Scilight

Complex Systems Stability & Control
https://www.sciltp.com/journals/cssc

Article

Check for updates

A BiLSTM–MLP–ReCast Method for Stock Price Forecasting

Feng Chen [1], Xinggui Li [1], Ruofeng Rao [2] and Jing Zhang [1,*]

[1] School of Mathematics and Computer Science, Panzhihua University, Panzhihua 617000, China
[2] School of Mathematics, Chengdu Normal University, Chengdu 611100, China
* Correspondence: zjpzh@pzhu.edu.cn

How To Cite: Chen, F.; Li, X.; Rao, R.; et al. A BiLSTM–MLP–ReCast Method for Stock Price Forecasting. *Complex Systems Stability & Control* **2026**, 2(3), 6. https://doi.org/10.53941/cssc.2026.100017

Received: 30 March 2026
Revised: 22 May 2026
Accepted: 22 May 2026
Published: 17 July 2026

Abstract: As financial market conditions continue to evolve, stock price sequences often exhibit strong nonlinear and nonstationary characteristics, reducing the effectiveness of traditional forecasting approaches. This paper proposes a stock price forecasting model based on the BiLSTM-MLP-ReCast framework. The model employs BiLSTM to capture bidirectional temporal dependencies in price sequences, integrates an MLP at the output stage for nonlinear transformation and introduces the ReCast mechanism to refine residual components, thereby mitigating lag effects and systematic bias. The framework is trained in an end-to-end manner, enabling joint optimization of temporal feature extraction, nonlinear mapping, and residual adjustment. Experimental results on three real-world stock datasets, including Google, Amazon, and Microsoft, demonstrate the effectiveness of the proposed framework. The complete BiLSTM-MLP-ReCast model achieved an R^2 of 0.9068 and an RMSE of 0.0520 on the Microsoft dataset. In addition, the model achieved the lowest MAPE values of 5.20% on Microsoft and 8.88% on Amazon, indicating improved prediction accuracy and robustness compared with conventional baseline models.

Keywords: stock price forecasting; deep learning; BiLSTM; MLP; ReCast

1. Introduction

The stock market is a key component of the financial system. Price movements reflect underlying economic conditions and market dynamics. Reliable stock price forecasting is therefore essential for investors and market participants [1]. However, financial time series are typically characterized by high volatility and strong nonlinearity, making this task particularly challenging [2].

Traditional forecasting methods, including simple moving averages (SMA), weighted moving averages (WMA), exponential smoothing (ETS), and naive methods, have long been used for stock price forecasting. These methods are mainly designed for linear patterns and often fail to capture complex nonlinear relationships, particularly in rapidly changing markets [3]. Recent review studies have highlighted the growing importance of hybrid deep learning and ensemble frameworks for modeling complex financial time series under volatile market conditions [4]. Deep learning provides a data-driven approach for financial time series modeling. CNN-based models extract local patterns from price data [5], while LSTM-based models capture long-term temporal dependencies [6,7]. Standard LSTM adopts a unidirectional structure and relies only on past information, which limits its ability to utilize full temporal context [8]. BiLSTM extends this structure by incorporating bidirectional information and provides a more complete contextual representation. In many stock forecasting studies, temporal features extracted by BiLSTM are directly connected to shallow fully connected forecasting layers [9,10]. Such architectures can achieve satisfactory forecasting performance, but shallow forecasting layers may not fully capture complex nonlinear interactions in highly volatile financial data. Siami-Namini et al. [11] reported that

BiLSTM improves predictive performance through better utilization of contextual information. Recent studies have explored hybrid architectures to improve nonlinear feature representation [12]. Sequence models such as RNNs, LSTMs, and GRUs still face challenges in modeling complex temporal dynamics. To address these limitations, recent studies have explored optimized recurrent architectures and attention-based mechanisms to improve nonlinear sequence modeling and long-term dependency learning in stock forecasting tasks [13]. Li et al. [14] proposed a stacked bidirectional GRU model based on smoothing and matrix decomposition, providing an effective strategy for mitigating long-term dependency issues. These approaches have also been extended beyond financial forecasting. Pan et al. [15] developed a multi-echelon tandem learning model for iron ore price forecasting, demonstrating the adaptability of such architectures in complex resource markets. Existing approaches still show limitations in capturing nonlinear interactions and handling dynamic error patterns in financial time series. The multilayer perceptron (MLP) is widely used as a nonlinear feature mapping module due to its strong representation capability. As a universal approximator, the multilayer perceptron can effectively capture complex nonlinear relationships, which making it suitable for modeling structured residual patterns in financial time series. Khan et al. [16] reported that integrating MLP with sequence models can improve nonlinear feature learning. Chen et al. [17] further demonstrated that combining BiLSTM with MLP can improve forecasting performance in time series forecasting tasks. In addition, recent hybrid forecasting frameworks have shown that decomposing structural components and modeling residual information can improve forecasting robustness in highly volatile stock markets [18].

However, existing deep learning models primarily focus on feature extraction and mapping, with limited consideration given to structured forecasting errors [19]. In highly volatile market environments, these errors often exhibit temporal patterns and may accumulate over time [20]. When these errors are not properly corrected, model outputs tend to exhibit forecasts lag or amplified bias [21]. Therefore, improving temporal modeling or nonlinear mapping alone is insufficient for capturing the full complexity of financial time series.

To address this issue, a hybrid forecasting framework, termed BiLSTM-MLP-ReCast, is proposed for next-day stock price forecasting. The proposed approach is built upon residual learning, where forecasting errors are regarded as structured discrepancies rather than random noise. BiLSTM captures bidirectional temporal dependencies, while MLP enhances nonlinear feature representation. The proposed ReCast mechanism is based on residual learning, in which the model learns the discrepancy between the forecasted trend and the actual observation instead of directly fitting the target values. This formulation enables the model to effectively capture structured residual patterns instead of treating them as noise. The forecasting task is decomposed into a coarse branch for trend estimation and a refinement branch for residual modeling. This decomposition aligns with the inherent characteristics of financial markets, where price movements typically consist of slow-varying trends and fast-changing fluctuations. This dual-branch design facilitates end-to-end optimization of both trend and residual components, thereby reducing forecasts lag and improving responsiveness to abrupt market changes. The integration of residual correction into the forecasting process improves adaptability under complex and highly volatile market conditions.

The main contributions of this study are summarized as follows:

- A unified BiLSTM-MLP-ReCast forecasting framework is developed, which integrates temporal modeling, nonlinear transformation, and residual refinement into a single end-to-end architecture.
- A residual learning-based ReCast mechanism is introduced to model structured forecasting errors and dynamically refine preliminary forecasts, thereby reducing forecasting lag and improving adaptability under volatile market conditions.
- Experiments on multiple real-world stock datasets show that the proposed approach achieves better forecasting accuracy and stability than baseline models.

