Modern education and development
ISSN: 3060-4567

DEVELOPMENT AND COMPARATIVE ANALYSIS OF A HYBRID LSTM–XGBOOST MODEL FOR STOCK MARKET TIME SERIES FORECASTING

Karshiyev Z.A., Erkinov F.G., Sattarov M.A.

Samarkand branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Email: mirzabeks@gmail.com

***Abstract**: Accurate stock market forecasting remains a challenging problem due to the nonlinear, noisy, and volatile nature of financial time series. This paper proposes a hybrid framework combining Long Short-Term Memory (LSTM) networks and eXtreme Gradient Boosting (XGBoost) for stock market prediction. LSTM serves as a temporal feature extractor to learn latent sequential representations, while XGBoost generates final predictions from the extracted features. The framework is evaluated on daily OHLCV stock data against standalone LSTM and XGBoost models using MAE, RMSE, MAPE, and R^2. Experimental results show that the standalone LSTM achieved the best performance (MAE = 15.4081, RMSE = 18.6627, MAPE = 6.1850%, R^2 = 0.501268), while both XGBoost and the hybrid model produced substantially larger errors with negative R^2 values. The findings demonstrate that, under the current experimental setting, model hybridization does not automatically improve forecasting accuracy and highlight the importance of compatibility between deep sequential feature extraction and tree-based regression.*

***Keywords:** stock market forecasting, financial time series, LSTM, XGBoost, hybrid model, deep learning, machine learning*

1. Introduction

Stock market forecasting is a critical problem in financial research, as reliable predictions support portfolio optimization, risk management, and investment decision-making. Financial markets are influenced by multiple interacting factors —

historical prices, trading volume, macroeconomic conditions, and market sentiment — making stock price behavior highly nonlinear and difficult to model [1], [8], [11].

With advances in artificial intelligence, machine learning and deep learning approaches have become increasingly prominent in financial applications, outperforming traditional linear models by discovering complex hidden relationships in financial data [1], [11], [13]. Among deep learning methods, Long Short-Term Memory (LSTM) networks have attracted particular attention for their ability to learn long-term temporal dependencies in sequential data [5], [7], [10].

However, single predictive models often fail to capture all structural properties of financial series. LSTM effectively extracts time-dependent patterns but may suffer from overfitting and limited feature interaction modeling [5], [10]. In contrast, XGBoost excels at nonlinear regression and handling structured features but cannot directly model sequential dependence [1], [11].

Recent research has focused on hybrid models — ARIMA–LSTM, CNN–LSTM, PCA-ICA-LSTM — to exploit complementary strengths [3], [5], [7], [9]. However, the specific combination of LSTM as a temporal feature extractor with XGBoost as a final regressor remains insufficiently explored for stock forecasting.

This paper proposes a hybrid LSTM–XGBoost model where LSTM extracts latent temporal features from historical sequences, and XGBoost produces the final forecast. The study evaluates whether this two-stage structure can outperform standalone models.

The contribution is threefold: (1) a practical hybrid architecture combining sequential representation learning with boosted regression; (2) comparative evaluation against baselines using multiple metrics; (3) extended statistical and residual analysis of model performance.

The paper is organized as follows: Section 2 reviews relevant literature; Section 3 presents the methodology; Section 4 describes the experimental design; Section 5 discusses results; Section 6 concludes.

2. Literature Review

Research on stock market forecasting has evolved from traditional statistical models toward machine learning and deep learning frameworks, driven by the inability of conventional models to capture the nonlinear, non-stationary structure of financial time series [1], [11].

Early methods relied on autoregressive models, which remain useful as benchmarks but struggle under volatile conditions. Machine learning methods — support vector machines, random forest, gradient boosting, and neural networks — have since become common in financial forecasting [1], [11], [13]. A systematic review showed that regression-based and ensemble-based methods provide strong results depending on feature space and evaluation framework [11]. Feature engineering, including OHLCV variables and technical indicators, plays a central role in predictive accuracy [4], [8].

