A Dual-Layer LSTM Stock Prediction Model Integrating Multi-Factor Attributes

Gesen Xu
Information and Computational Science, Guangdong University of Technology (Longdong Campus), Guangzhou, Guangdong, China,

Abstract: Stock price forecasting is a vital task in financial time series analysis. Traditional linear models (e.g., ARIMA, VAR, LASSO) fail to capture long-term dependencies and nonlinearities, while tree-based methods such as XGBoost depend heavily on feature engineering. This study proposes a dual-layer LSTM framework with differenced targets, Dropout regularisation, Huber loss, and adaptive optimisation strategies. Daily data of SPDB (2018–2024) are split into training, validation, and testing sets (70/10/20). Financial indicators are aligned with market data using carry-forward methods, and 12 high-quality factors are selected via IC analysis. The model is benchmarked against ARIMA, ARIMAX, XGBoost, and Transformer, evaluated by MAE, MSE, RMSE, and R^2. On the test set, the dual-layer LSTM achieves superior performance (MAE=0.0972, RMSE=0.1455, R^2=0.9848). Robustness and ablation analyses confirm that its deep architecture, resilient loss function, and factor integration collectively enhance accuracy, convergence, and stability, demonstrating its effectiveness in modelling complex financial time series.

Keywords: Stock Price Prediction; Machine Learning; Deep Learning; Long Short-Term Memory (LSTM); Financial Time Series

1. Introduction

Forecasting stock prices is a long-standing topic in financial time series research, aimed at improving investment decisions by uncovering historical patterns. Traditional models such as ARIMA[1] and VAR are theoretically comprehensive for stationary series but constrained by linear assumptions, limiting their ability to capture nonlinear volatility and long-term dependencies. Regression methods like LASSO (Tibshirani, 1996) [3] assist in factor selection but fail to model nonlinear interactions. More recently, tree-based models such as XGBoost (Chen & Guestrin, 2016) [4] have shown strength in nonlinear fitting and feature selection, though they still depend heavily on manual engineering and weakly represent temporal dynamics.

With deep learning advances, RNNs, LSTMs (Hochreiter & Schmidhuber, 1997) [6], and GRUs have become central to stock prediction, with LSTM widely applied in finance (Fischer & Krauss, 2018) [7]. Extensions include CNN-LSTM hybrids, attention-enhanced LSTMs, and Transformers for long-sequence modelling (Zhou et al., 2021) [8]. More recent studies integrate GANs with Transformers to enhance accuracy and robustness (Li & Xu, 2025) [9].

Nonetheless, existing work has two main shortcomings: shallow models often exhibit noise sensitivity and unstable convergence, and most studies rely on single variables such as closing prices, neglecting systematic factor engineering [5]. To address this, this study proposes a dual-layer LSTM with multi-factor integration. The framework employs Dropout, Xavier initialization, gradient clipping, dynamic learning rate scheduling, and Huber loss to enhance stability and generalization. Data are chronologically segmented and standardized, with high-quality factors selected via IC analysis to ensure consistency and reliability [10].

2. Research Methodologies and Innovative Elements

2.1 Research Methods

Long Short-Term Memory (LSTM) networks are an evolution of recurrent neural networks (RNNs). While RNNs derive temporal expressiveness from recurrent feedback loops, they often struggle with long-range dependencies due to vanishing or exploding gradients. LSTMs address these limitations by introducing a memory cell and gating mechanisms that selectively preserve and update information across time steps [11]. The memory cell acts as a “conveyor belt,” transmitting essential information while filtering redundant signals. This process is regulated by three gates-Forget, Input, and Output-that jointly balance memorization, update, and information release [12].

Forget Gate – determines the extent to which past memory is retained or discarded:

$$f_t = \sigma\left(\mathbf{W}_f \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_f\right) \qquad (1)$$

Input Gate & Candidate Memory – update the cell state with new information:

$$i_t = \sigma(\mathbf{W}_t[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_t) \qquad (2)$$

$$\underset{i}{\widetilde{C}} = \tanh\left(\mathbf{W}_e[\mathbf{x}_i, \mathbf{h}_{i-1}] + \mathbf{b}_c\right) \qquad (3)$$

$$\mathrm{C}_t = f_t \odot \mathrm{C}_{t-1} + i_t \odot \tilde{\mathrm{C}}_t \qquad (4)$$

Output Gate – regulates how much of the memory flows to the hidden state:

$$\sigma_t = \sigma(\mathbf{W}_\sigma[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_\sigma) \qquad (5)$$

$$\mathbf{h}_t = \sigma_t \odot \tanh\left(\mathbf{C}_t\right) \qquad (6)$$

Through this structured sequence of *forget–update–output*, LSTMs maintain stable gradient propagation and effectively capture nonlinear, long-term dependencies, outperforming conventional RNNs in financial time series forecasting.