The remainder of this paper is organized as follows. Section 2 reviews related studies on stock price forecasting, deep learning models, and hybrid forecasting frameworks. Section 3 presents the proposed BiLSTM-MLP-ReCast model and describes its overall architecture and learning process. Section 4 introduces the datasets, preprocessing procedures, experimental settings, and evaluation metrics. Section 5 discusses the experimental results, comparative analysis, and ablation studies. Finally, Section 6 presents the conclusion and future work.

2. Related Work

2.1. Deep Learning for Time Series Forecasting

Deep learning approaches have been extensively used in time-series forecasting. Stock price data exhibit strong temporal dependence and contextual characteristics, which makes recurrent neural networks suitable for this task. Traditional RNNs often suffer from vanishing or exploding gradients when dealing with long sequences, which limits their ability to retain long-term information [22].

LSTM addresses this issue by introducing gating mechanisms that regulate information flow, enabling more effective modeling of long-term dependencies. It has been widely applied in financial time-series forecasting. Standard LSTM operates in a unidirectional manner and relies mainly on past observations, which limits its ability to utilize full sequence context. BiLSTM extends LSTM by incorporating both forward and backward processing, allowing a more complete contextual representation. Previous studies indicate that BiLSTM generally achieves superior performance to unidirectional LSTM in volatile stock data environments [23]. In many existing forecasting architectures, the temporal representations generated by BiLSTM are directly mapped to forecasting outputs through shallow fully connected layers or linear forecasting layers. Such structures mainly perform weighted linear combinations of extracted temporal features, which may limit the ability of the model to capture higher-order nonlinear interactions and complex fluctuations in financial time series. Consequently, single-model architectures may still face difficulties in jointly modeling temporal dependencies and nonlinear feature relationships [24].

2.2. Hybrid Models for Financial Time Series

Given the limitations of single-model architectures, hybrid models have been increasingly explored in financial time-series forecasting. Existing studies suggest that hybrid approaches often outperform standalone deep learning models in terms of forecasting performance [25]. Combining BiLSTM with MLP has demonstrated promising results [26]. In this structure, BiLSTM captures bidirectional temporal dependencies, while MLP performs nonlinear transformation of extracted features, improving representation capability and enhancing forecasting accuracy. Despite these advantages, many existing hybrid models still rely on relatively simple stacking or feature fusion strategies. Such designs often lack mechanisms for refining intermediate forecasts. Under nonstationary conditions, these models may struggle to reduce forecasting bias, which leads to reduced model stability.

2.3. Residual Learning and Forecasting Refinement

Residual learning was originally introduced to alleviate the degradation problem in deep neural networks. Its key idea is to guide the model to learn residual mappings rather than directly fit the original function [27], while identity mappings improve gradient propagation [28].

This concept has been extended to time series forecasting as an error correction mechanism. In this setting, the model first generates preliminary forecasts and then models forecasting errors to refine the results. Studies have shown that incorporating residual information improves forecasting performance [29]. The introduction of error correction structures into LSTM-based models has demonstrated effectiveness in financial forecasting tasks [30]. Existing approaches still have limitations. Some methods rely on simple linear correction forms and cannot adequately capture nonlinear residual patterns. The correction module is often decoupled from the main model, restricting joint optimization. Under highly volatile market conditions, these limitations may reduce correction effectiveness.

2.4. Research Gap and Motivation

Existing deep learning and hybrid forecasting models have achieved promising results in financial time series forecasting, but several problems remain. Many studies mainly focus on temporal feature extraction, while the nonlinear transformation stage after sequence modeling is relatively simple and often relies on shallow forecasting layers [31]. This limits the ability of the model to capture complex nonlinear interactions in highly volatile stock price series. Some hybrid forecasting frameworks improve forecasting performance through feature fusion, but structured forecasting errors and residual patterns are still insufficiently explored. In financial markets, forecasting residuals often contain temporal dependency and systematic bias rather than purely random fluctuations [32]. Existing residual correction methods are also commonly implemented as separate post-processing procedures, which restricts joint optimization between the forecasting and refinement stages [33]. This may reduce adaptability under abrupt market fluctuations and nonstationary conditions. Based on these observations, this study proposes a unified BiLSTM-MLP-ReCast framework that combines bidirectional temporal modeling, nonlinear feature transformation, and residual refinement within an end-to-end architecture. The proposed framework aims to improve forecasting accuracy, reduce forecasting lag [34], and enhance robustness under volatile market conditions.

To further compare existing studies and position the proposed framework, Table 1 summarizes representative works on stock price forecasting using deep learning, hybrid architectures, and residual correction methods. The selected studies are compared in terms of model architecture, datasets, input features, forecasting horizons, evaluation metrics, and main limitations.

Table 1. Summary and comparison of related deep learning architectures for next-day stock price forecasting.

Study	Architecture & Main Limitation	Dataset	Input	Horizon	Metrics
Pawar et al. (2018) [3]	LSTM-based RNN; limited nonlinear feature transformation	NSE	closing prices	t + 1	RMSE, MAPE
Chen et al. (2020) [17]	Attention-MLP-BiLSTM; lacks residual refinement	Shanghai Composite & S&P 500	Prices, volume, indicators	t + 1	MAE, MSE, RMSE
Jiang (2021) [31]	DL Overview; highlights nonlinear bottleneck in linear outputs.	Various financial datasets	Mixed historical sequential and text features	Mixed	RMSE, MAE, Directional Accuracy
Kehinde et al. (2023) [4]	LSTM-Dense with Dual-Stage Attention; ignores systematic bias in prediction errors.	CSI 300, S&P 500	OHLCV	t + 1	MSE, MAE, MAPE
Varshney & Sharma (2024) [10]	Stacked DL with Error Correction; refinement performed separately (decoupled).	Financial indices	Prices, indicators	t + 1	RMSE, MAE, R^2
Li et al. (2025) [19]	Dual-branch GRU; lacks end-to-end joint gradient optimization.	CSI 300 and S&P 500	OHLCV, macro variables	t + 1	RMSE, MAE, Directional Accuracy
Proposed Work	BiLSTM-MLP-ReCast	Google, Amazon, Microsoft	OHLCV	t + 1	MSE, RMSE, R^2 MAPE, DA

3. Methodology

This section presents the proposed BiLSTM-MLP-ReCast framework. The overall architecture of the model is shown in Figure 1.