Among deep learning methods, LSTM-based architectures have become dominant in financial sequence modeling. LSTM networks preserve long-term information through memory cells and gating mechanisms, making them suitable for time-dependent prediction [5], [7], [10]. Comparative studies have shown that LSTM models effectively capture dynamic market patterns and outperform classical methods [5], [10]. CNN- and LSTM-based models have also demonstrated strong performance under walk-forward settings [7].

Several hybrid architectures have been proposed. An ensemble model combining ARMA, CNN, and LSTM improved forecasting robustness [3]. A hybrid ARIMA–LSTM model reduced error by decomposing linear and nonlinear components [5]. A PCA-ICA-LSTM framework achieved strong results for S&P 500 forecasting through dimensionality reduction before deep learning prediction [9]. Studies integrating macroeconomic, technical, and sentiment indicators demonstrated enhanced predictive capability [8], while cross-market analysis found that forecasting accuracy is associated with market efficiency [13]. Fractional optimizers for LSTM improved convergence in some contexts [2].

Despite strong support for hybrid approaches, a gap remains in the combination of LSTM and XGBoost for stock forecasting. Since LSTM excels at sequence encoding and XGBoost at structured nonlinear regression, their integration offers a strong methodological rationale. This paper addresses that gap by proposing a two-stage LSTM–XGBoost hybrid model.

3. Methodology

3.1. Research Framework

The proposed framework consists of three stages: data preparation, temporal feature extraction using LSTM, and final regression using XGBoost. Let the time series be:

$$X = \{x_1, x_2, \dots, x_T\}$$

where $x_t \in \mathbb{R}^n$ represents the n-dimensional feature vector at time t, including OHLCV values and optional technical indicators.

3.2. Data Preparation

Min–Max normalization is applied to reduce scale-related bias:

$$x_t^{norm} = \frac{x_t - x_{min}}{x_{max} - x_{min}}$$

where x_{min} and x_{max} are computed from the training set only.

3.3. Sliding Window Representation

A sliding window of size k constructs input–target pairs:

$$Z_t = \{x_{t-k}, x_{t-k+1}, \dots, x_{t-1}\}, y_t = x_t^{close}$$

The forecasting problem is learning the mapping $f(Z_t) \to y_t$.

3.4. LSTM Feature Extraction

LSTM transforms each input sequence Z_t into a latent representation through gating mechanisms:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

The final latent feature vector is the last hidden state: $F_t = h_t^{(last)}$.

3.5. XGBoost Regression

XGBoost constructs an additive ensemble of regression trees on the extracted features:

$$\hat{y}_t = \sum_{m=1}^{M} f_m (F_t)$$

with objective function:

$$L = \sum_t l(y_t, \hat{y}_t) + \sum_m \Omega(f_m)$$

where $l(\cdot)$ is squared error loss and $\Omega(f_m)$ is a regularization term.

3.6. Hybrid Model Definition

The hybrid architecture is:

$$\hat{y}_t = G(F_t) = G(LSTM(Z_t))$$

where $LSTM(Z_t)$ generates the temporal representation and $G(\cdot)$ is XGBoost.

3.7. Evaluation Metrics

Four standard metrics are used:

$$MAE = \frac{1}{N}\sum_{t=1}^{N} | y_t - \hat{y}_t |$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N} (y_t - \hat{y}_t)^2}$$

$$MAPE = \frac{100}{N}\sum_{t=1}^{N} \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

$$R^2 = 1 - \frac{\sum(y_t - \hat{y}_t)^2}{\sum(y_t - \bar{y})^2}$$

4. Model Architecture and Experimental Design

4.1. Proposed Model Architecture

The architecture follows sequential blocks: (1) OHLCV input data; (2) preprocessing and normalization; (3) sliding window transformation; (4) LSTM feature extraction; (5) XGBoost regression; (6) final prediction output.

LSTM does not directly provide the final forecast — it transforms the input into a latent representation used by XGBoost. Experiments are conducted in Python using PyTorch for the LSTM block and XGBoost library for regression.

Data are divided chronologically (70% training, 15% validation, 15% test) without random shuffling to preserve temporal structure.