As illustrated in Figure 1, the LSTM cell comprises three interacting gates-input, forget, and output-that collectively regulate information flow across time steps.

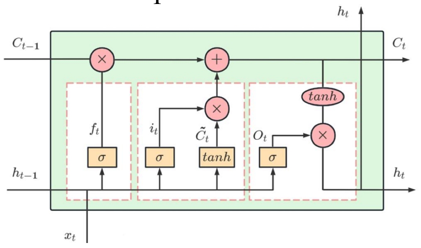

Figure 1. Schematic Diagram of LSTM Neurons

At the neural level, this mechanism is supported by activation functions that introduce nonlinearity, enhancing learning efficiency and representational capacity. The sigmoid and hyperbolic tangent (tanh) functions are fundamental in regulating information flow within the gates:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, 0 < \sigma(x) < 1 \qquad (7)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, -1 < \tanh(x) < 1 \qquad (8)$$

These activation functions ensure that information retention, update, and release occur within bounded ranges, thereby stabilizing training and improving the robustness of LSTM models in sequence prediction tasks.

2.2 Innovative Features

In financial time series forecasting, relying solely on closing prices inadequately reflects market dynamics [20]. It is thus necessary to integrate multiple indicators representing price movements, though excessive inclusion risks redundancy and noise [19]. To address this, a factor screening mechanism based on statistical testing is introduced, replacing traditional empirical or full-factor selection [13]. This process eliminates ineffective variables, enhances interpretability, and aligns prediction outcomes with factor variations. Financial and technical features are synchronised using a “disclosure lag + carry-forward” approach to ensure temporal consistency [18]. Information Coefficient (IC) analysis is applied to assess factor predictive power, combining IC values, *t*-statistics, and positive IC ratios as selection criteria. Ultimately, 12 core factors-including Bollinger Bands, momentum, moving averages, Fourier components, and ATR-are retained as inputs for LSTM modeling [14].

Building on a single-layer LSTM, a dual-layer hierarchical structure is proposed. The first layer captures local dependencies and short-term variations, while the second consolidates cross-period information to extract long-term trends, enabling progressive integration from local to global features [15].

To further improve stability and generalization, several optimizations are adopted. Standardization is fit only on the training set to prevent leakage. Huber loss replaces MSE for robustness to anomalies [16]. Gradient clipping, Dropout, L2 regularization, and adaptive learning rates ensure smooth convergence and limit overfitting. Xavier initialization mitigates gradient issues in deeper recursions. These combined structural and training enhancements strengthen the model’s ability to capture nonlinear dynamics and long-term dependencies [17].

3. Experimental Research

3.1 Data Acquisition

This study employs historical stock data of Shanghai Pudong Development Bank (SPDB) from 2018 to 2024, obtained via the Tushare platform. The dataset contains 14 daily-level fields, including trading date, stock code, open, high, low, close, pre-close, price change, percentage change, volume, value, and company fundamentals. The closing price (Close) is selected as the dependent variable, as it is less affected by intraday volatility and better reflects market consensus. A time series chart of SPDB's closing price is constructed to illustrate its dynamic trend and provide the foundation for subsequent modeling and analysis. As shown in Figure 2, a time-series chart of SPDB's closing price is constructed to illustrate its dynamic trend and provide the foundation for subsequent modeling and analysis.

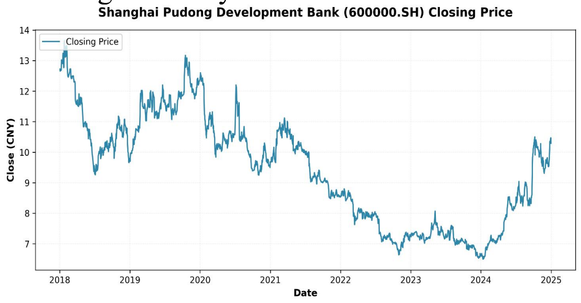

Figure 2. Shanghai Pudong Development Bank Stock Closing Price Time Series Chart

By examining the closing price trend of Shanghai Pudong Development Bank (2018–2024), the series is overall non-stationary with phased fluctuations. Prices fell sharply from 2018 to 2019, partially recovered in 2020–2021 but remained in a descending channel, reached a nadir in 2023, and rebounded markedly in early 2024, maintaining an upward trajectory thereafter. Despite volatility, the long-term trend over the last two years indicates a transition from decline to recovery.

