Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 18, No. 1, 2026

Stock Index Forecasting and Automated Trading System Based on LSTM-TCN Model

Yaoxing Ye * and Lei Xuan

Guangdong Country Garden School, Foshan, Guangdong, China

*** Corresponding author:** Yaoxing Ye (Email: Princessyyx@outlook.com)

Abstract: Currently, traditional models, like ARIMA, can hardly capture long-term dependencies and local volatility patterns of stock index time series simultaneously. Under this circumstance, this article constructs an LSTM+TCN automatic trading system for Hang Seng Index, NASDAQ Composite Index and Nikkei 225 Index. This system uses processed technical indicators, which can be obtained from the raw data directly, as feature engineering. Then, it predicts the direction and fluctuation magnitude of the first and the third days separately, and generates trading signals through a consensus mecha-nism among three branches. With a trailing stop strategy, the system shows a great trading performance in the back-test, even though the results of the prediction layer are not really ideal.

Keywords: Quantitative Trading; Long Short-Term Memory (LSTM); Time Series Convolutional Network (TCN); Signal Generation; Trailing Stop.

1. Introduction

With the development of quantitative investment, using machine learning to predict financial time series and generate trading signals has become an impor-tant research direction. Undoubtedly, machine learning has certain advantages in capturing nonlinear relationships and temporal dependencies [1], such as in the financial markets.

However, a single model is often difficult to maintain stable performance in different market states. LSTM excels at modeling long-term dependencies [2], while TCN excels at extracting local and multi-scale temporal structures[3]. Therefore, the combination of the two has strong complementary potential.

In the stock index market, the issue is more significant. The stock index is not only influenced by price history, but also by various factors such as macro environment, market sentiment, and geopolitics. This high dimensionality makes it difficult for a single neural network model to have a stable performance.

Additionally, the prediction accuracy of either traditional prediction relying solely on fundamentals or a machine learning model is not necessarily equal to trading profitability. Based on this, this article is not only going to simply construct a single model for prediction, but also going to design a comprehensive fusion model seeking practical trading results.

Finally, this article will conduct a back-testing. In specific, the results of this system will compare with the baseline of always bullish and the return rate of other mainstream models to verify the validity of the system with various statistical indicators.

2. Related Work and Motivation

In terms of financial forecasting, Hu, Zhao, and Khushi systematically com-piled the deep learning methods in foreign exchange and stock price forecasting, showing that there are diverse machine learning models for market prediction and plenty of methods for performance evaluation. The paper also points out that the accuracy and return rate vary with the use of different models, such as CNN, NLP, and Reinforcement Learning[4, 5].

In terms of model structure, the CNN-LSTM stock time series prediction frame-work proposed by Aadhitya et al. demonstrates the feasibility of combining two neural networks into a fusion model[6]. Although this work focuses on individual stocks, the idea of complementary models has direct inspiration for this article.

Additionally, a few recent papers further extends this idea to the TCN-LSTM combination. Using TCN instead of CNN to enhance multi-scale time series mod-eling capabilities is widely considered as one of the most important abilities in the field of high-dimensional market prediction. Dai et al. used TCN to predict the change of ultra-high frequency financial price, and found that its description of dynamic process was better than GARCH model and LSTM framework[7]. The Modern TCN proposed by Luo and Wang directly optimized the traditional TCN, which shows that the pure convolution structure can still maintain efficiency and prediction performance in time series analysis[8]. However, due to the limitations of the hardware, this article uses the simple original TCN as a branch of the model.

3. Model Design

The overall process of the complete system for stock index prediction and au-tomatic trading can be summarized into four stages: data preparation (data layer), feature engineering (feature layer), model prediction and fusion decision-making (prediction layer), automatic trading and back-testing evaluation (execution layer). First, in the data layer, the project uses daily frequency data from the Hang Seng Index (HSI), Nasdaq Composite Index (IXIC), and Nikkei 225 Index (N225). Each record retains elements including date, opening price, highest price, lowest price, closing price, trading volume, and Market Panic Index (VIX). The evalua-tion is conducted from January 1, 2025 to February 28, 2026. The corresponding number of forecast samples is about 284, 289, and 280, respectively. We use a uni-fied format to ensure the data interface can be used for subsequent model training and back-testing without any adaptive procedures.

