance test, K-fold cross-checking, and random re-sampling test are carried out. Finally, the model performance and stability are comprehensively analyzed.

2 Methodology and theory

2.1 Deep Learning Algorithms

2.1.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) use parameter sharing between neurons in different layers to greatly reduce the model parameter magnitude, which solves the curse of dimensionality as well as the problem of falling into a local optimum to some extent. In this paper, we argue that if multiple types of indicators of a certain moment in the stock market are organized into a feature map, CNNs can learn from this map and extract market representations for that period. As a result, CNNs can be used to construct a time-timing model, thus realizing the design of time-timing strategies.

The basic structure of a CNN consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. The CNN model is usually built from the input layer, multiple convolutional layers and pooling layers are stacked sequentially, then the fully connected layer is accessed, and finally the result is obtained through the output layer, and the structure of the convolutional layer is shown in Fig. 1. The convolutional layer utilizes the convolution kernel to carry out convolutional operation on the input feature map, and then outputs the newly obtained convolutional feature map, and passes this output as the input of the next layer into the next layer.

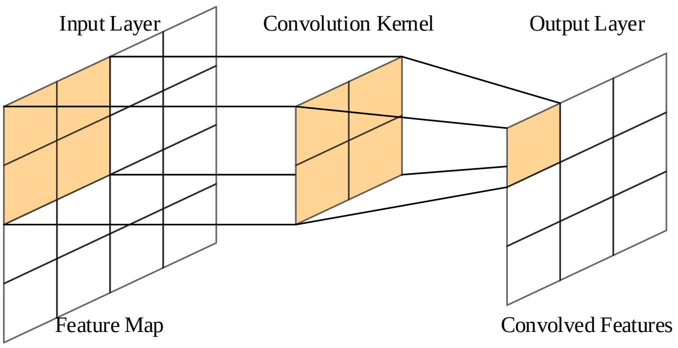

Figure 1: Convolutional Layer structure

The convolutional layer is calculated as:

$$u_j^l = \sum_{i \in M_j} x_{ij}^{l-1} * k_{ij}^l + b_j^l \tag{1}$$

where l represents the lth convolutional layer, in Fig. 1, the input layer/feature map is the $l-1$th layer, and the output layer/convolved features are the lth layer. j is the jth channel, i.e., the j th output term of the convolved feature map, and there are 9 output terms in the l th layer of Fig. 1, for a total number of 9 channels. M_j is a subset of channels of the partial feature map of channel j of the $l-1$ th convolutional layer corresponding to the l th convolutional layer, and M_j is $\{1,2,3,4\}$ in Fig. x_{ij}^{l-1} represents the i -th input of the feature map of channel j in the convolution layer at layer $l-1$, corresponding to the convolution layer at layer l . k_{ij}^{l} is the i th weight of the convolution kernel corresponding to the j th channel of the l th convolutional layer. b_j^l is the bias term of the j th channel of the l th convolutional layer. * is the convolution symbol.

$$x_j^l = f\left(u_j^l\right) \tag{2}$$

The function f is an activation function, and in this paper we use the common function, i.e:

$$f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The structure of the pooling layer is shown in Fig. 2, where the pooling layer samples the output of the previous layer as an output, Eq:

$$u_j^l = \beta_j^l \; pooling\left(x_{ij}^{l-1}\right) + b_j^l \tag{4}$$

where β_j^l is the weight of the l th pooling layer corresponding to channel j . $pooling(\;)$ function is the pooling function, and in this paper, the max-pooling method is used, i.e., the largest of the i inputs x_{ij}^{l-1} of the partial feature maps corresponding to channel j in layer $l-1$ is selected as the output of channel j in layer l . b_j^l is the bias term.

$$x_j^l = g\left(u_j^l\right) \tag{5}$$

The function g is also an activation function, and the sigmoid function is also used in this paper.

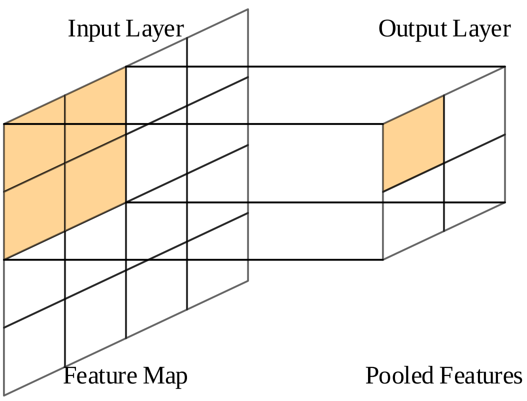

Figure 2: Structure diagram of the pooling layer

2.1.2 Long and short-term memory networks

The core structure of Long Short-Term Memory (LSTM) is a cellular state, but this state cannot individually determine what should be remembered and what should be forgotten. The storage and updating of the memory state is mainly realized by its original three gates (Forget Gate, Input Gate and Output Gate). The Forget Gate filters out unimportant information from the past, while the Input Gate recognizes emerging important information and stores it in the cell state.

The LSTM architecture is shown in Fig. 3, and the input at the moment of t is represented in the structure diagram X_t, the output of the LSTM cell at the moment of $t-1$ is represented in h_{t-1}, the memory of the cell at the moment of $t-1$ is represented in C_{t-1}, and the x represents the element-by-element multiplication. The small box with σ in the cell represents the sigmoid activation function, and the small box with tanh represents the tanh activation function.