Building on this analysis, company-level financial data are integrated to enhance factor comprehensiveness. Multidimensional disclosures-including profitability (e.g., ROE, gross and net profit margins), growth (e.g., revenue and net profit growth rates), operations (e.g., asset turnover, receivables turnover), and cash flow (e.g., operating and free cash flow)-are combined with market indicators such as trading price and volume. These variables provide fundamental perspectives on operating conditions and risk, while offering explanatory elements linked to stock price dynamics. The approach thus achieves feature-level fusion of financial and technical information, enabling a more complete characterization of SPDB's price evolution and establishing a robust foundation for factor screening and modeling. The obtained financial (partial) data is shown in Table 1.

Table 1. Financial Data(Partial)

Stock Name	YOYLiability	rogAvg	npMargin	YOYEquity	AssetTurnRatio	CFOTooR
SPDB	0.0632	0.0585	0.3109	0.0918	0.0135	2.7429

3.2 Data Preprocessing

The dataset, comprising raw daily market and multidimensional financial data, was rigorously cleansed and standardized to ensure reliability. Field anomalies were rectified through type conversions (e.g., date to datetime, numeric to float) and removal of erroneous values. Missing entries caused by disclosure schedules or trading suspensions were imputed using a unified forward–backward method across securities and time series.

To mitigate the impact of extreme values, outliers beyond the 3σ rule $[\mu-3\sigma,\mu+3\sigma]$ were detected and winsorized at both tails. Continuous variables were then normalized to $[-1,1]$ using:

$$X^{scaled}=2\times\frac{X-X_{\min}}{X_{\max}-X_{\min}}-1 \tag{9}$$

where $X_{\min}$ and $X_{\max}$ denote the minimum and maximum values during the sample period. This process compresses value ranges, maintains monotonicity, and prevents large-scale variables from dominating optimization, yielding clean and comparable inputs for factor selection and dual-layer LSTM modeling.

3.3 Feature Engineering

This research identifies five types of information-trend, volatility, momentum, volume, and cycle-by selecting 12 indicators from the technical and financial indicator pools based on IC size comparisons for incorporation into the model. Let the closing price, high price, low price, and trading volume be represented as c_t ,H_t,L_t,V_t. The n-day rolling windows for the simple moving average (SMA) and standard deviation (Std) are designated as

$$\mathrm{MA}_n(t)=\frac{1}{n}\sum_{i=0}^{n-1}C_{t-i} \tag{10}$$

$$\sigma_n(t)=\sqrt{\frac{1}{n-1}\sum_{i=0}^{n-1}\left(C_{t-i}-\mathrm{MA}_n(t)\right)^2} \tag{11}$$

Trend Category

1. **MA14** (14-day simple moving average)

$$\mathrm{MA14}(t)=\frac{1}{14}\sum_{i=0}^{13}C_{t-i} \tag{12}$$

2. **MA21** (21-day simple moving average)

$$\mathrm{MA21}(t)=\frac{1}{21}\sum_{i=0}^{20}C_{t-i} \tag{13}$$

3. **EWMA26** (26-day Exponentially Weighted Moving Average, $\alpha = 2/(26+1)$

$$\mathrm{EWMA26}(t) = \alpha C_t + (1-\alpha)\mathrm{EWMA26}(t-1), \alpha = \frac{2}{27} \quad (14)$$

4. **Bollinger Middle Band** (20)

$$\mathrm{BB\_}^{\mathrm{middle}}(t) = \mathrm{MA20}(t) \quad (15)$$

5. **Bollinger Band Upper Band** (20, 2σ)

$$\mathrm{BB\_upper}(t) = \mathrm{MA20}(t) + 2\sigma_{20}(t) \quad (16)$$

6. **Bollinger Band Lower Bound** (20, 2σ)

$$\mathrm{BB\_lower}(t) = \mathrm{MA20}(t) - 2\sigma_{20}(t) \quad (17)$$

Fluctuating Category

7. **ATR14** (14-day smoothed average of Wilder's Average True Range) First define the Average True Range

$$\mathrm{TR}(t) = \max\{H_t - L_t, |H_t - C_{t-1}|, |L_t - C_{t-1}|\} \quad (18)$$