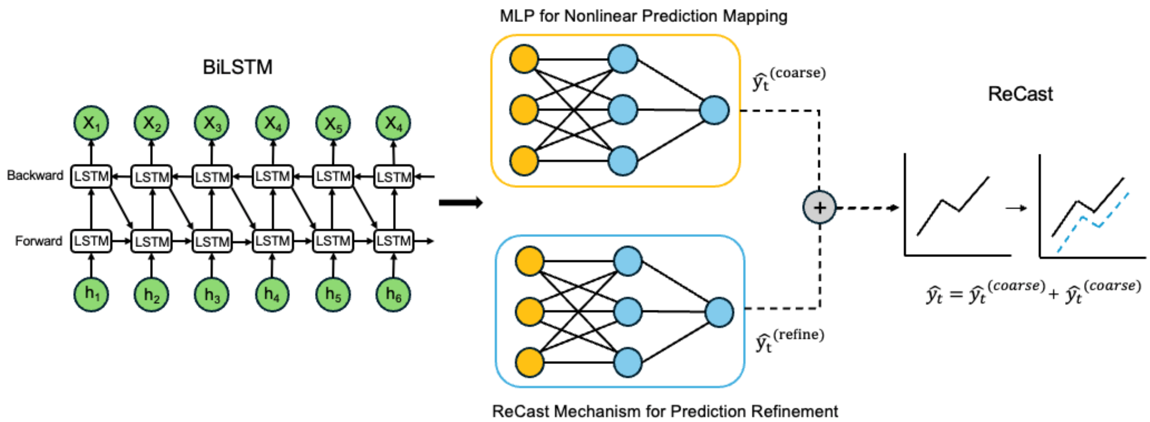

Figure 1. Structural diagram of the BiLSTM-MLP-ReCast hybrid forecasting model.

The model takes historical price sequences as input and performs temporal feature extraction, nonlinear mapping, and forecasting refinement to generate the final forecast. The BiLSTM module captures bidirectional temporal dependencies and produces robust feature representations. The MLP module further transforms these features through nonlinear mapping to generate preliminary forecasts. The ReCast module models and corrects forecasting residuals, reducing systematic bias. All components are trained jointly under a unified loss function, enabling end-to-end optimization of the framework.

3.1. Problem Formulation

Let x_τ denote the input feature vector at an internal sequence step τ. For forecasting at time step t, the historical stock sequence within a lookback window of length L is defined as:

$$X_t = [x_{t-L+1}, x_{t-L+2}, \dots, x_t] \quad (1)$$

where L denotes the lookback window length. The objective of the proposed model is to forecast the next-day closing price $\hat{y}_{t+1}$ based on the historical sequence X_t. The forecasting process can be expressed as:

$$\hat{y}_{t+1} = F(X_t) \tag{2}$$

where $F(\cdot)$ denotes the forecasting mapping learned by the proposed BiLSTM-MLP-ReCast framework.

3.2. BiLSTM Temporal Feature Extraction

To capture temporal dependencies in financial time series, the proposed framework employs a Bidirectional Long Short-Term Memory (BiLSTM) network for sequential feature extraction. Within the input sequence X_t, let x_τ denote the input feature vector at the internal sequence step τ and let $h_{\tau-1}$ denote the hidden state from the previous recurrent step. The gating mechanism of LSTM regulates information flow through forget, input, and output gates:

$$\mathrm{f}_\tau = \sigma(\mathrm{W_f}[h_{\tau-1}, \mathrm{x}_\tau] + \mathrm{b_f}) \tag{3}$$

$$i_\tau = \sigma(W_i[h_{\tau-1}, x_\tau] + b_i) \tag{4}$$

$$o_\tau = \sigma(W_o[h_{\tau-1}, x_\tau] + b_o) \tag{5}$$

where f_τ, i_τ and o_τ denotes the forget, input, and output gates at recurrent step τ.

In time-series forecasting, standard LSTM captures temporal dependencies in a single direction and mainly relies on past observations. BiLSTM extends this structure by processing the sequence in both forward and backward directions, enabling a more complete contextual representation. The forward and backward hidden states are computed as:

$$\overrightarrow{h_\tau} = \mathrm{LSTM}(x_\tau, \overrightarrow{h_{\tau-1}}) \tag{6}$$

$$\overleftarrow{h_\tau} = \mathrm{LSTM}\left(X_\tau, \overleftarrow{h_{\tau+1}}\right) \tag{7}$$

The final temporal representation is obtained by concatenating the hidden states from both directions:

$$H_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right] \tag{8}$$

where H_t denotes the concatenated bidirectional temporal representation. The representation H_t is subsequently transformed by the MLP module for nonlinear feature mapping.

3.3. Nonlinear Feature Transformation via MLP

The MLP module performs nonlinear feature transformation on the bidirectional temporal representation generated by the BiLSTM module, enabling the framework to capture complex feature interactions in financial time series.

Let H_t denote the bidirectional temporal representation generated by the BiLSTM module. The MLP module transforms H_t through two fully connected layers:

$$z_t = \phi(W_1 H_t + b_1) \tag{9}$$

$$m_t = W_2 z_t + b_2 \tag{10}$$

where z_t denotes the hidden nonlinear representation after ReLU activation, and m_t represents the transformed feature vector. W_1, W_2, b_1 and b_2 denote the weight matrices and bias vectors of the corresponding fully connected layers. Here, $\phi(\cdot)$ denotes the ReLU function. The transformed feature representation m_t is subsequently passed to the ReCast module for forecasting refinement and residual correction.

3.4. ReCast Residual Refinement Mechanism

In financial time-series forecasting, stock price movements often contain both slow-varying trends and fast-changing fluctuations. A single forecasting branch may struggle to capture these different temporal characteristics effectively. To address this issue, the proposed ReCast mechanism decomposes the forecasting process into a coarse branch for trend estimation and a refinement branch for residual correction. This structure improves forecasting responsiveness and model stability.

Based on this decomposition, the ReCast module produces a coarse forecast and a residual correction. Let m_t denote the transformed nonlinear feature representation generated by the MLP module. The coarse branch produces a preliminary trend forecast:

$$\hat{y}_{t+1}^{(c)} = \mathrm{W}_c m_t + b_c \tag{11}$$

The residual component is defined as the discrepancy between the target value and the coarse forecast:

$$e_{t+1} = y_{t+1} - \hat{y}_{t+1}^{(c)} \tag{12}$$

The refinement branch generates an estimated residual correction $\hat{r}_{t+1}$ to approximate the residual error e_{t+1}:

$$\hat{r}_{t+1} = \mathrm{W}_r m_t + b_r \tag{13}$$

The final forecast is obtained by adding the estimated correction to the coarse forecast:

$$\hat{y}_{t+1} = \hat{y}_{t+1}^{(c)} + \hat{r}_{t+1} \tag{14}$$

The residual correction term is learned through the global forecasting error during end-to-end training. Thus, the residual branch does not require a separately precomputed residual target during inference. Instead of treating forecasting errors as random noise, the proposed framework explicitly models residual variations for forecasting refinement. The final forecast combines the coarse trend estimate and the learned residual correction to capture both global trends and local fluctuations.