LSTM configuration: hidden units: 64; layers: 2; activation: tanh; optimizer: Adam; loss: MSE; batch size: 32; epochs: 100 (with early stopping).

XGBoost configuration: estimators: 200; max depth: 5; learning rate: 0.05; subsample: 0.8; objective: squared error.

Sliding window length: $k = 20$.

4.2. Baseline Models

The hybrid model is compared with standalone LSTM and standalone XGBoost to isolate the contribution of hybridization.

5. Results and Discussion

5.1. Forecasting Results

Contrary to the initial hypothesis, the standalone LSTM model achieved the best predictive performance, while both XGBoost and LSTM–XGBoost performed substantially worse.

Table 1. Forecasting performance comparison

Model	MAE	RMSE	MAPE (%)	R^2
LSTM	15.4081	18.6627	6.1850	0.501268
XGBoost	53.3694	59.9840	21.5766	−4.152139
LSTM–XGBoost	49.7821	56.0888	20.1114	−3.504731

The LSTM model clearly outperformed across all metrics, achieving the lowest errors and the only positive R^2 value, indicating moderate explanatory ability. The negative R^2 values for XGBoost and the hybrid model indicate predictions worse than a naive mean-based baseline.

5.2. Graphical Presentation

Figure 1 presents the MAE, RMSE, MAPE, and R² comparisons. The LSTM model achieved substantially lower errors across all metrics, while the hybrid model remained closer to XGBoost than to LSTM.

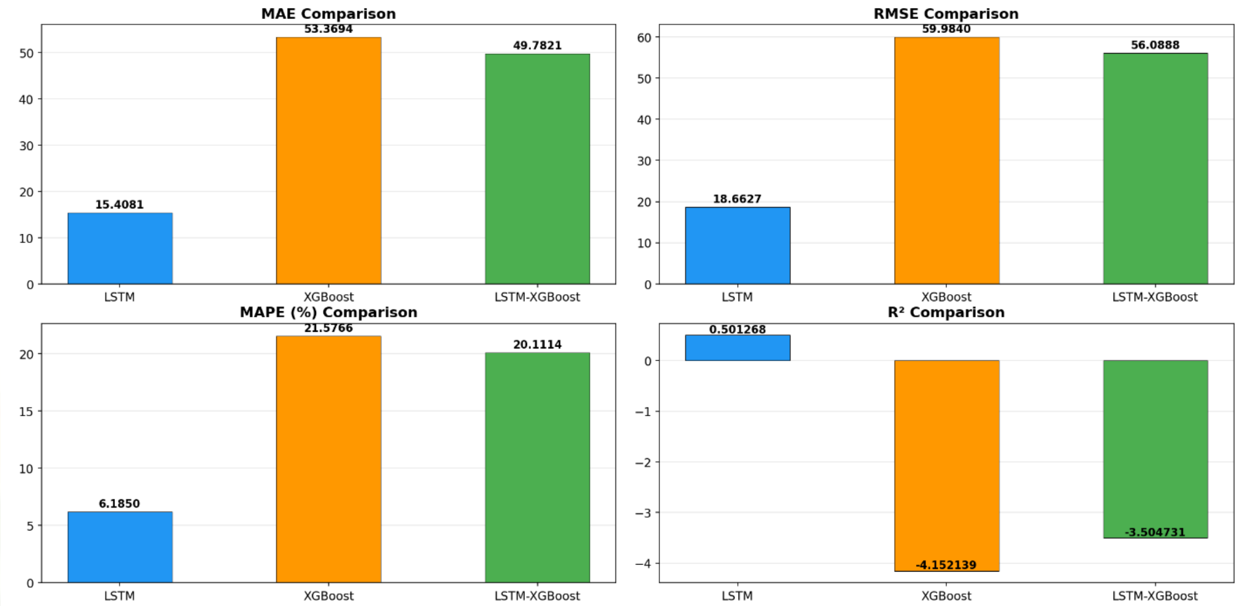

Figure 1. MAE, RMSE, MAPE and R² comparison

Figure 2 shows actual versus predicted values for the hybrid model. The predicted series remained nearly flat, failing to capture price variation.