Subsequently, acquire utilizing Wilder smoothing (or Simple Moving Average)

$$\mathrm{ATR14}(t) = \frac{1}{14}\sum_{i=0}^{13} \mathrm{TR}\,(t-i)(?\,ò\, ATR14(t) = \frac{13}{14}\mathrm{ATR14}(t-1) + \frac{1}{14}\mathrm{TR}(t)) \quad (19)$$

Momentum and Energy

8. **Price Momentum** (10-Day Logarithmic Momentum)

$$\mathrm{MOM10}(t) = \ln\left(\frac{C_t}{C_{t-10}}\right) \quad (20)$$

9. **Volume Momentum** (1-Day Logarithmic Momentum)

$$\mathrm{VolMOM1}(t) = \ln\left(\frac{V_t}{V_{t-1}}\right) \quad (21)$$

Periodic / Frequency Domain

(Apply a sliding window of length N at time t to perform the DFT on $(C_{t-N+1}, \dots, C_t)$; in this study, N is chosen to be consistent with the sampling rate, e.g., 64 or 128.)

Denote the complex coefficient of the k -th discrete frequency as

$$\hat{C}_k(t) = \sum_{n=0}^{N-1} C_{t-n}\, e^{-j2\pi kn/N} \quad (22)$$

10. **First-order harmonic amplitude** (Fourier-1)

$$\mathrm{Fourier}_{-1}(t) = \frac{1}{N}\left|\hat{C}_1(t)\right| \quad (23)$$

11. **Fifth-order harmonic amplitude** (Fourier-5)

$$\mathrm{Fourier}_{-5}(t) = \frac{1}{N}\left|\hat{C}_5(t)\right| \quad (24)$$

Robust Reinforcement

12. **20-day rolling standard deviation** (volatility proxy)

$$\mathrm{Std20}(t) = \sigma_{20}(t) \quad (25)$$

All factor calculations are conducted on a rolling window with a left-open and right-closed interval, ensuring that indicators at time t depend only on known data within $[t-n+1, t]$; Parameters (mean, standard deviation, rank transformation) are standardized on the training set and directly applied to validation and testing, while cross-sectional variables are aligned by same-day z-scores to ensure comparability and stationarity. Price-based indicators adopt intra-series standardization for temporal consistency.

After handling missing values, constant series, and extreme values, candidate factors are screened using the cross-sectional information coefficient (IC) to evaluate their linear relation with future returns. Since the dataset involves a single stock rather than a cross-sectional sample, the IC here is adapted to measure time-series correlations between factor values and future returns. The daily IC series is used to compute the mean, standard deviation, fraction of positive ICs, and Newey–West (HAC) adjusted *t*-statistics. Factors are selected according to absolute mean IC, statistical significance, and proportion of positive ICs, with robustness required across horizons $h \in \{1,5,10,20\}$ Based on IC ranking of correlation with future returns (see Table 2), the final set of high-quality factors is retained.

Table 2. Results of Information Coefficient (IC) Tests for Candidate Factors

Factor Name	IC	T	Positive IC Ratio
Momentum_10	0.230	2.17	65.0%
Lower_band	0.230	22.23	100.0%
Upper_band	0.134	12.43	100.0%
Volume_Momentum	0.133	1.94	57.0%
BB_middle	0.124	7.57	93.6%
Rolling_21	0.110	6.27	84.0%
EWMA_26	0.109	6.21	84.0%
MA14	0.108	6.10	84.0%
MA21	0.108	6.10	84.0%
Fourier_5	0.101	6.16	84.0%
Fourier_1	0.088	5.65	74.0%
ATR	0.025	2.35	70.0%

3.4 Evaluation Metrics

To comprehensively evaluate the performance of price prediction, this study employs three metrics: mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and the coefficient of determination (R^2). Let the sample size of the test set be n, with the actual and predicted values denoted by y_i and $\hat{y}_i$, respectively, and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. The four metrics are defined as follows:

Mean Absolute Error (MAE)

$$E_{\mathrm{MAE}} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (26)$$

The Mean Absolute Error (MAE) provides a clear depiction of the average deviation size, maintains the same dimensionality as the original pricing, is resistant to outliers, and enables reliable comparisons among various models.