The entire framework is trained using the Mean Squared Error (MSE) loss function:

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{15}$$

where Θ denotes the set of model parameters. The MSE loss jointly optimizes the coarse forecasting and residual refinement branches under an end-to-end training framework.

3.5. Overall Forecasting Pipeline

The proposed framework follows a sequential forecasting pipeline consisting of BiLSTM, MLP, and ReCast modules. Historical stock sequences are first processed by the BiLSTM module to extract bidirectional temporal representations. The extracted features are then transformed by the MLP module through nonlinear mapping. Finally, the ReCast module generates a coarse forecast and a residual correction, which are combined to generate the final stock closing price forecast.

The overall forecasting process can be expressed as:

$$H_t = F_{\mathrm{BiLSTM}}(X_t) \tag{16}$$

$$m_t = F_{\mathrm{MLP}}(H_t) \tag{17}$$

$$\hat{y}_{t+1} = F_{\mathrm{ReCast}}(m_t) \tag{18}$$

where X_t denotes the input stock sequence, H_t denotes the bidirectional temporal representation extracted by the BiLSTM module, and m_t denotes the transformed nonlinear feature representation generated by the MLP module. The ReCast function $F_{\mathrm{ReCast}}(\cdot)$ performs the coarse forecasting and residual refinement process described in Section 3.4.

4. Experiments

This section evaluates the performance of the proposed BiLSTM-MLP-ReCast model on multiple real-world stock datasets. The experimental setup, data preprocessing procedures, and evaluation metrics are described in detail, followed by a comparison with baseline models.

4.1. Data Description

In this study, Google, Amazon, and Microsoft were selected to evaluate the proposed forecasting framework. These companies are representative technology firms with high trading activity and substantial market volatility. Compared with broad market indices, individual stocks usually exhibit stronger firm-specific fluctuations and more complex nonlinear price movements, making them suitable for evaluating temporal dependencies, nonlinear feature interactions, and forecasting residuals. The dataset spans from 31 July 2017, to 31 July 2023, and includes Open, High, Low, Close, Adjusted Close, and Volume features. Although longer historical series are available, long-term financial data often contain changing market conditions and structural shifts that may reduce the relevance of earlier patterns to current forecasting tasks. The selected six-year period covers multiple market phases, including the pre-pandemic period, the COVID-19 disruption, and the post-pandemic recovery stage. All data were obtained from Yahoo Finance. A one-step-ahead forecasting strategy was adopted. The dataset was divided chronologically into training, validation, and test sets using a standard ratio of 70:15:15. To provide a clearer understanding of the statistical characteristics of the selected datasets, descriptive statistics of the stock closing prices are summarized in Table 2.

Table 2. Descriptive statistics of stock closing prices.

Dataset	Mean	Std	Min	Max	Skewness	Kurtosis	Observations
Google	85.50	31.23	45.33	150.71	0.51	−1.13	1510
Amazon	113.99	37.10	46.93	186.57	0.23	−1.14	1510
Microsoft	195.08	81.50	71.41	359.49	0.10	−1.34	1510

4.2. Data Pre-Processing

The feature variables used in this study exhibit noticeable differences in numerical scale, particularly between price-related and volume-related variables. To reduce the impact of scale differences on model training, Min-Max normalization was applied to linearly map each feature to the range (0, 1). This preprocessing step helped stabilize parameter updates and improve training convergence. A sliding lookback window with a length of L = 20 trading days was used to construct input-output pairs for one-step-ahead forecasting. To avoid data leakage during preprocessing and model training, the normalization parameters were estimated exclusively from the training set and then applied unchanged to the validation and test sets, ensuring that future observations were not introduced into the training process. All evaluation metrics reported in this study were calculated on normalized values to reduce the influence of absolute price scale differences among different stocks and to ensure fair cross-dataset comparison. Consequently, the evaluation primarily reflects the relative forecasting performance and temporal pattern modeling capability of different architectures.

4.3. Metrics and Implementation Details

Stock price forecasting is a complex task requiring multiple evaluation metrics to assess model performance. In this study, MSE, RMSE, and R^2 are used to evaluate the predictive accuracy and stability of the model.

(1) Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{19}$$

where n denotes the number of samples, y_i represents the true value of the i-th sample, and $\hat{y_i}$ denotes the corresponding predicted value. A lower MSE indicates a smaller deviation between predicted and actual values.

(2) Root Mean Squared Error

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{20}$$

RMSE measures the average magnitude of forecasting errors. A lower RMSE value indicates better predictive performance.

(3) Coefficient of Determination

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{21}$$

where $\bar{y}$ denotes the mean of the observed values. A value of R^2 closer to 1 indicates a better fit of the model.

All experiments were conducted on a server running the Linux operating system. The hardware configuration includes an NVIDIA RTX A6000 graphics card. The model was implemented using Python 3.11.9 and the PyTorch 2.4.0 deep learning framework.

The BiLSTM network adopted a bidirectional architecture and was trained using the Adam optimizer with dropout to reduce overfitting. Hyperparameters were selected based on prior studies on stock forecasting [26] and further refined through validation experiments. Grid search was performed over the lookback window length $\mathrm{L} \in \{10, 20, 30\}$, hidden dimension $\mathrm{H} \in \{32, 64, 128\}$ and the number of BiLSTM layers $\mathrm{N} \in \{1, 2, 3\}$. Dropout rates between 0.1 and 0.4, learning rates within $\{10^{-3}, 10^{-4}, 10^{-5}\}$, and batch sizes within {16, 32, 64} were explored during the validation-based hyperparameter search. The final hyperparameter settings were selected according to validation performance and summarized in Table 3. The optimal hyperparameter combination was determined based on the lowest validation-set MSE and then used for final test-set evaluation. Baseline models were also individually fine-tuned within comparable parameter ranges to ensure a fair comparison.