Secondly, in the feature layer, the system constructed 9 core input features based on the original market data, including logarithmic return rate, gap return rate, RSI14, RSI change, normalized ATR, EMA difference ratio, closing price relative to EMA10 deviation ratio, high-low volatility interval ratio, and logarith-mic feature of trade volume. The reason for this article to select them for feature engineering is that trend, momentum, and volatility, three significant features in terms of financial market, can be clearly demonstrated by these technical indica-tors.

Furthermore, at the prediction layer, the project not only predicts the direction of price change of tomorrow, but also provides both the direction and magnitude of the future 1st and 3rd days relative to the current benchmark closing price. In this way, the model can provide short-term and medium-term price changes separately, which play a pivotal role in the following trading signal generation with the consensus mechanism

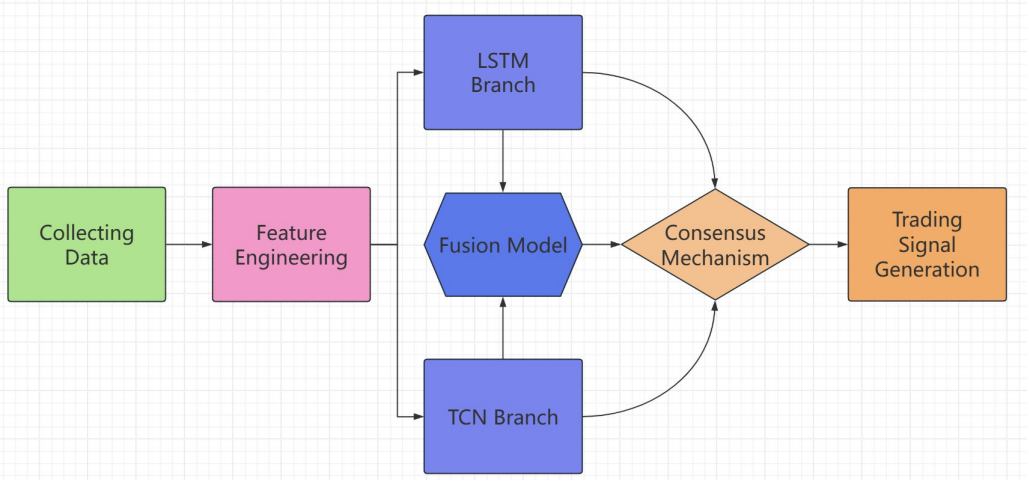


Figure 1. Flow Chart

Finally, at the execution layer, the project generates trading signals by the con-sensus mechanism and carries out back-testing with a trailing stop strategy as the core execution mechanism. In terms of experimental indicators, the prediction re-sult analysis mainly focuses on the direction accuracy, magnitude percentage er-ror, and MAPE. The trading result analysis focuses on the triggered count, yield, winning rate, cumulative income, and annualized income. Then, we will compare the result with the strong baseline, always bullish, to check the effectiveness and contribution of the model. This design also conforms to the complete framework view emphasized in financial machine learning: the model, label, implementation, and evaluation must be put in the same strategic chain for unified inspection[10]. Therefore, this article evaluates the prediction layer and the transaction layer, re-spectively. The entire flow chart is shown above in Figure 1

4. Methodology

4.1. Problem Definition

This article decomposes the problem of automatic stock index trading into two levels: price change forecasting and trading execution. In the prediction layer, for each trading day t, use a historical window of length L to construct an input sequence.

$$\mathbf{X}_t = [\mathbf{x}_{t-L+1}, \mathbf{x}_{t-L+2}, \ldots, \mathbf{x}_t] \in \mathbb{R}^{L\times d}$$

Among them, $\mathbf{x}_t$ is the eigenvector of the th day, and in this article d =9 is taken, including indicators such as return, volatility, momentum and trading vol-ume. Based on the closing price C_t, define the relative returns for the 1st and 3rd day in the future as

$$r_t^{(h)} = \frac{C_{t+h} - C_t}{C_t}, \qquad h \in \{1, 3\}.$$

This way of directly defining multiple future horizon revenue targets based on the current time point is consistent with the modeling idea of direct multi-horizon forecasting, and is more suitable for directly mapping the forecast results to the trading decisions in the future[9].