Mean Square Error (MSE)

$$E_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \quad (27)$$

MSE is dimensionless and directly proportional to the square of the price. In contrast to MAE, it prioritizes the significance of substantial deviations, magnifying discrepancies in prediction errors quadratically. It functions as an essential statistic for assessing the overall fitting precision of a predictive model.

Root Mean Square Error (RMSE)

$$E_{\text{RMSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \quad (28)$$

RMSE possesses the same dimensionality as price; however, its squared component exacerbates significant mistakes, rendering it more responsive to spike variations. This facilitates the accurate detection of model discrepancies during times of significant volatility.

Coefficient of Determination (R²)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (29)$$

The R^2 metric quantifies the proportion of variance elucidated in relation to the "sample mean baseline," with a maximum value of 1 and no minimum limit (it may fall below 0). A greater value signifies a more robust fit.

All measures in this study are computed on a rigorously time-segmented test set; data normalization settings are only derived from the training set and applied to validation and testing to avert information leakage. In performing a thorough assessment, reduced MAE, MSE, and RMSE values, along with an elevated R^2 value, signify enhanced performance.

3.5 Model Construction

Time-series samples are constructed under strict information boundaries. Based on daily stock data and technical factors, the dataset is divided chronologically into training, validation, and test sets (70%, 10%, 20%). After splitting, scale transformation is applied to prevent information leakage. With a time step of L=90, a sliding window generates multivariate inputs: one channel represents the price-difference sequence, and 12 additional channels correspond to technical features (moving averages, exponential moving averages, Bollinger bands, volatility, ATR, momentum, frequency-domain harmonics, etc.). The choice of $L = 90$ reflects roughly one quarter of trading days, allowing the model to capture medium-term temporal dependencies and periodic market fluctuations while avoiding excessive sequence length that may hinder convergence. The RobustScaler is fitted on the training set. Consequently, the input tensor is represented as $(L, 1+F)$, where F denotes the number of technical features.

The predictive framework is a dual-layer LSTM network. The first layer contains 128 hidden units to capture temporal dependencies, and the second layer contains 64 hidden units to abstract higher-order features. Dropout (0.15) is applied between layers and outputs. Parameters are initialized with Xavier normalization. Huber loss is used as the training objective, and optimization is performed with Adam, accompanied by L2 regularization and gradient clipping (norm=1.0). Learning rate scheduling (ReduceLROnPlateau) and early stopping prevent overfitting, while batch size is set to 64 with a maximum of 100 epochs. Data order is strictly preserved to maintain temporal consistency.

3.6 Model Comparison

To evaluate the performance of the dual-layer LSTM model, this paper introduces ARIMA, ARIMAX, XGBoost, and Transformer models for comparison. To guarantee comparability, all models adhere to an identical data processing protocol: initially, time series data is segmented; subsequently, standardized parameters are fitted to the training set and applied to the validation and test sets; ultimately, the input window and target variable configurations remain uniform across models.

ARIMA

The ARIMA(p, d, q) model is utilized to represent the linear autoregressive structure of the price difference sequence.

$$\Phi(L)(1-L)^d y_t = \mu + \Theta(L)\varepsilon_i,\ \varepsilon_i \sim \text{i.i.d.}(0,\sigma^2) \quad (30)$$

Here, L denotes the lag operator, with .$\Phi(L) = 1 - \phi_1 L - \cdots - \phi_p L^p, \Theta(L) = 1 + \theta_1 L + \cdots + \theta_o L^q$ The price series is differenced once to achieve weak stationarity, and the orders are selected based on AIC, BIC, and residual diagnostics. For reporting purposes, ARIMA(1,1,0) is taken as a representative model for comparison. Parameters are estimated using maximum likelihood estimation, while the Ljung–Box test is applied to validate the adequacy of model fitting.

ARIMAX

In the ARIMA framework, the exogenous feature vector z_t (technical factors), consistent with the deep learning model, is incorporated to form the regression-type ARIMAX model:

$$\Phi(L)(1-L)^d y_t = \mu + \beta^{\top}\mathbf{z}_t + \Theta(L)\varepsilon_t \quad (31)$$

It can alternatively be represented as a lag polynomial of the exogenous variables, $\mathbf{B}(L)\mathbf{z}_t$. In contrast to the pure ARIMA model, ARIMAX preserves the ARMA structure of the residuals while explicitly integrating information on trends, volatility, and momentum, thus improving linear interpretability. The estimating and diagnostic processes are the same as those previously outlined.