On the specified hardware configuration, the average training time of the proposed framework is approximately 60 s per dataset, comparable to standalone BiLSTM models. The computational complexity was mainly dominated by the BiLSTM module. For a lookback window of length L, hidden dimension H, and input feature dimension D, the computational complexity can be approximated as $O(L(H^2 + HD))$. Since the recurrent hidden-state computation term H^2 dominates the overall cost, the complexity can be further approximated as $O(LH^2)$. In terms of space complexity, the parameter complexity was mainly determined by the BiLSTM module and can be approximated as $O(H^2 + HD)$. The MLP and ReCast modules mainly consisted of lightweight feedforward layers with relatively few additional parameters. Although nonlinear activation functions were used in these modules, they did not introduce recurrent computations across time steps. Therefore, their additional computational and memory overhead remained relatively small while still enabling effective nonlinear feature transformation and residual refinement.

Table 3. Model hyperparameter settings.

Parameter	Value
Window length	20
Hidden dimension	64
Number of layers	2
Dropout rate	0.2
Batch size	32
Learning rate	1×10^{-5}
Epochs	64

5. Results

The proposed model was trained and evaluated on the test set. The forecasting results for the Google, Amazon, and Microsoft datasets were illustrated in Figures 2, 3, and 4, respectively.

Across all three datasets, the predicted curves closely follow the actual price trends, indicating that the proposed model can effectively capture the underlying dynamics of stock price movements. To further evaluate the predictive performance, quantitative comparisons with baseline models were conducted, and the results were summarized in Tables 4–6.

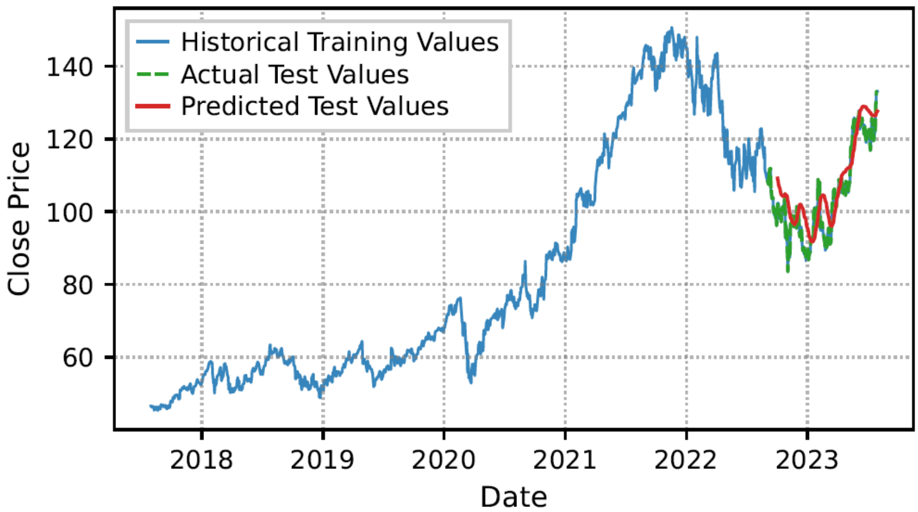

Figure 2. Forecasting results of the BiLSTM-MLP-ReCast model on the Google stock dataset.

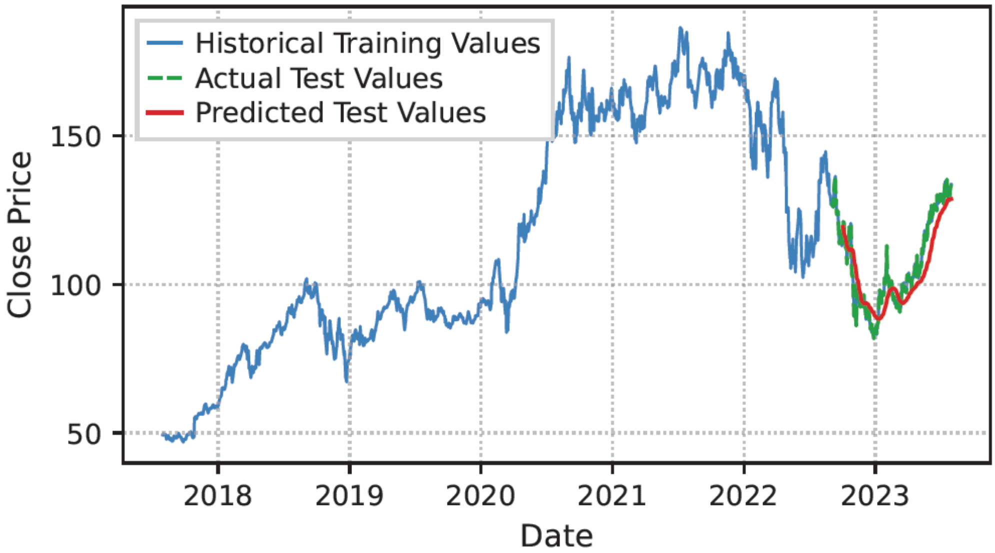

Figure 3. Forecasting results of the BiLSTM-MLP-ReCast model on the Amazon stock dataset.

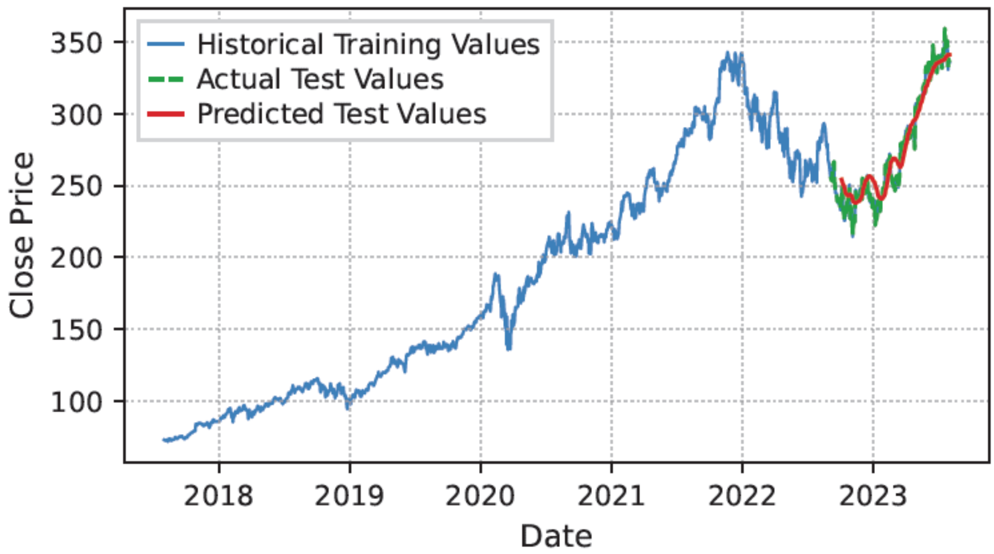

Figure 4. Forecasting results of the BiLSTM-MLP-ReCast model on the Microsoft stock dataset.