In order to reflect both market direction and volatility intensity, this article constructs direction labels and magnitude labels:

$$y_t^{(h)} = \mathbb{I}(r_t^{(h)} > 0), \qquad m_t^{(h)} = |r_t^{(h)}|.$$

Therefore, the model outputs a set of probabilities for sign and magnitude estimates on each horizon:

$$(\hat{p}_t^{(h)}, \hat{m}_t^{(h)}), \qquad h \in \{1, 3\},$$

By using

$$\hat{r}_t^{(h)} = \mathrm{sgn}(\hat{p}_t^{(h)} - \tau_h)\hat{a}_t^{(h)},$$

restore predicted returns in the market sense, where is the calibrated classifica-tion threshold. Compared to the approach of only predicting the direction of rise and fall, the setting above is more suitable for trading research and trading practice, as splitting the future price change into two dimensions of "direction" and "inten-sity" can retain more trading information than a single rise and fall label[11].

4.2. LSTM and TCN Forecasting Branches

The LSTM branch is mainly used to extract dependency relationships over a longer time range. Given input sequence $\mathbf{X}_t$ The encoding process of can be sim-plified as

$$\mathbf{H}_t^L = \mathrm{LSTM}(\mathbf{X}_t).$$

where, $\mathbf{H}_t^L$ represents the temporal encoding result at the time point in the LSTM branch. LSTM is good at handling long-term dependence through gated mem-ory mechanism. Its basic idea comes from the classical structure proposed by Hochreiter and schmidhuber. Then, the model uses the last hidden state or atten-tion pooling results as a sequence

representation[2], and further outputs logits and magnitude prediction in the direction:

$$\hat{\mathbf{p}}_t^{L} = \sigma(\phi_s(\mathbf{H}_t^{L})), \qquad \hat{\mathbf{m}}_t^{L} = \phi_m(\mathbf{H}_t^{L}).$$

where $\mathbf{p}^{L}$ represent directional probability vector generated by the LSTM branch, $\mathbf{m}_t^{L}$ represents the corresponding magnitude prediction vector, $\phi_s(\cdot)$ and $\phi_m(\cdot)$ In-dicates the direction head and magnitude head respectively, $\sigma(\cdot)$ is Sigmoid map-ping, used to compress the direction output to the [0,1] interval.

The TCN branch is mainly used to capture local fluctuations and multi-scale temporal structure. The core idea is to expand the receptive field through expan-sion convolution, and the expansion rate of the ℓ layer is set as

$$d_\ell = 2^{\ell-1}, \qquad \ell = 1, 2, \ldots, K.$$

Where d_ℓ represents the expansion rate of the convolution block at layer ℓ and k represents the total number of layers of TCN. According to the discussion of the sequential convolution network by Bai et al. [3], the extended convolution structure enables the model to cover a longer time range while controlling the size of the parameters. Therefore, the timing code of TCN branch can be written as:

$$\mathbf{H}_t^{T} = \mathrm{TCN}(\mathbf{X}_t).$$

Then, through further derivation, we get:

$$\hat{\mathbf{p}}_t^{T} = \sigma(\psi_s(\mathbf{H}_t^{T})), \qquad \hat{\mathbf{m}}_t^{T} = \psi_m(\mathbf{H}_t^{T}),$$

where, $\mathbf{H}_t^{T}$ represents the temporal encoding result at the time point in the TCN branch. $\mathbf{p}^{L}$ represent directional probability vector, $\hat{\mathbf{m}}^{L}$ represents the correspond-ing magnitude prediction vector, $\psi_s(\cdot)$ and $\psi_m(\cdot)$ show corresponding output mapping. Thus, LSTM put more emphasis on long-term dependence on infor-mation, and TCN focus more on local mode and multi-scale structure, making them complementary in the stock index prediction[2, 3].

4.3. Multi-Task Learning Objective

In this article, the direction prediction is regarded as a binary classification prob-lem, the magnitude prediction is regarded as a regression problem, and the joint loss is used for training. The total objective function is written as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\mathrm{dir}} + \lambda_2 \mathcal{L}_{\mathrm{mag}},$$

where $\mathcal{L}$ represent total loss function. λ_1 and λ_2 control the importance weight of direction task and magnitude task, respectively.