XGBoost

Ensemble tree methods adopt an additive modeling framework

$$\widehat{Y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), f_k \in F_{CART} \quad (32)$$

and is optimized by a regularized objective function based on the second-order Taylor expansion:

$$L = \sum_i l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),\ \Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (33)$$

In this context, T represents the quantity of leaves, w_j the leaf weights, and γ, λ the regularization terms controlling model complexity and L_2 penalty. At each iteration, the first- and second-order gradients (g_i, h_i) of the samples are used to compute the gain and determine the optimal split. The input vector $\mathbf{x}_i$ is constructed by flattening and aggregating price differences and technical factors within a sliding window of length L. Hyperparameters (e.g., learning rate, maximum depth, subsampling rate) are tuned on the validation set to ensure comparability.

Transformer

This study utilizes a self-attention architecture consisting exclusively of stacked encoder layers to depict the sliding-window sequence. Denote the input sequence as $\mathbf{X} \in \mathbb{R}^{L \times F}$, which is linearly embedded into $\mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}_e \in \mathbb{R}^{L \times d_{\text{model}}}$, with positional encoding added to preserve temporal information. Each encoder layer consists of multi-head self-attention (MHA) and a feed-forward network (FFN), combined with residual connections and layer normalization in the Pre-LN form:

4. Experimental Results

4.1 Evaluation of the Predictive Performance of the Dual-Layer LSTM Model

Figure 3 compares the projected trajectory of the trained two-layer LSTM with the actual sequence over the first 250 observations. The two curves align closely in overall trend, peaks, troughs, and inflection points, with deviations mainly during periods of high volatility. The model responds effectively to short-term changes and consistently captures medium-term trends, demonstrating its ability to characterize time-dependent structures.

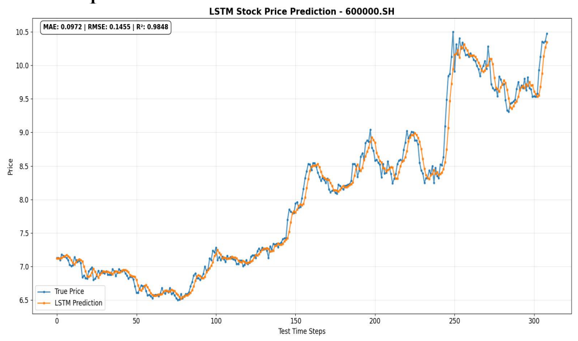

Figure 3. Dual-Layer LSTM Model Prediction Visualization

Within the consolidated training framework, the dual-layer LSTM reached its optimal validation loss (0.2416) around epoch 10, with adaptive learning rate decay and early stopping at epoch 25, indicating mild overfitting but stable convergence. On the test set, it achieved strong predictive accuracy (MAE=0.0972, RMSE=0.1455, R²=0.9848), explaining most variance in price fluctuations. Relative error metrics (MAPE=1.24%, sMAPE=1.25%) confirm minimal error magnitude. Visual comparisons show close alignment with trends, peaks, troughs, and inflection points, with deviations mainly during high volatility. Overall, the dual-layer LSTM demonstrates high efficacy in numerical fitting and trend analysis, though short-term directional classification leaves room for improvement.

4.2 Comparative Experimental Analysis

As summarized in Table 3, the dual-layer LSTM attains optimal performance with MAE = 0.0972, RMSE = 0.1455, and R² = 0.9848, while the Transformer achieves a slightly lower RMSE, reflecting greater sensitivity to large errors. LSTMs therefore show superiority in bias control and explained variance, whereas Transformers hold a minor advantage in RMSE.

Relative to statistical baselines, improvements are substantial. Compared with ARIMA (MAE=0.1602, RMSE=0.2451, R²=0.9186), the LSTM reduces MAE and RMSE by 39% and 41%, and raises R² by 0.066. Against ARIMAX

(MAE=0.1464, RMSE=0.2029, R^2=0.9710), it lowers MAE and RMSE by 34% and 28%, with an R^2 gain of 0.014, underscoring the limits of linear short-memory models in capturing nonlinear, long-range dependencies.

Table 3. Comparison of Model Metrics

Method	MAE	RMSE	R^2
ARIMA	0.1602	0.2451	0.9186
ARIMAX	0.1464	0.2029	0.9710
XGB	0.1472	0.2037	0.9705
Transformer	0.1029	0.1435	0.9722
Dual-layer LSTM	0.0972	0.1455	0.9848

Compared with XGBoost (MAE=0.1472, RMSE=0.2037, R^2=0.9705), the LSTM reduces MAE and RMSE by 34% and 29% while improving R^2 by 0.014, indicating that tree models, though effective in nonlinear fitting, underperform in temporal representation without sequential memory.