Table 4. Performance comparison on the Google stock dataset.

Model	MSE	RMSE	R^2
BiLSTM	0.004160	0.064496	0.732201
BiLSTM-MLP	0.004240	0.065112	0.727060
BiLSTM-MLP-ReCast	0.003279	0.057266	0.788880

Table 5. Performance comparison on the Amazon stock dataset.

Model	MSE	RMSE	R^2
BiLSTM	0.002811	0.053020	0.733939
BiLSTM-MLP	0.002399	0.048979	0.772945
BiLSTM-MLP-ReCast	0.002021	0.044958	0.808700

Table 6. Performance comparison on the Microsoft stock dataset.

Model	MSE	RMSE	R^2
BiLSTM	0.007848	0.088587	0.729832
BiLSTM-MLP	0.004538	0.067366	0.843769
BiLSTM-MLP-ReCast	0.002709	0.052044	0.906752

To further illustrate the differences among models, the forecasting results of different methods are shown in Figures 5–7.

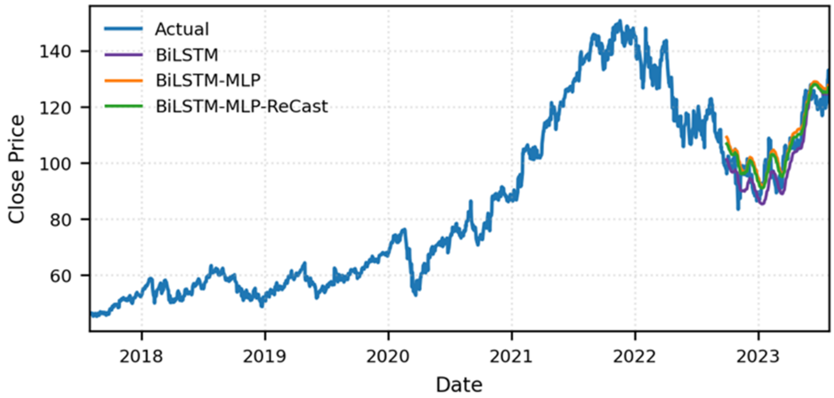

Figure 5. Comparison of forecasting results of different models on the Google stock dataset.

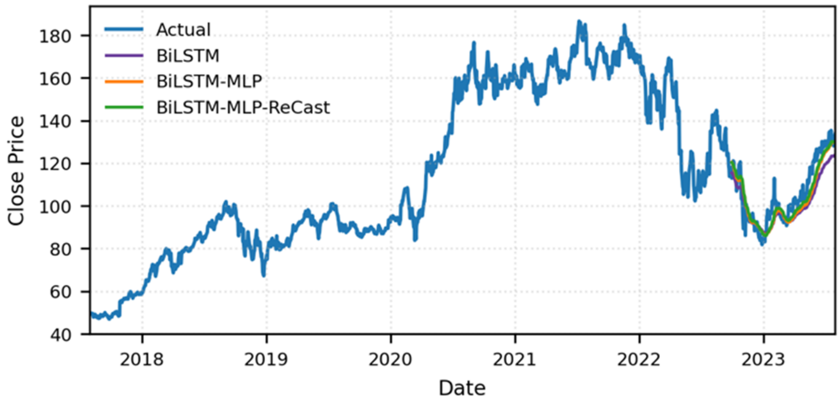

Figure 6. Comparison of forecasting results of different models on the Amazon stock dataset.

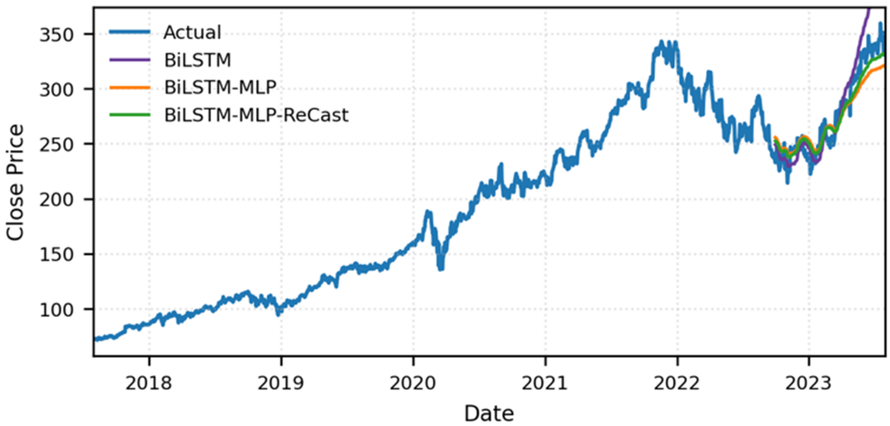

Figure 7. Comparison of forecasting results of different models on the Microsoft stock dataset.

The BiLSTM-MLP-ReCast model tracks actual price trends more closely than baseline models, exhibiting smaller deviations, especially under high volatility.

5.1. Experiment Analysis

On the Google dataset, the BiLSTM-MLP-ReCast model achieved the best performance across all evaluation metrics. Compared with the BiLSTM model, the proposed model achieves a reduction in Mean Squared Error (MSE) from 0.004160 to 0.003279, representing a reduction of approximately 21.2%. The Root Mean Squared Error (RMSE) decreased to 0.057266. Meanwhile, the R^2 value increased to 0.7889, corresponding to improvements of about 7.8% and 8.5% compared with BiLSTM and BiLSTM-MLP, respectively. These results indicate a closer fit to the actual price trends. The most significant improvement was observed on the Microsoft dataset. The MSE decreased from 0.007848 to 0.002709, representing a reduction of approximately 65%, while the RMSE decreased to 0.052044. The R^2 value increased to 0.9068, corresponding to improvements of approximately 24.3% and 7.5% compared with BiLSTM and BiLSTM-MLP, respectively. This indicates that the proposed model performs particularly well on time series exhibiting strong trends and structural variations. Overall, the proposed model achieves consistent improvements across different datasets.