This method of dividing the prediction task into two dimensions of "direc-tion+intensity" aligns better with the idea of emphasizing the consistency of label design and implementation goals in financial machine learning [10]. Specifically, the direction part uses weighted binary cross entropy:

$$\mathcal{L}_{\mathrm{dir}} = \sum_{h \in \{1,3\}} \omega_h \, \mathrm{BCE}(\hat{p}_t^{(h)}, y_t^{(h)})$$

Considering that the stock index market has both large trend fluctuations and a large number of small daily fluctuations, this model trains with both the absolute deviation and relative deviation between the predicted value and the real value in this part, which can be expressed as:

$$\mathcal{L}_{\mathrm{mag}} = \sum_{h \in \{1,3\}} \omega_h \left[\alpha \left|\hat{m}_t^{(h)} - m_t^{(h)}\right| + (1-\alpha)\frac{\left|\hat{m}_t^{(h)} - m_t^{(h)}\right|}{\max(m_t^{(h)}, \varepsilon)}\right], \qquad a \in [0,1]$$

where α is the weight for the error functions, ε is a stabilization constant, which is used to avoid error distortion caused by too small denominator when the real fluctuation is very small.

The consideration of such a design is: if only absolute deviation is considered, the model may ignore small and medium fluctuations; If we only look at the rel-ative deviation, it is easy to be over-sensitive to the minimum fluctuation. The combination of the two is more conducive to the stability judgment of the model in different fluctuation environments [4, 10].

4.4. Fusion Model

Let $h \in \{1,3\}$ represent the prediction horizon, superscript L, T, F represent LSTM, TCN and fusion model respectively.

This model calculates the probability of sign of the fusion model with the for-mula:

$$\hat{p}_t^{F}{}_{,h} = w_d \hat{p}_t^{L}{}_{,h} + (1 - w_d)\hat{p}_t^{T}{}_{,h}, \qquad w_h \in [0,1],$$

Where W_d represents the weight of LSTM on h th horizon in the fusion model for direction.

Similarly, the magnitude of the fusion model is:

$$\hat{m}_t^{F}{}_{,h} = w_m \hat{m}_t^{L}{}_{,h} + (1 - w_m)\hat{m}_t^{T}{}_{,h}, \qquad a_h \in [0,1],$$

where W_m represents the weight of LSTM on h th horizon in the fusion model for magnitude.

Then, let τ_h be the direction discrimination threshold of the h horizon, and the prediction of the price change after fusion is defined as

$$\hat{\Delta}_t^{F}{}_{,h} = \mathrm{sgn}(\hat{p}_t^{F}{}_{,h} - \tau_h)\hat{m}_t^{F}{}_{,h},$$

where $\hat{\Delta}_{t,h}^{F}$ represents the predicted price change at time point t hth horizon given by the fusion model.

This is because when two branches give different predicted directions, the pre-dicted price change of the fusion model might lose some important information, like fluctuation intensity, if it simply uses the mean value of the two predicted price changes. By contrast, this model generates sign and magnitude separately, allowing the financial meaning to be retained, which is essential for the following execution layer.

4.5. Trading Signal

In the execution layer, noise is filtered by the consensus mechanism. Essen-tially, such a double-horizon consensus mechanism aims to exchange lower trad-ing frequency for higher execution stability.

Let $j \in \{L, T, F\}$ represent LSTM, TCN, and fusion model, respectively, $h \in \{1, 3\}$ represent the prediction horizon. Then, define the predicted sign as:

$$s_t^{j}{}_{,h} = \mathrm{sgn}(\hat{p}^{j}{}_{t,h} - \tau_h) \in \{-1, +1\}.$$

If the three branches are in the same direction on the horizon on the first day, define:

$$A_t = \mathbb{I}(s_t^{L}{}_{,1} = s_t^{T}{}_{,1} = s_t^{F}{}_{,1}), \qquad s_t^{*} = s_t^{F}{}_{,1} \text{ when } A_t = 1.$$

Next, define the number of votes consistent with s_t^{*} on horizon on the third day as:

$$V_t = \mathbb{I}(s_t^{L}{}_{,3} = s_t^{*}) + \mathbb{I}(s_t^{T}{}_{,3} = s_t^{*}) + \mathbb{I}(s_t^{E}{}_{,3} = s_t^{*}).$$

Then the final trading signal is defined as:

$$q_t = s_t^* \mathbb{I}(A_t = 1)\, \mathbb{I}(V_t \geq 2), \qquad q_t \in \{-1, 0, +1\},$$

where $q_t = 1$ means long, $q_t = -1$ means short, and $q_t = 0$ means keep watching.