In comparison with Transformer, the LSTM achieves 5.5% lower MAE and higher R^2, while the Transformer yields 1.4% lower RMSE. Overall, by mean deviation and explained variance, the LSTM remains the superior model.

5. Melting Experiments and Analysis

5.1 Ablation Study 1

Table 4. Comparison of Metrics Between Single-Layer LSTM and Dual-Layer LSTM Models

Construction	MAE	RMSE	R^2
Single-layer LSTM	0.0994	0.1486	0.9841
Dual-layer LSTM	0.0972	0.1455	0.9848

As shown in Table 4, the dual-layer LSTM model notably attained steady convergence by the 41st iteration, but the single-layer LSTM necessitated ongoing iterations until the 55th iteration to obtain a similar level of convergence. The dual-layer structure attains loss convergence more rapidly, demonstrating enhanced efficiency in feature extraction and gradient propagation. As a result, both prediction accuracy and training efficiency are improved. This discovery corroborates the previous claim that "layered feature extraction alleviates gradient vanishing and enhances model stability," hence reinforcing the benefits of deep architectures in managing intricate financial time-series data.

5.2 Ablation Study 2

As summarized in Table 5, the model trained with Huber Loss showed marginal superiority over MSE across MAE, MASE, and R^2. Early stopping occurred at iteration 41 versus 75 for MSE, indicating faster convergence and greater stability. This advantage arises because MSE is highly sensitive to outliers, while Huber behaves like MSE for small errors and shifts to L1 for large ones, reducing the influence of extremes. Although final metrics differ little, Huber Loss achieves a better balance between convergence efficiency and generalization, ensuring higher accuracy with shorter training time. It is therefore adopted as a suitable loss for subsequent evaluations.

Table 5. Comparison of Huber Loss and MSE Loss

Construction	MAE	MASE	R^2
Huber Loss	0.0972	0.1455	0.9848
MSE Loss	0.0991	0.1492	0.9839

5.3 Ablation Study 3

Table 6. Comparison of Dual-Layer LSTM Model with Dropout vs. Dual-Layer LSTM Model without Dropout

Construction	MSE	Number of iterations
Dual-Layer LSTM (dropout)	0.8006	41
Dual-Layer LSTM (no dropout)	0.8013	50

This research further investigates the influence of Dropout on model convergence and generalization performance through comparing experiments on regularization procedures. In terms of convergence speed, the dual-layer LSTM with Dropout achieved early stopping at the 41st iteration, whereas the model without Dropout persisted until the 50th iteration; this suggests that Dropout enhances the pace of convergence to the optimal point on the validation set. The mean squared error (MSE) values for the Dropout and non-Dropout models were 0.8006 and 0.8013, respectively, indicating a minimal difference. In the experimental conditions, Dropout did not markedly enhance the final generalization error but did decrease the iterations needed for convergence, as presented in Table 6.

5.4 Ablation Study 4

Table 7. Comparison of Dual-Layer LSTM Model with Feature Factors vs. Dual-Layer LSTM Model without Feature Factors

Construction	MSE	R^2	Number of iterations
Dual-Layer LSTM(factor)	0.8006	0.9844	41
Dual-Layer LSTM (no factor)	0.8105	0.9840	75

The multi-factor model reached early termination at the 41st iteration, as shown in Table 7, but the univariate model necessitated iterations up to the 75th round for stable convergence. This suggests that the supplementary technological aspects expedited the model optimization process to some degree. Both models demonstrated comparable errors on the training and validation datasets. On the test set, the multi-factor model attained a mean squared error (MSE) of 0.8006, indicating a about 1.22% enhancement over the featureless model's MSE of 0.8105. Simultaneously, R^2 rose from 0.9840 to 0.9844, but MAE diminished from 0.0995 to 0.0982. Despite being rather modest, these increases regularly illustrate the beneficial influence of technical aspects on forecast accuracy.