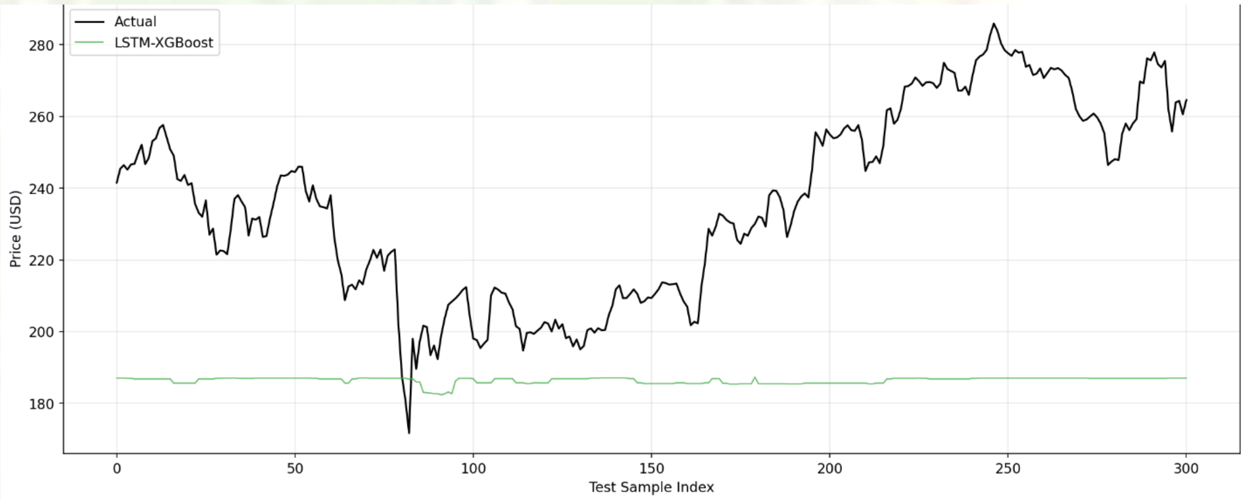

Figure 2. Real and Predicted results — Hybrid model

Figure 3 compares all models. The LSTM model followed the actual trend most closely, while XGBoost and the hybrid model produced oversmoothed predictions with limited responsiveness to fluctuations.

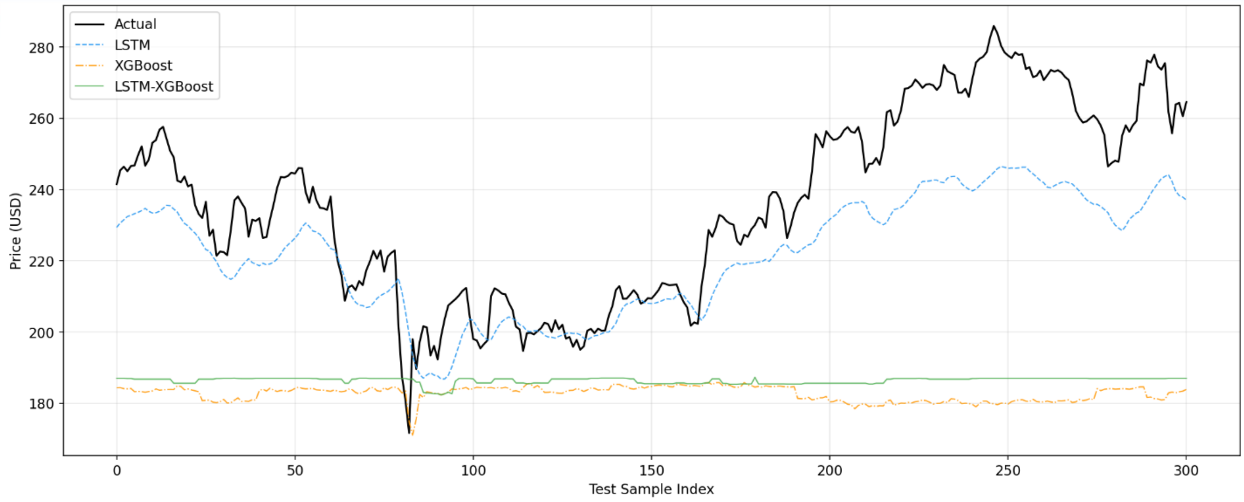

Figure 3. Real and Predicted for all models

5.3. Statistical Analysis

5.3.1. Residual Variance Analysis

The LSTM model produced more concentrated errors than XGBoost and the hybrid model, consistent with lower MAE and RMSE values. The hybrid model, although slightly better than standalone XGBoost, still exhibited high error dispersion.