5.2. Ablation Experiment

The ablation results indicate that the contribution of each module varies across datasets. On the Amazon dataset, the complete BiLSTM-MLP-ReCast framework achieves the best overall performance, reducing RMSE from 0.053020 in the BiLSTM model to 0.044958 and improving R^2 from 0.733939 to 0.808700. On the Microsoft dataset, the proposed framework further reduces RMSE from 0.088587 to 0.052044 while improving R^2 from 0.729832 to 0.906752. These results demonstrate that the combination of nonlinear feature transformation and residual refinement is particularly effective under stronger market volatility and more complex nonlinear dynamics. On the Google dataset, the BiLSTM-ReCast variant slightly outperforms the complete model, achieving an RMSE of 0.056278 compared with 0.057266 for the full framework. This suggests that when market fluctuations are relatively smooth, the additional nonlinear mapping introduced by the MLP module may provide limited improvement while slightly increasing sensitivity to local noise. Nevertheless, the complete framework maintains competitive performance on the Google dataset while demonstrating stronger overall generalization performance across multiple datasets. Removing the ReCast module generally increases forecasting errors and reduces R^2 values, indicating the importance of residual refinement for forecasting stability. Removing the MLP module also reduces forecasting performance on more volatile datasets, although its contribution varies across different market conditions.

5.3. Comparative Experiments

Comparative experiments were conducted on three stock datasets (Google, Amazon, and Microsoft) to evaluate the forecasting performance of the proposed framework against conventional deep learning models. CNN, LSTM, and BiLSTM were selected as baseline methods, and all models were evaluated under the same data partitioning strategy and evaluation metrics. The comparative results are summarized in Table 7. The proposed BiLSTM-MLP-ReCast framework achieves competitive forecasting performance across all datasets and generally outperforms the baseline models in terms of RMSE and R^2. Compared with CNN and standard LSTM models, the proposed framework provides more stable forecasting performance under volatile market conditions. The results indicate that integrating bidirectional temporal modeling, nonlinear feature transformation, and residual refinement improves the ability of the framework to capture complex stock price dynamics. Mean Absolute Percentage Error (MAPE) and Directional Accuracy (DA) were also incorporated to provide additional evaluation perspectives. The corresponding results are presented in Table 8. The proposed BiLSTM-MLP-ReCast framework achieves competitive MAPE performance across all datasets while maintaining comparable directional consistency. The results confirm the robustness and effectiveness of the proposed framework.

Table 7. Comparison of evaluation indicators of the different variants.

Model	Google		Amazon		Microsoft	
	RMSE	R^2	RMSE	R^2	RMSE	R^2
CNN	0.061812	0.754023	0.049859	0.764711	0.076062	0.800828
LSTM	0.076314	0.625067	0.048054	0.781440	0.061555	0.869557
BiLSTM	0.064496	0.732201	0.053020	0.733939	0.088587	0.729832
MLP-ReCast	0.060779	0.762178	0.056728	0.695417	0.078723	0.786647
BiLSTM-ReCast	0.056278	0.796096	0.055297	0.710589	0.065339	0.853028
BiLSTM-MLP	0.065112	0.727060	0.048979	0.772945	0.067366	0.843769
BiLSTM-MLP-ReCast	0.057266	0.788880	0.044958	0.808700	0.052044	0.906752

Table 8. Additional evaluation results in terms of MAPE and DA.

Model	Google		Amazon		Microsoft	
	MAPE	DA	MAPE	DA	MAPE	DA
CNN	8.5540	45.15	10.4513	53.40	7.0036	47.09
LSTM	10.5763	48.54	9.7106	52.91	5.7293	47.09
BiLSTM	9.0012	43.69	11.3496	51.46	7.3171	45.63
MLP-ReCast	8.6082	49.03	11.7821	56.31	7.2565	48.06
BiLSTM-ReCast	8.0268	44.66	12.0154	51.46	6.2207	45.15
BiLSTM-MLP	10.0783	47.09	9.5921	53.88	6.4868	46.60
BiLSTM-MLP-ReCast	8.7643	44.17	8.8783	51.94	5.2016	45.63

6. Conclusions and Future Work

6.1. Conclusions

This paper proposes a hybrid stock price forecasting framework based on BiLSTM, MLP, and ReCast modules. The framework integrates bidirectional temporal modeling, nonlinear feature transformation, and residual refinement within a unified end-to-end architecture. Experimental results on Google, Amazon, and Microsoft stock datasets demonstrate the effectiveness of the proposed method for financial time series forecasting. Compared with conventional baseline models, the proposed architecture delivers competitive forecasting accuracy across multiple datasets. On the Microsoft dataset, the complete BiLSTM-MLP-ReCast model achieved an R^2 of 0.9068 and an RMSE of 0.0520, together with a MAPE of 5.20%. On the Amazon dataset, the model achieved the best overall forecasting performance, with an RMSE of 0.0450 and an R^2 of 0.8087. For the Google dataset, it also maintained stable forecasting capability and achieved an R^2 of 0.7889. These results indicate that integrating temporal modeling, nonlinear representation, and residual refinement can effectively improve forecasting accuracy and robustness for complex stock price forecasting tasks. Several limitations remain in the proposed framework. The current model mainly relies on historical OHLCV data and does not incorporate external information such as news sentiment, macroeconomic indicators, or market events. In addition, the experiments focus primarily on statistical forecasting performance and do not consider practical trading constraints such as transaction costs, slippage, or portfolio execution strategies. The forecasting improvements are relatively modest on datasets with smoother market fluctuations and lower volatility. Future work will explore the integration of external financial information, lightweight model architectures, and more realistic trading-oriented evaluation strategies to improve both forecasting performance and practical applicability.

6.2. Future Work

Future work will focus on improving model efficiency by simplifying its structure and reducing computational overhead. Incorporating additional information, such as macroeconomic indicators and sentiment data may help better capture complex market dynamics and improve robustness under different market conditions. Extending the model to multi-step forecasting tasks may further improve its practical usefulness, particularly for real-world decision-making scenarios.

Author Contributions

F.C.: conceptualization, methodology, software, data curation, formal analysis, investigation, validation, writing—original draft preparation; X.L.: resources; R.R.: resources; J.Z.: supervision, project administration, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This research was supported by the Open Research Fund of the Sichuan Provincial Engineering Technology Research Center for BIM+ Application and Intelligent Visualization (Grant No. BIM-2024-Z-01).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The authors confirm that the data supporting the findings of this study are available within the article.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

No AI tools were utilized for this paper.