6. Conclusion

In the experiments, the dual-layer LSTM outperformed ARIMA, ARIMAX, XGBoost, and even the Transformer-only model in predictive accuracy and goodness-of-fit. This highlights the strength of recurrent networks in capturing nonlinear dependencies and validates the role of hierarchical feature extraction in improving stability and generalization. Convergence efficiency was also superior: the dual-layer LSTM converged at iteration 41 versus 55 for the single-layer model, while replacing MSE with Huber Loss reduced convergence time (41 vs. 75 epochs) and enhanced robustness to extreme fluctuations. Dropout regularization further stabilized training and mitigated overfitting.

Incorporating technical factors identified via IC analysis provided consistent gains over single-variable inputs, enabling more effective integration of multi-source information and enhancing both predictive accuracy and interpretability. The dual-layer LSTM's advantages thus stem from the synergy of its hierarchical architecture and optimized training strategy, confirming its practical value in stock forecasting.

Nevertheless, limitations remain. The factor set emphasizes technical indicators while omitting sentiment or macroeconomic variables, and experiments are restricted to the banking sector, leaving generalization to other markets untested. Moreover, despite faster convergence than single-layer models, the architecture remains sensitive to hyperparameters and computationally demanding. Future research should expand factor types, including text and macroeconomic data, and conduct broader cross-market validations to strengthen robustness and applicability. In addition, subsequent studies could explore attention-enhanced recurrent structures (e.g., Transformer-LSTM hybrids) and employ Bayesian or reinforcement-learning-based optimization for adaptive hyperparameter tuning. Finally, integrating explainable AI tools such as SHAP or Grad-CAM can further improve interpretability and facilitate real-world financial decision-making.

References

[1] Shumway, R. H., & Stoffer, D. S. (2017). ARIMA models. In Time series analysis and its applications. Springer Texts in Statistics. Springer, Cham. https://doi.org/10.1007/978-3-319-52452-8_3

[2] Canova, F. (1995). The economics of VAR models. In K. D. Hoover (Ed.), Macroeconometrics (pp. 31–65). Recent Economic Thought Series, vol 46. Springer, Dordrecht. https://doi.org/10.1007/978-94-011-0669-6_3

[3] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288.

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

[5] Nelson, D. M. Q., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 1419–1426). IEEE. https://doi.org/10.1109/IJCNN.2017.7966019

[6] 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

[7] 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

[8] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325

[9] Li, S., & Xu, S. (2025). Enhancing stock price prediction using GANs and transformer-based attention mechanisms. Empirical Economics, 68(1), 373–403. https://doi.org/10.1007/s00181-024-02644-6

[10] Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. C. (2022). Predicting stock market index using LSTM. Machine Learning with Applications, 9, 100320. https://doi.org/10.1016/j.mlwa.2022.100320

[11] Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C.-W. (2023). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. Multimedia Systems, 29, 1751–1770. https://doi.org/10.1007/s00530-021-00758-w

[12] Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock price prediction using the ARIMA model. In 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (pp. 106–112). IEEE. https://doi.org/10.1109/UKSim.2014.67

[13] Wang, Y., & Guo, Y. (2020). Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost. China Communications, 17(3), 205–221. https://doi.org/10.23919/JCC.2020.03.017

[14] Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C., & Liu, P. (2022). Transformer-based attention network for stock movement prediction. Expert Systems with Applications, 202, 117239. https://doi.org/10.1016/j.eswa.2022.117239

[15] J.H. Ruan, K. Wang, T.Y. Feng, et al. (2025). Stock prediction methods based on Transformer. Computer and Digital Engineering, 53(5), 1375–1380, 1433.

[16] X. Yang. (2024). Stock price prediction based on an improved Transformer model (Master’s thesis, Jiangxi University of Finance and Economics). https://doi.org/10.27175/d.cnki.gjxcu.2024.000530

[17] W.D. He, G. He, & Y. Zhou. (2025). Stock price prediction based on a TPE-Informer-LSTM hybrid framework. Journal of Chongqing Technology and Business University (Natural Science Edition), 1–9. https://link.cnki.net/urlid/50.1155.N.20250703.1553.006

[18] S.C. Liu. (2025). Analysis of stock price prediction methods based on the LSTM machine learning model (Master’s thesis, Hangzhou Dianzi University). https://doi.org/10.27075/d.cnki.ghzdc.2025.000075

[19] R.Q. Sun. (2016). Research on a price-trend prediction model of U.S. stock indices based on LSTM neural networks (Master’s thesis, Capital University of Economics and Business).

[20] Y. Peng , Y.H. Liu, & R.F. Zhang. (2019). Modeling and analysis of stock price prediction based on LSTM. Computer Engineering and Applications, 55(11), 209–212.