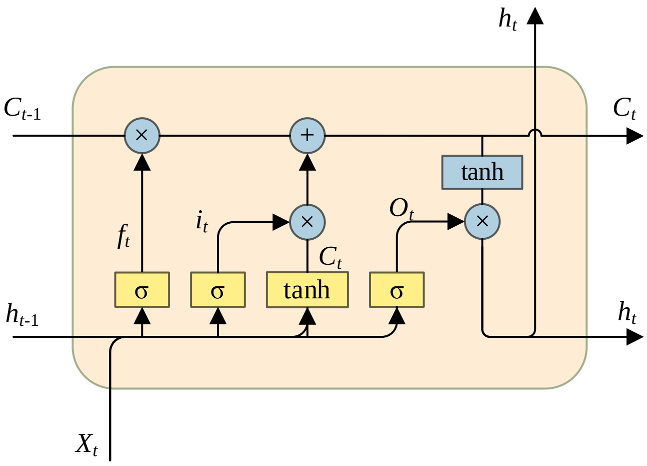

Figure 3: LSTM architecture

The first step in the execution step of the LSTM network is to determine what old information should need to be forgotten for the previous cell state. The forgetting gate f_t outputs a vector with each element belonging to $(0,1)$ through the inputs X_t and h_{t-1}, stating the extent of what needs to be kept in the cell state C_{t-1}. b_f and W_f are the bias terms and input weights of the forget gate f_t, respectively, representing the multiplication of matrices. The specific expression for f_t is:

$$f_t = \sigma\left(W_f \cdot \left(X_t, h_{t-1}\right) + b_f\right) \tag{6}$$

The second step is to determine what new information should be entered for the current cell state. First, X_t and h_{t-1} determine the change in information by σ in the input gate i_t. Then X_t and h_{t-1} get new candidate cell information $\tilde{C}_t$ via $\tanh$, which produce transformations on the cell's information update. b_i and W_i are the bias term and input weight of the input gate i_t, and b_c and W_c are the bias term and input weight of the candidate cell state $\tilde{C}_t$, respectively. The specific calculation method is as follows:

$$i_t = \sigma\left(W_i \cdot \left(X_t, h_{t-1}\right) + b_i\right) \quad (7)$$

$$\tilde{C}_t = \tanh\left(W_c \cdot \left(X_t, h_{t-1}\right) + b_c\right) \quad (8)$$

The third step is the update of the current cell state. The forgetting gate f_t and the previous cell state C_{t-1} are multiplied to determine the forgotten information. In the second step the current candidate cell state $\tilde{C}_t$ and the input gate i_t have been determined, and the multiplication determines the new candidate cell information that needs to be added. Based on the above calculations, it is possible to calculate the updated value of the cell state C_t at the moment t, denoted by * for dot product, as follows:

$$C_t = i_t * \tilde{C}_t + f_t * C_{t-1} \quad (9)$$

After the cell state update values are computed, the final step allows the computation of the value of the output gate O_t. b_o and W_o are the bias term and input weight of the output gate O_t, respectively, and the final output result is determined by O_t and C_t. The final output result is determined by and:

$$O_t = \sigma\left(W_o \cdot \left(X_t, h_{t-1}\right) + b_o\right) \quad (10)$$

$$h_t = O_t * \tanh\left(C_t\right) \quad (11)$$

With the above calculations, the LSTM will be able to effectively utilize the inputs to give it a long term memory function.

2.2 Theories of quantitative financial investment

Quantitative investing is an investment approach developed based on modern theories of finance, with the efficient market hypothesis being one of the key guiding principles in its design. Under the three-tiered efficient market theory, a weak efficient market is characterized by the price of a stock containing all the historical information thus making it impossible for the basic technical indicators to gain any more excess profits. The semi-strong efficient market hypothesis assumes that public information, including financial information, macroeconomic information, and industry research reports, has been reflected in stock prices to the extent that fundamental analysis cannot continue to generate excess returns, and the strong efficient market hypothesis assumes that insider information is also incapable of generating excess returns. The widespread existence of anomalies such as momentum effect, reversal effect, scale effect, value effect, etc. in the real market provides theoretical support for quantitative strategies, and these market failures create a possible space for systematic data mining and modeling to capture excess returns.

Portfolio theory utilizes Markowitz's mean-variance model to quantify risk into yield variance, and establishes a mathematical relationship between return and risk so that investors can construct an efficient frontier to achieve the optimal risk-return ratio, and the diversification effect brought about by the different correlations between assets can greatly reduce the risk level of the entire portfolio.Sharpe's development of the Capital Asset Pricing Model (CAPM) reduces risk by proposing the concept of a market portfolio. Reduces the difficulty of optimization by proposing the concept of a market portfolio, assuming that expected return is linearly related to systematic risk:

$$E(R_i) = R_f + \beta_i (E(R_m) - R_f) \tag{12}$$

In Ross's arbitrage pricing theory, asset returns are decomposed into risk premiums constituted by multiple factors, and a multi-factor quantitative investment system has been established on this basis; on the basis of the CAPM model, the Fama-French three-factor model, which contains market capitalization factor and book-to-market factor, has been proposed. After that, a five-factor model containing momentum factor and inversion factor is proposed successively, as well as a six-factor model containing market capitalization factor, book-to-market ratio factor, momentum factor and inversion factor, which makes the factor trading theory enriched and developed.