5.3.2. Residual Diagnostics

Figure 4 presents residual distributions. LSTM residuals are more compact and centered, while XGBoost and hybrid residuals show much broader spreads.

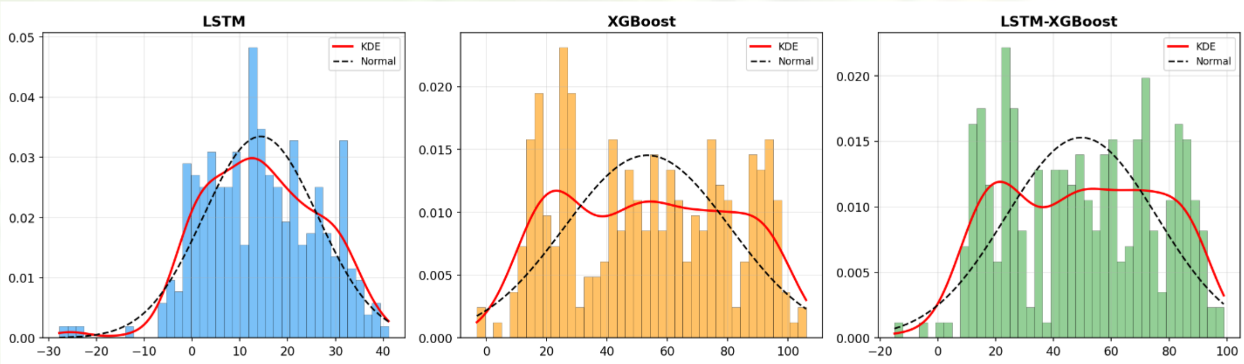

Figure 4. Residual Distributions

Figure 5 illustrates residual time series. LSTM residuals fluctuate in a narrow range, whereas XGBoost and hybrid residuals display large persistent positive deviations, suggesting systematic underestimation of actual prices.