References

1. Sezer, O.B.; Gudelek, M.U.; Ozbayoglu, A.M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. https://doi.org/10.1016/j.asoc.2020.106181.
2. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669.
3. Pawar, K.; Jalem, R.S.; Tiwari, V. Stock Market Price Forecasting Using LSTM RNN. In *Emerging Trends in Expert Applications and Security*; Springer: Singapore, 2018; pp. 493–503.
4. Kehinde, T.O.; Chan, F.T.S.; Chung, S.H. Scientometric review and analysis of recent approaches to stock market forecasting: Two decades survey. *Expert Syst. Appl.* **2023**, *213*, 119299. https://doi.org/10.1016/j.eswa.2022.119299.
5. Valova, I.; Gueorguieva, N.; Aayushi, T.; et al. Hybrid Deep Learning Architectures for Stock Market Forecasting. In Proceedings of the 9th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'23), London, UK, 3–5 August 2023.
6. Mehtab, S.; Sen, J.; Dutta, A. Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models. In *Machine Learning and Metaheuristics Algorithms, and Applications*; Springer: Singapore, 2021; pp. 88–106.
7. Sang, S.; Li, L. A Stock Prediction Method Based on Heterogeneous Bidirectional LSTM. *Appl. Sci.* **2024**, *14*, 9158. https://doi.org/10.3390/app14209158.
8. Jayanth, T.; Manimaran, A.; Siva, G. Enhancing Stock Price Forecasting with a Hybrid SES-DA-BiLSTM-BO Model. *IEEE Access* **2024**, *12*, 173618–173637.
9. Zhang, J.; Ye, L.; Lai, Y. Stock Price Prediction Using CNN-BiLSTM-Attention Model. *Mathematics* **2023**, *11*, 1985. https://doi.org/10.3390/math11091985.
10. Varshney, R.P.; Sharma, D.K. Optimizing Time-Series Forecasting Using Stacked Deep Learning Framework with Error Correction. *Expert Syst. Appl.* **2024**, *249*, 123487.
11. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The Performance of LSTM and BiLSTM in Forecasting Time Series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019.
12. Karim, M.E.; Foysal, M.; Das, S. Stock Price Prediction Using Bi-LSTM and GRU-Based Hybrid Deep Learning Approach. In *Proceedings of Third Doctoral Symposium on Computational Intelligence*; Springer Nature: Singapore, 2023; pp. 701–711.
13. Oyewola, D.O.; Kehinde, T.O.; Akinwunmi, S.A.; et al. Stock market prediction with optimized PLSTM-AL in smart urban cities. *Financ. Res. Open* **2025**, *1*, 100019. https://doi.org/10.1016/j.finr.2025.100019.
14. Khan, W.A.; Chung, S.H.; Liu, S.Q.; et al. Smoothing and Matrix Decomposition-Based Stacked Bidirectional GRU Model for Machine Downtime Forecasting. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 7215–7227.
15. Pan, W.; Liu, S.Q.; Kumral, M.; et al. Iron Ore Price Forecast based on a Multi-Echelon Tandem Learning Model. *Nat. Resour. Res.* **2024**, *33*, 1969–1992. https://doi.org/10.1007/s11053-024-10360-2.
16. Khan, W.A.; Chung, S.; Awan, M.U.; et al. Machine learning facilitated business intelligence (Part I): Neural networks learning algorithms and applications. *Ind. Manag. Data Syst.* **2020**, *120*, 164–195.
17. Chen, Q.; Zhang, W.; Lou, Y. Forecasting Stock Prices Using a Hybrid Deep Learning Model Integrating Attention Mechanism, Multi-Layer Perceptron, and Bidirectional Long-Short Term Memory Neural Network. *IEEE Access* **2020**, *8*, 117365–117376. https://doi.org/10.1109/access.2020.3004284.
18. Kehinde, T.O.; Adedokun, O.J.; Kareem, M.K.; et al. STL-ELM: A computationally efficient hybrid approach for predicting high volatility stock market. *Intell. Syst. Appl.* **2025**, *27*, 200564.
19. Li, J.; Li, Z.; Lei, W. The LSTM Model with Error Rectification in Stock Price Prediction. In Proceedings of the 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 3–5 December 2021. https://doi.org/10.1109/mlbdbi54094.2021.00035.

20. Zhang, S.; Chen, Y.; Zhang, W.; et al. A Novel Ensemble Deep Learning Model with Dynamic Error Correction. *Inf. Sci.* **2021**, *544*, 427–445.
21. Wan, W.D.; Bai, Y.; Lu, Y.N.; et al. A Hybrid Model Combining a Gated Recurrent Unit Network Based on Variational Mode Decomposition with Error Correction for Stock Price Prediction. *Cybern. Syst.* **2024**, *55*, 1205–1229. https://doi.org/10.1080/01969722.2022.2137634.
22. Pascanu, R.; Mikolov, T.; Bengio, Y. On the Difficulty of Training Recurrent Neural Networks. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
23. Kim, T.; Kim, H.Y. Forecasting Stock Prices with a Feature Fusion LSTM-CNN Model. *PLoS ONE* **2019**, *14*, e0212320.
24. Khashei, M.; Bijari, M. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Appl. Soft Comput.* **2011**, *11*, 2664–2675. https://doi.org/10.1016/j.asoc.2010.10.015.
25. Farhadi, A.; Zamanifar, A.; Alipour, A.; et al. A Hybrid LSTM-GRU Model for Stock Price Prediction. *IEEE Access* **2025**, *13*, 117594–117618. https://doi.org/10.1109/access.2025.3586558.
26. Liu, Z. Improving Stock Price Forecasting with M-A-BiLSTM: A Novel Approach. *Front. Appl. Math. Stat.* **2025**, *11*, 1588202.
27. He, K.; Zhang, X.; Ren, S.; et al. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. He, K.; Zhang, X.; Ren, S.; et al. Identity Mappings in Deep Residual Networks. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645.
29. Zhang, G.P. Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. *Neurocomputing* **2003**, *50*, 159–175. https://doi.org/10.1016/s0925-2312(01)00702-0.
30. Liu, Y.; Duan, J.; Meng, J. Difference Attention Based Error Correction LSTM Model for Time Series Prediction. *J. Phys. Conf. Ser.* **2020**, *1550*, 032121.
31. Jiang, W. Applications of deep learning in stock market prediction: Recent progress. *Expert Syst. Appl.* **2021**, *184*, 115537. https://doi.org/10.1016/j.eswa.2021.115537.
32. Gu, S.; Kelly, B.; Xiu, D. Empirical asset pricing via machine learning. *Rev. Financ. Stud.* **2020**, *33*, 2223–2273. https://doi.org/10.1093/rfs/hhaa009.
33. Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. M5 accuracy competition: Results, findings, and conclusions. *Int. J. Forecast.* **2022**, *38*, 1346–1364.
34. Wen, Q.; Zhou, T.; Zhang, C.; et al. Transformers in time series: A survey. In Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023), Macao, China, 19–25 August 2023; pp. 6750–6757.