The practical use of quantitative investment strategies in the financial market has shown a diversified development trend. Statistical arbitrage utilizes the relationship of assets to historical statistics to make opposite transactions to profit when the spread deviates, momentum strategy verifies the medium-term momentum effect to buy the strong and sell the weak, and mean reversion strategy utilizes the short-term overreaction of prices to establish the reverse investment logic to perform better in commodity futures and foreign exchange. Factor investing strategies consciously expose themselves to a number of specific risk factors such as the value factor, the quality factor, the low volatility factor, the profitability factor and so on in order to obtain a factor risk premium. Machine learning introduces algorithms such as Random Forest, Support Vector Machines, and Neural Networks into the field of factor investing, which outperforms traditional linear methods in feature selection, pattern recognition, and nonlinear modeling. As a result, quantitative investing can handle more complex data structures and capture more subtle investment information. The complexity of market microstructure has a significant impact on strategy execution. Trading frictions such as bid-ask spreads, market shock costs, liquidity constraints, etc., can eat up theoretical returns, especially in the case of high-frequency trading strategies that require accurate estimation and control of trading costs. Data quality issues such as missing values, outliers, survival bias, forward-looking bias, etc., can lead to significant deviations in backtesting results. Different data providers, lag effects, and changes in company behavior affecting past data need to be taken into account in modeling.

Model risk is a growing concern as a unique type of risk in quantitative investing. Problems such as overfitting, parameter instability, and model failure can lead to a significant deterioration in out-of-sample performance, and techniques such as Black-Litterman modeling, Bayesian methods, and integrated learning are widely used for risk mitigation. Regulatory factors can also affect the application of quantitative strategies, such as short-selling restrictions, trading taxes, position limits, and disclosure requirements have changed the way the market operates and how investors behave, which requires quantitative investors to adjust their strategies accordingly to regulatory changes; meanwhile, the market structure is also changing and can affect the effectiveness of quantitative strategies, such as algorithmic trading, high-frequency trading, and passive The growth of investments will all lead to different price formation mechanisms and liquidity providers. As more and more arbitrage funds enter the market, some traditional market oddities will be eliminated, and quantitative investors will have to continuously optimize their own strategies to obtain excess returns, in this process, the relevant technical team ability to put forward higher requirements, only with stronger R & D capabilities and risk control capabilities in order to be able to achieve excellent performance in the industry competition.

3 Modeling and experimental design

Regulatory level of the fund investment operation process, the high degree of concern for the comprehensibility and deep neural network algorithms own "non-interpretability" characteristics also formed a relatively sharp conflict. Most of the literature does not provide sufficient explanation and interpretation of the investment decision-making behavior of DNNs, which will face certain legal challenges in the real investment operation. In addition, the out-of-sample generalization test is insufficient, and a large number of factors are only tested with data from a certain point in the past, with less testing of the long-term stability of the factors, which is prone to the problem of factor failure. The standardization of data preprocessing and variable construction needs to be strengthened, and scholars use different ways of cleaning the data set, constructing feature dimensions, and labeling methods, which leads to the difficulty of reproducing the experimental process and the difficulty of comparative analysis of experimental conclusions. Meanwhile, the modeling of transaction costs and market shocks is also too simple, and most of the work only considers the simple linear transaction costs, but does not take into account the impact of liquidity constraints, market shocks and other factors in the real trading process. To address the above points, the research direction of this paper focuses on improving the model's interpretability and generalization ability to out-of-sample, accurately modeling the transaction cost, etc., and establishing a portfolio model based on CNN-LSTM network, aiming to provide a more effective and feasible method for quantitative financial investment.

3.1 Data Acquisition and Preprocessing

3.1.1 Data acquisition

In the portfolio strategy established in this paper, the dataset required for model training, including historical quotes, turnover, technical indicators, macro factors and the corresponding data of fundamentals. The stock data used in this paper comes from Wind software and Flush financial system from 2010-01 to 2023-12, including the daily open-high-low-close price and volume and turnover information of CSI 300 constituent stocks. At the same time, in order to ensure data integrity, this paper has screened all the variables, removing the time of establishment less than 3 years, suspension of trading for a long time and may be delisted deal with the company to get a total of 280 stocks as the research object.

In this paper, TA-Lib in Python is used to calculate the technical indicators, including moving averages, relative strength index, Bollinger bands, exponential smoothing average, stochastic oscillator and more than thirty common technical indicators. And we use industry-recognized standard parameter settings to ensure that they are the same as the actual market. Macro-fundamental data comes from the National Bureau of Statistics and CEIC database, including GDP growth rate, CPI, PMI, money supply, interest rates and other important macro-economic variables, which are converted from monthly to daily using linear interpolation to match the time granularity of the stock data. Company fundamentals data are derived from annual and quarterly periodic reports of listed companies, involving core financial indicators such as operating income, net profit, gearing ratio, return on net assets, and earnings per share. After completing the standardization process, the official disclosure date of listed companies' financial reports is aligned as the available time point of the data to avoid forward-looking bias and future information leakage.