Once the consensus mechanism generates a trading signal at the time point t and the total assets of the current account are W_t, the system will make a transac-tion, according to the predicted direction, with the value of funds defined as:

$$I_t = \frac{q_t \times W_t\, \mathbb{I}(q_t \neq 0)}{n},$$

where this model takes 2 as the value of n, whose trading meaning is the maximum number of transaction orders simultaneously.

4.6. Trailing Stop Strategy

Regarding the trailing stop, assuming that the entry price is P_t^{in}, and the initial stop loss ratio is η, the stop loss positions of long and short positions are respec-tively:

$$P_{t,\text{stop}}^{\text{long}} = P_t^{\text{in}}(1 - \eta), \qquad P_{t,\text{stop}}^{\text{short}} = P_t^{\text{in}}(1 + \eta).$$

When the market moves toward the favorable direction, the system updates the mobile stop loss position according to the step size, ηP_t^{in}, to keep the trend profit as much as possible and control the retreat. Here, the model takes 0.01 as the η value, which means the system can always leave the market at a price where it is only 1 percent lower than the highest point.

Under this strategy, we will not be subject to the return ceiling brought by the profit stop position, and we can reasonably control the risk while ensuring the income at the same time.

5. Results and Analysis

5.1. Forecasting Results

From the results of the most representative fusion model, the performances of three types of stock indexes are significantly different. The direction prediction of Hang Seng index is relatively stable, and the direction accuracy of the first and third days in the future is 0.514; The direction accuracy of the NASDAQ composite index was 0.519 on the first day and dropped to 0.481 on the third day; The direction accuracy on the first day of Nikkei 225 is 0.504, and the one of the third days is only 0.436. Thus, with the increase of prediction span, the decline in the performance of direction prediction is more obvious. The result is shown in Table 1.

Table 1. Directional Forecasting Performance of the LSTM+TCN Fusion

Market	Day1 Sign Acc.	Day3 Sign Acc.
HSI	0.514	0.514
IXIC	0.519	0.481
N225	0.504	0.436

In terms of range prediction, the system generally underestimates the three types of stock indexes. The range percentage errors of Hang Seng Index on the 1st and 3rd days were about 0.554 and 0.639, respectively; The NASDAQ composite index is about 0.658 and 0.731; Nikkei 225 is about 0.648 and 0.706. This shows that although the model can combine the output direction and magnitude, the de-scription of the highly volatile market is conservative, especially in the prediction of further horizon, making it fail to provide the executive layer with effective sup-port. The result is shown in Table 2.

Table 2. Magnitude Forecasting Performance of the LSTM+TCN Fusion

Market	Day1 Mag. Error	Day3 Mag. Error	Day1 MAPE	Day3 MAPE
HSI	0.554	0.639	3.434	2.091
IXIC	0.658	0.731	0.985	0.868
N225	0.648	0.706	4.823	1.265

Still, the result of the fusion model is better than the two split branches in terms of magnitude prediction. Taking HSI as an example, almost all the indicators of the fusion model are lower than two split branches, which means that the fusion model has certain validity. The result is shown in Table 3.

Table 3. Magnitude Comparison of Three HSI Branches

Branch	Day1 Mag. Error	Day3 Mag. Error	Day1 MAE	Day3 MAE
LSTM	0.600	0.696	272.93	492.31
TCN	0.678	0.709	264.23	476.09
Fusion	0.554	0.639	272.68	457.39

Further analysis shows that the overall model is not stable and is significantly better than the strong baseline of always bullish, which matches the common phe-nomenon in financial forecasting research: when there is an obvious upward trend in the market within the sample range, as in the stock index market, the model may not have an advantage in statistical accuracy. This is also consistent with the conclusion of the recent systematic review of stock market prediction, that is, al-though accuracy is the most commonly used, it is not enough to fully reflect the actual value of the model[5].

5.2. Trading Results

Although the performance of the prediction layer is not really ideal, the results of the trading layer still have stable performance. According to the philosophy of this article, the most pivotal criteria of a trading system is its practical return rate. Next, this article will elaborate on the advantages and stability of the tailing strategy in this section.

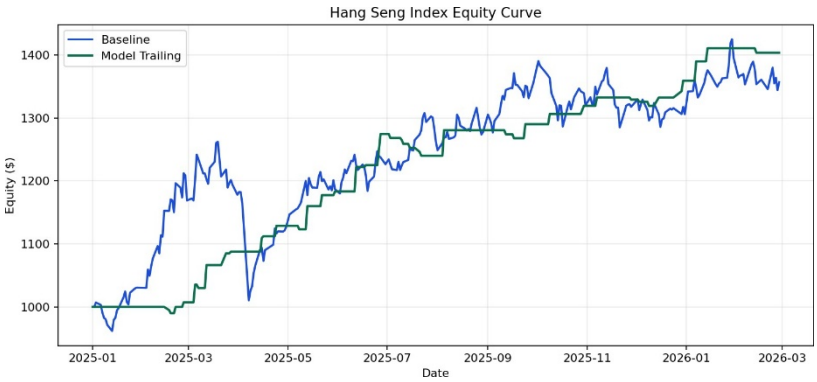


Figure 2. Hang Seng Index Equity Curve

On the Hang Seng Index (HSI), the system triggered a total of 92 transactions, including 56 long positions and 36 short positions, with a trigger accuracy of 0.554. The final wealth of the trailing stop loss strategy reached 1403.82, with an annual-ized return of about 34.03% , and 63 transactions were completed, with a winning rate of about 0.619, and a cumulative return of about 40.38% . The system in HSI not only uses the rising stage to be long, but also uses the falling stage to be short to a certain extent. Therefore, its stability is fully demonstrated by two-way trad-ing with returns higher than the baseline. The specific equity curve is shown in Figure 2.

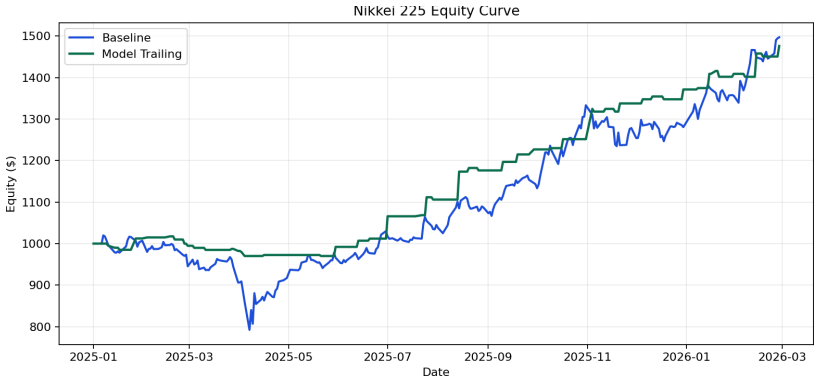


Figure 3. N225 Equity Curve

On Nikkei 225(N225), the system triggered 146 transactions, with a trigger accuracy of 0.534; The final wealth of the trailing stop loss strategy was 1476.17. The trailing strategy executed 86 transactions in total, with a winning rate of about 0.477, lower than HSI and IXIC, but the average return of a single transaction re-mained positive, with a cumulative return of about 47.62%. N225 has strong trend attributes within the evaluation range, so the stability of the market is mainly re-flected in its ability to follow trends, maintain profitability, quickly retreat against the trend, and maintain overall revenue structure stability. The specific equity curve is shown in Figure 3.

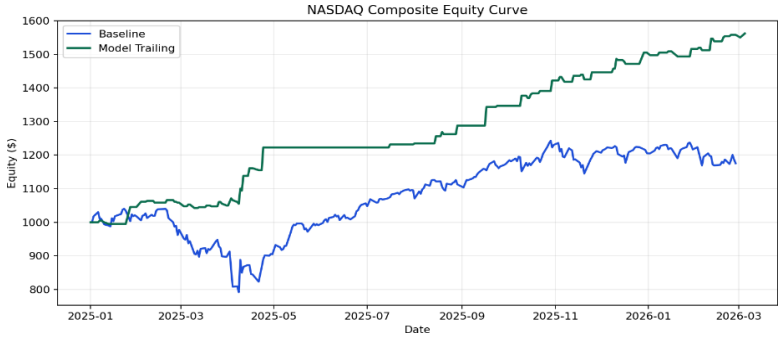


Figure 4. NASDAQ composite Equity Curve

On the NASDAQ composite index(IXIC), the consensus strategy triggered 235 trading signals, with a trigger accuracy of 0.557, all of which were long signals.