3.1.2 Data pre-processing

Data cleansing is carried out chronologically to verify the data situation and to resolve data problems. Different approaches are taken to fill in data for data gaps, such as using previous prices to fill in occasional gaps in prices, and leaving the volume of trades unchanged when there is a chance of a value of zero. Multiple consecutive missing days are filled using linear interpolation, or moving average. Outlier detection A variety of testing strategies are designed based on statistical theory, and the 3σ method is utilized to detect obvious misrecorded data. Then, the box plot method is used to detect abnormal price fluctuations, and different treatment strategies are used for the identified outliers according to their causes, including direct deletion of obvious erroneous data and reasonable extreme values, which are retained but given special attention in the subsequent analysis.

In the data preprocessing stage, the data correction is carried out for the stock price changes caused by ex-dividend, ex-rights, stock bonus, etc. of listed companies, on the basis of which the corrected price series is obtained, and a series of relevant characteristic factors are constructed from the aspects of price trend, price pattern, market atmosphere, and financial indexes, etc., respectively, in the feature extraction stage. Among them, the price trend refers to the change in the size of asset returns in different cycles, including 1-day, 5-day, 10-day, 20-day returns and the corresponding cumulative returns and volatility. The above characteristics can better capture the short and medium-term trend information of the stock price. The technical pattern is formed based on the price-volume combination of the formation of patterns, such as the relative position of price, the relative strength of volume, price-volume fit, etc., using certain mathematical processing methods to the complex graphs represented by the formation of the technical pattern characteristics expressed in simple numbers.

Data standardization processing using Z-score standardization method to ensure the comparability of different volume characteristics, standardization formula for:

$$X_{standardied} = \frac{X-\mu}{\sigma} \tag{13}$$

Here X denotes the original eigenvalue, μ is the eigenmean, σ is the eigenstandard deviation, and the standardization of various features can avoid the large fluctuation caused by the difference in the magnitude of different features when using the neural network for learning, which is conducive to the convergence of the model. Considering the strong time series nature of some macro indicators, a sliding window is used to standardize them to match their dynamic change characteristics, and the window size is set to one year, i.e., 252 working days. The Market Sentiment Factor measures the speculative psychology of the market based on a combination of technical indicators, such as the RSI which indicates overbought and oversold, the position of the Bollinger Band which indicates the relative position of the price, and the change in the signal line of the MACD which indicates a reversal trend.

The basic dataset is shown in Table 1, which includes price data, technical indicators data, macroeconomic data, fundamental data and news sentiment data of stocks, with a total sample size of 2,081,491 and restricts all the vacancies in it to less than 5%. And the quality characteristics of fundamentals are obtained based on financial indicators normalized and ranked according to their size, specifically profitability, growth, value and financial condition. The above characteristics give a quantitative measure of a stock's intrinsic value, and taking into account the characteristics of financial time series data, the samples are partitioned using time series cross-validation to prevent future leakage. In this case, the training set is the first ten years of data (2010-01 to 2020-12). The validation set is the eleventh year's data (2021) used for the tuning parameter, and the early termination mechanism. The test set is the data of the last two years (2022-01 to 2023-12) is used for prediction effect evaluation to ensure that the whole modeling process follows the time series, to prevent the occurrence of overfitting phenomenon caused by random slicing, and to improve the model stability.

Table 1: Statistics of basic information of the dataset

Data type	Sample size	Sample size	Time span	Missing rate (%)	Data source
Stock price data	958,400	6	2010-2023	0.12	Wind, Tonghuashun
Technical indicator data	958,400	32	2010-2023	1.85	TA-Lib calculation
Macroeconomic data	3,422	15	2010-2023	2.31	National Bureau of Statistics
Fundamental data	4,480	28	2010-2023	3.67	Report of Listed Companies
News sentiment data	156,789	5	2015-2023	0.89	Financial media

3.2 Model Construction and Training

3.2.1 Modeling

The quantitative investment model based on CNN should fully consider the characteristics of financial time-series data, for this reason, the paper adopts a one-dimensional convolutional layer + LSTM hybrid network in order to adapt to the multi-dimensional financial characteristics data input. The hybrid neural network structure in the paper mainly takes into account the local perception characteristics of the convolutional layer on the data, with a view to effectively extracting the spatial correlation that exists between different financial markets, and fully combines the LSTM's ability to deal with long and short-term memory in the ability to learn the sequence information. The multidimensional feature matrix obtained after sufficient preprocessing is used as input, and the convolution kernel in layer 1, a one-dimensional convolution kernel of size (3, 64) is used to carry out convolution operation on the input feature matrix, and the ReLU activation function is utilized to increase the nonlinear relationship between neurons. Finally, a Batch Normalization layer is added to accelerate the training process and improve the network robustness. The later convolutional layers use 128 convolutional kernels of size 3 to continue to obtain more abstract feature representations, and a Maximum Pooling layer is set up after each convolutional layer for downsampling, which can achieve the purpose of decreasing the number of parameters and increasing the translation invariance of the feature representations.