The final wealth of the trailing stop loss strategy reached 1561.98, with a cumula-tive return of about 56.20% and an annualized return of about 46.97%. From the trading statistics, the tailing strategy completed 110 transactions, with a winning rate of about 0.527. IXIC's results show that the advantage of the system is not that the direction prediction is absolutely ahead, but that it can more stably identify the

homeopathy stage, rapidly shrink the wrong trading through trailing stop loss, and continue to retain the effective market, so as to form a gradual accumulation of equity expansion. The specific equity curve is shown in Figure 4.

In general, IXIC demonstrated a stronger ability to expand revenue, HSI showed the effectiveness of bi-directional trading, and N225 reflected more obvious de-fensive attributes. Thus, the three show that the trailing stop strategy is the core source of the stability of the automated trading system in this article.

5.3. Result Comparison

Compared to other mainstream models, this system has certain advantages than those signal neural network models, but it is weaker than some of models that have a reinforcement mechanism, statistically. The comparison result is shown in Table 4.

Table 4. Return Rate Comparison Between Mainstream Models and This Study

Subject	annualized return rate
LSTM-based models [4]	1.0–1.2
DNN-based models [4]	1.0–1.2
HSI (this study)	34.03%
CNN-based models [4]	1.2–1.4
N225 (this study)	39.83%
IXIC (this study)	46.97%
RL-based models [4]	1.0–1.2 / 1.8–2.0

Still, because the survey does not clearly illustrate the trading tactics, including the trading frequency, stop loss strategy, of those models, we cannot rashly claim which model is essentially better.

6. Conclusion

In conclusion, this article draws three main points. Firstly, LSTM and TCN have significant complementarity in stock index time series modeling. LSTM is more suitable for extracting long-term dependencies and state transition information, while TCN is more suitable for extracting local fluctuations and multi-scale pat-terns. The fusion of the two can form a more robust output framework. Secondly, the fusion model is able to retain more effective financial information by adopting a separation weighting ensemble strategy, resulting in better performance.

Then, trading rules have a significant impact on the return. The model itself has merely medium-level direction prediction ability and even an unsuccessful magnitude prediction outcome, but through the consensus mechanism and trailing stop strategy, the system has formed a relatively stable income structure on dif-ferent stock indexes. Therefore, the results of the trading layer better reflect the actual value of this project than those of the prediction layer.

Of course, there are still some limitations to the project. Firstly, the magnitude prediction is relatively lower, indicating that the current feature and loss design do not adequately characterize extreme fluctuations. Secondly, the model still lacks generalization capability between different stock indices and has strong market specificity. Again, current research is mainly based on price and technical indi-cators and has not yet incorporated exogenous information such as macro, news, and sentiment. Subsequent work can incorporate fundamental language and mar-ket sentiment analysis to further enhance the robustness and practicality of the system.

References

[1] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054.

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[3] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271. https://doi.org/10.48550/arXiv.1803.01271.

[4] Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. Applied System Innovation, 4(1), 9. https://doi.org/10.3390/asi4010009.

[5] Lin, C. C., & Marques, J. A. L. (2024). Stock market prediction using artificial intelligence: A systematic review of systematic reviews. Social Sciences & Humanities Open, 9, 100864. https://doi.org/10.1016/j.ssaho.2024.100864.

[6] Aadhitya, A., Rajapriya, R., Vineetha, R. S., & Bagde, A. M. (2023). Predicting stock market time-series data using CNN-LSTM neural network model. arXiv preprint arXiv:2305.14378. https://doi.org/10.48550/arXiv.2305.14378.

[7] Dai, W., Wu, J., Zhang, X., et al. (2021). Price change prediction of ultra-high-frequency financial data based on temporal convolutional network. Expert Systems with Applications, 182, 115235. https://doi.org/10.1016/j.eswa.2021.115235.

[8] Luo, D., & Wang, X. (2024). ModernTCN: A modern pure convolution structure for general time series analysis. arXiv preprint arXiv:2403.01493. https://doi.org/10.48550/arXiv.2403.01493.

[9] Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017). A multi-horizon quantile recurrent forecaster. arXiv preprint arXiv:1711.11053. https://doi.org/10.48550/arXiv.1711.11053.

[10] López de Prado, M. (2018). Advances in financial machine learning. Wiley.

[11] Zong, C., Shao, J., Lu, W., & Zhuang, Y. (2024). Stock movement prediction with multimodal stable fusion via gated cross-attention mechanism. arXiv preprint arXiv:2406.06594. https://doi.org/10.48550/arXiv.2406.06594.