Among them, the LSTM layer has 256 hidden units to learn the complex long- and short-term dependencies of time series. The long term amnesia of ordinary RNN can be solved by using LSTM, as well as feature extraction for the long term memory of the market. Avoiding the occurrence of overfitting is a major difficulty in this model, in the paper, a Dropout layer is added after the LSTM layer, and the Dropout ratio is set to 0.3, which can effectively improve the robustness of the model under the guarantee of a higher degree of model fitting. The hidden layer is 128 nodes, and the RELU function is selected to fully extract the information, and the output layer is 3 nodes representing the actions of buying, not operating, and selling, respectively, and the SMH activation function is selected as the output to satisfy the probabilistic nature of the output.

The loss function adopts the cross-entropy loss to accurately optimize the performance performance of the multiclassification prediction task:

$$L = -\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{14}$$

where N denotes the number of samples, y_i is the true label, and $\hat{y}_i$ is the model prediction probability, the loss function is able to effectively penalize misclassification and guide the model to learn the correct decision boundary.

3.2.2 Model training

The model is optimized using Adam with a learning rate of 0.001 and a learning rate decay strategy, and the parameters of the model are further fine-tuned at the late stage of training. batch_size is set to 64 to ensure the balance between training speed and memory, the maximum number of epochs is 200, and an early stopping strategy is set to avoid the model from overfitting. If there is no change in the validation set loss in 10 epochs, the training is terminated early. To improve the robustness of the model, we used a data augmentation strategy to randomly apply different transformations to each sample during the training process, such as adding Gaussian noise and sliding the window in the time dimension, so as to increase the richness of the samples. Weight initialization uses Xavier initialization to ensure reasonable probability of the distribution of activation function values at each layer, and gradient trimming is used to avoid gradient explosion and helps to smooth training over the entire network. Hyperparameter optimization utilizes a grid search combined with a Bayesian optimization algorithm for the best combination of hyperparameters, whose main objectives include the number of filters, the number of hidden units, the learning rate, and the batch size, among others. In this study, we consider a large number of possible candidate setups and evaluate each candidate setup using 5-fold time-series cross-validation to ensure that all candidate setups are adequately evaluated, and the optimal candidate setup is found in the validation set as the final model setup.

In addition, in order to better adapt to the time-varying characteristics of financial markets, this paper implements online learning so that the model can be incrementally updated with newly arrived market information without having to re-train the entire model, and online learning empowers the model to be more time-adaptive to capture changes in the market structure and to adjust its forecasting behavior accordingly. The change in loss during model training is shown in Fig. 4, which shows that after 100 rounds, the training loss is 0.402%, and the loss for validating the trained model against the test set is 0.506%. Compared with the traditional linear model, the results are significantly improved, which shows the feasibility as well as the power of deep learning methods in the field of quantitative investment.

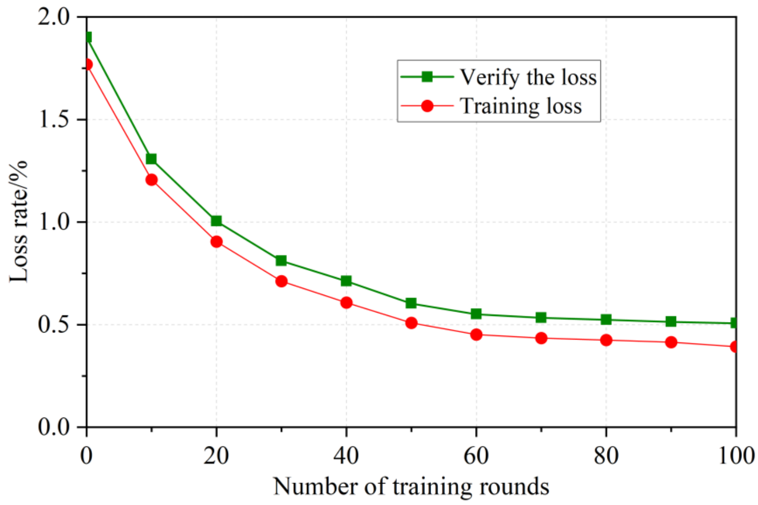

Figure 4: Loss changes during the model training process

3.3 Experimental design and model evaluation

The empirical tests designed in this paper are based on time series backtesting to evaluate the effectiveness of the quantization strategy for CNN-LSTM hybrid networks, and to simulate the evolution of the model over time by dividing the sample interval (2010-2020) into a training set (252 days) and a test set (63 days) according to a sliding window. Python 3.8, TensorFlow 2.6, and Pandas 1.3 tools were used in the software platform to complete the entire data science analysis. At the same time, the data are preprocessed in strict accordance with the way of taking values forward, no future data are used in the process of model construction, and the data are selected in the corresponding technical factors, financial data, and macro factors in strict accordance with the real time point of data generation, so as to ensure the validity of the backtesting results.

Considering the classification performance of machine learning and the risk-return characteristics of financial investment, the evaluation indexes constructed include the ratio that can represent the correct prediction results of the model for the three kinds of decisions of buy, hold and sell, i.e., the classification accuracy rate, and the higher the rate indicates that the basic prediction performance of the model is better, but it can't reflect the problem that different prediction errors bring the investors' losses of different sizes. The accuracy rate is the proportion of the data judged to be correct in the positive category; the accuracy rate is the proportion of the data judged to be correct from all the positive categories.

The F1 value expresses the composite metric in terms of the reconciled average of the two; a detailed analysis of the confusion matrix can guide the next step in how to improve the model. The use of logarithmic returns facilitates the statistical treatment of portfolio returns; annualized returns discounted to one year directly tell investors an approximate investment benefit. The Sharpe ratio is an important measure of the magnitude of risk-adjusted returns and is calculated as:

$$S = \frac{R_p - R_f}{\sigma_p} \tag{15}$$

The information ratio is the ratio of the expected value of excess return to its standard deviation, which is used to reflect the stability of excess return relative to the benchmark, and is suitable for evaluating the value creation ability of actively managed products. Maximum retracement represents the maximum potential loss of the risk faced by the strategy, and refers to the maximum retracement amplitude from the historical high point to the lowest point that occurs afterward, which is an important reference value for risk averse investors, volatility reflects the price fluctuation of the portfolio in terms of the standard deviation of the rate of return, the win rate statistic reflects the ratio of the number of correctly predicted trades to the total number of trades, and the changeover rate reflects the impact of the strategy's trading frequency on the cost control impact.

Statistical significance tests were performed using t-tests and Wilcoxon signed rank tests to test the significance of model performance improvement and p-values were used to make judgments about the significance of performance differences. Sensitivity analyses test strategy stability by examining changes in factors such as training window size, rebalancing period, and transaction expense ratio. The out-of-sample test includes changes in the market during special periods such as the stock market crash event in 2015, the U.S.-China trade war in 2018, and during the New Crown epidemic outbreak in 2020. The applicability of the investment model under different market conditions is tested, and the comprehensive operation of the investment model is visualized to investors in the form of net value charts, frequency distribution charts of returns, and risk-return matrix charts.

4 Empirical studies

4.1 Analysis of experimental data and results

The results of the empirical analysis are shown in Table 2, which shows that the hybrid CNN-LSTM model proposed in this paper achieves good results on the CSI 300 stock index, and a large number of trading samples are obtained after out-of-sample testing of the data of the selected 280 stocks for the 2-year period from 2022-01 to 2023-12, i.e., a total of 672,000 valid samples are obtained, where there are 20 days of each trading day of the historical data along with 81 variables. In terms of accuracy, the hybrid CNN-LSTM network reaches 68.5%, which is 8.7% higher than the 59.8% of RF and 12.2% higher than the 56.3% of SVM. It reaches 71.2% on the precision metric, i.e., 7 out of 10 stocks predicted as buys are correct. Its recall is 65.8%, indicating that about 65% of the correct stocks are successfully selected for investment.F1 = 68.4%, indicating that the accuracy and recall balance value is 68.4%.

Table 2: Experimental result data

Model type	Performance indicators				
	Accuracy rate /%	Precision rate /%	Recall rate /%	F1 /%	Training time /h
CNN	63.7	66.4	61.2	63.7	8.7
LSTM	65.2	67.9	62.5	65.1	15.6
Random Forest	59.8	62.1	57.4	59.6	2.1
Support Vector machine	56.3	58.7	53.9	56.2	4.5
Logistic regression	52.1	54.3	49.8	51.9	0.8
CNN-LSTM	**68.5**	**71.2**	**65.8**	**68.4**	**12.3**
Investment strategy	Financial indicators				
	Annualized return /%	Sharpe ratio	Informatio n ratio	Maximum drawdown /%	Winning rate /%
Random Forest	12.4	1.18	0.72	-22.1	54.7
Buy-hold strategy	8.9	0.73	0.41	-35.4	48.2
Momentum strategy	6.2	0.58	0.28	-28.9	51.6
CNN-LSTM	**18.7**	**1.42**	**0.89**	**-15.2**	**58.3**

During the backtesting period, the CNN-LSTM strategy achieved an annualized return of 18.7%, which is 9.8 percentage points higher than the annualized return of 8.9% of the CSI 300 index during the same period. The Sharpe ratio is 1.42, indicating that the strategy achieves a high level of return while taking a certain amount of risk. The Information Ratio is 0.89, gaining a more stable excess return while comparing to the CSI 300 Index. The maximum retracement was -15.2%, indicating that the strategy also took a certain level of risk in its investments, but still within acceptable limits. The strategy's 58.3% win rate indicates that more than half of the trades have a chance of being profitable, and the monthly returns are slightly right-skewed (0.34), which approximates a standard normal distribution (with a kurtosis of 2.87), with an overall low level of volatility. Looking at the rolling window, the strategy's performance changes considerably over time, averaging about 2.3% per month in excess returns in the first six months of the year, declining in the second six months due to the market, averaging about 1.1% per month, and picking up to about 1.8% per month in 2023 as the model's fit improves. And in each industry allocation shows that the strategy for technology, pharmaceuticals, consumer and other industries have significant excess returns, for the traditional cyclical industries slightly less effective, and this industry characteristics also reflects the ability of deep learning algorithms to capture the complexity of non-linear information; In addition, we also conducted a transaction cost backtesting, the results show that when the transaction cost is 0.0005, the strategy can still be obtain more than about 15% after-tax return. The cost saving effect is good.

4.2 Model Optimization and Strategy Improvement

Further analysis of the previous data reveals that the results obtained using the CNN-LSTM model are highly unstable, with an accuracy of only 61.3% in a falling market, while the accuracy in a rising market is as high as 74.2%, indicating that the model is not sufficiently sensitive to changes in the market. Aiming at the above problems, a multi-scale model reconstruction method based on the multi-head self-attention module is proposed, which utilizes the multi-head self-attention to weight each channel and adds residual blocks between the CNN and LSTM layers to improve the learning efficiency of the deep neural network. The results of the model before and after the improvement are shown in Table 3.After further integration, the total prediction accuracy during the test sample period is 75.3%, which is 6.8 percentage points higher than the original model, the annualized return reaches 26.8%, the Sharpe index improves to 1.84, and the information ratio rises to 1.28, and the maximal retraction decreases to a relatively tolerable range of-10.9%, and the overall risk-reward characteristics are all substantially improved, indicating the good applicability of our proposed method. This shows that our proposed method has good applicability and lays a good mathematical foundation and technical platform for the application of deep learning algorithms in the field of quantitative trading.

Table 3: Performance Comparison Before and After Model Optimization

Performance indicators	Original	Structural optimization	Hyperparameter optimization	Comprehensive optimization
Overall accuracy rate (%)	68.5	71.2	72.8	75.3
Buying signal accuracy (%)	73.6	76.4	78.1	80.7
Signal holding accuracy (%)	69.2	72.1	73.5	76.2
Accuracy rate of sell signal (%)	62.8	65.3	67.9	71.4
Bull market accuracy rate (%)	74.2	77.1	78.6	81.3
Accuracy rate of volatile markets (%)	65.8	68.9	70.4	73.1
Bear market accuracy rate (%)	61.3	64.7	66.8	69.5
Annualized rate of return (%)	18.7	21.4	23.2	26.8
Sharpe ratio	1.42	1.58	1.67	1.84
Information ratio	0.89	1.02	1.14	1.28
Maximum drawdown (%)	-15.2	-13.8	-12.4	-10.9
Training time (h)	12.3	18.7	15.6	22.4

Among them, hyperparameter tuning uses a Bayesian search algorithm based on the Gaussian process regression model to perform an automatic search for hyperparameter combinations including the learning rate, batch size, and number of layers, and greatly improves the search speed compared to the grid search. For the data expansion part, the real perturbations obtained by the fast gradient sign method are utilized against samples to improve the robust performance of the model and increase the difficulty of the learning task while ensuring that the original labeling information is not lost. In this paper, the dynamic weight allocation mechanism based on the performance of the validation set is utilized to fuse the prediction outputs of the sub-models, which results in the reduction of the prediction variance of the integrated model compared to a single model, and significantly improves the robustness performance of the investment strategy. After 100 experiments, the optimal combination of hyper-parameters is selected as Reducing the number of convolutional kernels in the convolutional layer (from 64~128 to 96~192), which can extract the characterization information under different time spans; Increasing the number of layers of LSTM (from 256 to 384), which can enhance the learning ability of the time-series signals; and Adopting 8 Attention Heads, which can improve the model accuracy while guaranteeing a certain training speed. The use of 8 Attention Heads ensures a certain training speed and improves the accuracy of the model. The cos-decaying learning rate is used to prevent the model from converging too early, and the batch_size is set to 96 to achieve a balance between the memory and the gradient variance, and a variety of regularization methods are used in the model, such as label smoothing, mixup training, random depth, etc., and label smoothing is used to increase the robustness to the correct category. To increase the robustness to the correct category, a noise of 0.1 is added to the correct category. Mixing, on the other hand, is a weighted average of the current sample using the feature vectors and labels of other samples to form a new sample.

Model combination is the result of training multiple sub-models with different stochastic seeds and data augmentation methods to improve the stability of the prediction results; dynamic training enables the model to be updated continuously with market changes, where it is set to be updated once a month to balance model robustness and operational efficiency, and is found to be effective in controlling the model's volatility level in out-of-sample periods. Enhance the functions of the risk control module, such as dynamic stop-loss, position management, correlation monitoring, etc. Among them, dynamic stop-loss dynamically adjusts the stop-loss position according to the current volatility of the market, so as to strictly control the risk in the case of severe volatility and avoid large retracements, and appropriately relax the stop-loss point in the calm period to reduce the cost brought by frequent trading.

5 Conclusion

In this paper, a CNN model is used to extract the deep spatial information features among the dimensions of the input variables, and combined with LSTM to extract the temporal information features of the dimensions of the financial market to build a hybrid neural network model (CNN+LSTM) for stock investment prediction. The out-of-sample validation experiments on the market capitalization-weighted index of CSI 300 constituents show that the hybrid model achieves a correct rate of 75.3% in all the prediction results, which is higher than the other two models RF (59.8%) and SVM (56.3%). The results are statistically significant ($p<0.001$). In terms of the importance of features, technical indicators and price momentum are the most important factors. Meanwhile, the dynamic weighting of different features is achieved using multi-attention, and the prediction performance in bull, shock, and bear markets is relatively stable (81.3%, 73.1%, and 69.5% correct). This paper demonstrates that deep learning's powerful automatic feature extraction and nonlinear fit modeling capabilities provide a fully automated system solution for quantitative investing, which, combined with online learning as well as a risk control module, is able to industrially ground the strategy and evolve with the environment.

Funding

University-level scientific research project of Wuhu Vocational Technical University: "The Impact of National Industry-Education Integration Policies on Enterprise Innovation: A Case

Study of Anhui Province", Project Number: wzyrc202410

Key Project of Humanities and Social Sciences Research of University-level Scientific Research Project of Wuhu Vocational Technical University: "Construction and Empirical Test of the Valuation Model for Intangible Cultural Heritage IP Licensing Based on the Modified Income Approach: A Case Study of Wuhu Zhang Family Traditional Women's Handicraft Skills", Project Number: wzdrwzd202601.

About the Author

Wangqi Shen, male, from Anqing, Anhui Province, graduated from Nanjing University of Technology with a bachelor's degree in Bioengineering and a master's degree in accounting from Shanghai University. He completed the doctoral program in management from Laihim University. Now he is a lecturer at Wuhu Vocational and Technical College, and his research interests include enterprise ESG practice, financial management, corporate governance and enterprise capital operation. His research specialties include mathematical linear regression analysis and the construction of mathematical model system.

Wenxiao Zhuo, female, is from Suzhou, Anhui Province. She graduated from Anhui Science and Technology University with a bachelor's degree and obtained a Doctor of Veterinary Medicine degree from Huazhong Agricultural University in 2023. Currently, she works at the School of Food and Bioengineering of Wuhu Vocational and Technical College, focusing on preventive veterinary medicine and bioengineering.

References

[1] Fang, Y., Chen, J., & Xue, Z. (2019). Research on quantitative investment strategies based on deep learning. Algorithms, 12(2), 35.

[2] Yao, J., Li, Z., Cui, T., & Xi, H. (2022). Quantitative Investment Trading Model Based on Model Recognition Strategy with Deep Learning Method. Wireless Communications and Mobile Computing, 2022(1), 8856215.

[3] Li, W., Hu, C., & Luo, Y. (2023). A deep learning approach with extensive sentiment analysis for quantitative investment. Electronics, 12(18), 3960.

[4] Li, X., & Hong, R. (2025, March). A-Share Quantitative Investment Strategy for Multiple Objectives Driven by Machine Learning and Deep Learning. In Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence (pp. 945-950).

[5] Davis, C. (2025). The elasticity of quantitative investment. The Review of Financial Studies, 38(10), 2845-2886.

[6] Sareewiwatthana, P., & Janin, P. (2017). Tests of quantitative investing strategies of famous investors: Case of Thailand. Investment management and financial innovations, (14,№ 3 (contin. 1)), 218-226.

[7] Wu, W. (2023). Risk Management of Enterprise Quantitative Investment Strategies through Data Modeling. Journal of Engineering, Project & Production Management, 13(1).

[8] Khodayari Gharanchaei, M., Panda, P. P., & Chen, X. (2024). Quantitative Investment Diversification Strategies via Various Risk Models. Journal of Advancements in Applied Business Research, 13(1), 41-48.

[9] Cong, S., & Zhou, Y. (2023). A review of convolutional neural network architectures and their optimizations. Artificial Intelligence Review, 56(3), 1905-1969.

[10] Zhang, Z., Dai, H. N., Zhou, J., Mondal, S. K., García, M. M., & Wang, H. (2021). Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels. Expert Systems with Applications, 183, 115378.

[11] Wu, J. M. T., Li, Z., Srivastava, G., Tasi, M. H., & Lin, J. C. W. (2021). A graph-based convolutional neural network stock price prediction with leading indicators. Software: Practice and Experience, 51(3), 628-644.

[12] Memiş, E., Akarkamçı, H., Yeniad, M., Rahebi, J., & Lopez-Guede, J. M. (2024). Comparative study for sentiment analysis of financial tweets with deep learning methods. Applied Sciences, 14(2), 588.

[13] Zolotareva, E. (2021, June). Applying convolutional neural networks for stock market trends identification. In International Conference on Artificial Intelligence and Soft Computing (pp. 269-282). Cham: Springer International Publishing.

[14] Abid, M., & Saqlain, M. (2023). Utilizing edge cloud computing and deep learning for enhanced risk assessment in China’s international trade and investment. International journal of knowledge and innovation studies, 1(1), 1-9.

[15] Chen, H., Liang, D., & Zhao, L. L. (2019, April). A quantitative trading method using deep convolution neural network. In IOP Conference Series: Materials Science and Engineering (Vol. 490, No. 4, p. 042018). IOP Publishing.

[16] Huang, S., Cao, L., Sun, R., Ma, T., & Liu, S. (2024). Enhancing Portfolio Optimization: A Two-Stage Approach with Deep Learning and Portfolio Optimization. Mathematics, 12(21), 3376.

[17] Maratkhan, A., Ilyassov, I., Aitzhanov, M., Demirci, M. F., & Ozbayoglu, A. M. (2021). Deep learning-based investment strategy: technical indicator clustering and residual blocks. Soft Computing, 25(7), 5151-5161.

[18] Luo, Y., Zhang, Z., Jiang, R., & Liu, Y. (2024, July). Deep Learning Based K-Line Chart Recognition for Financial Quantitative Investment Analysis. In International Symposium on Neural Networks (pp. 198-210). Singapore: Springer Nature Singapore.