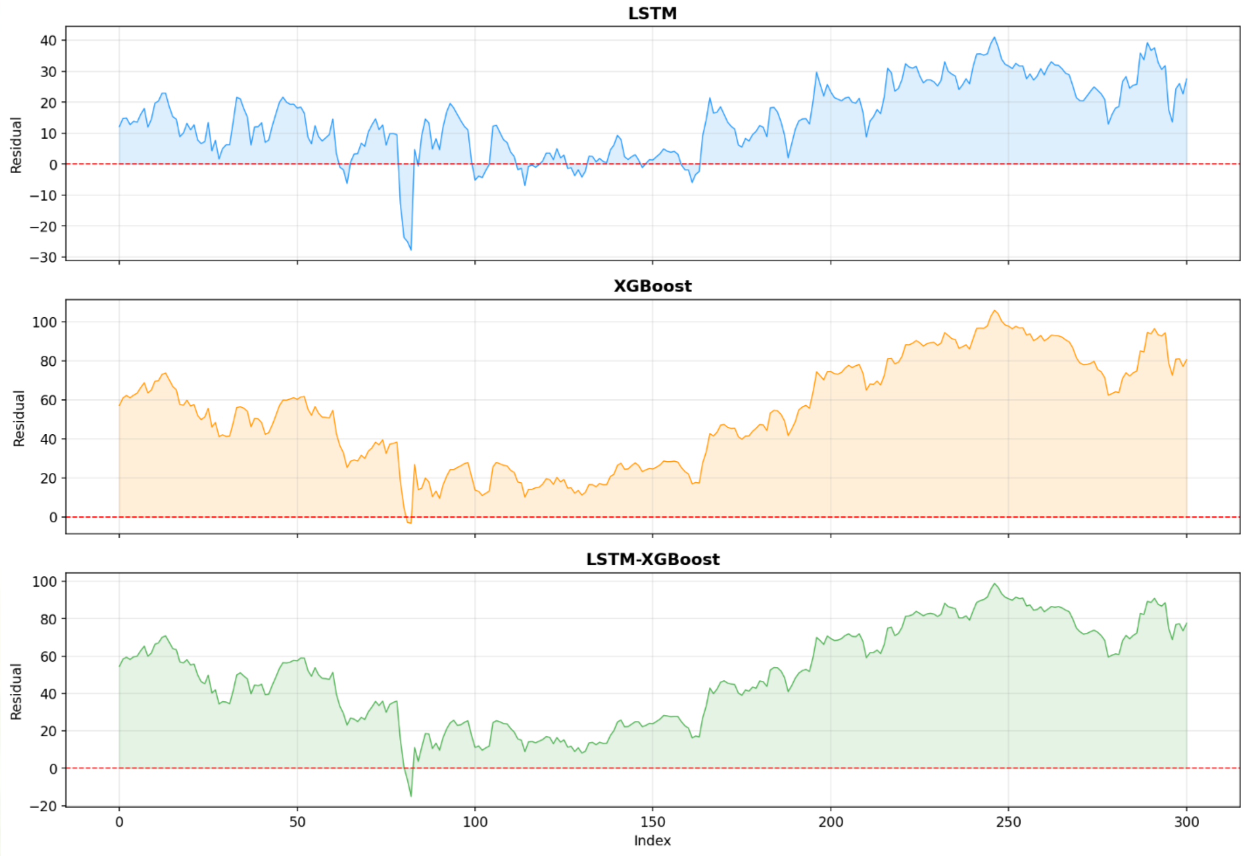

Figure 5. Residual Time Series

Figure 6 presents residual autocorrelation functions. All models exhibit strong positive autocorrelation, implying none fully captured the temporal dependence structure. This is especially problematic for XGBoost and hybrid models, whose residuals remain highly structured and non-random.

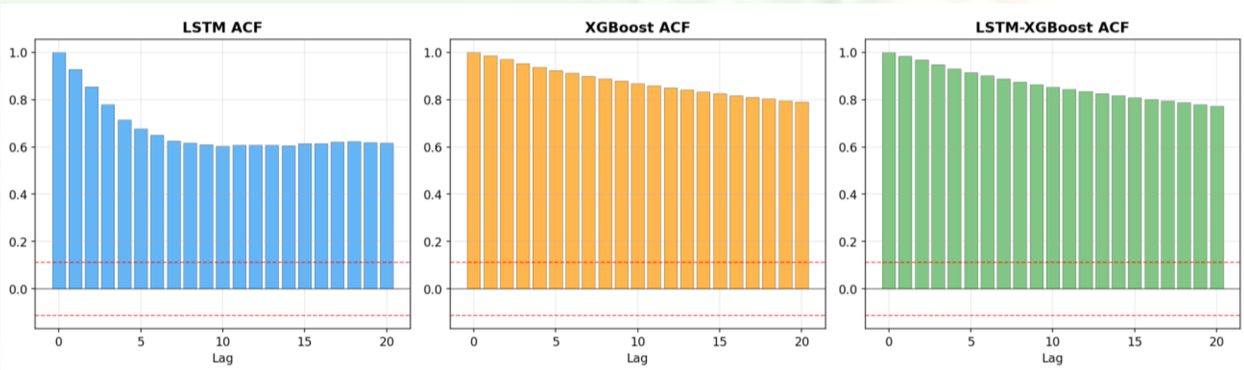

Figure 6. Residual ACF

These findings indicate that the hybrid model's second-stage XGBoost regressor did not effectively transform the latent LSTM features into accurate predictions.

5.4. Training Behavior

ISSN: 3060-4567

Figure 7 shows LSTM training and validation loss curves. Training loss decreased rapidly and stabilized, while validation loss also declined substantially. The manageable gap between curves indicates acceptable generalization, reflected in the superior test metrics.

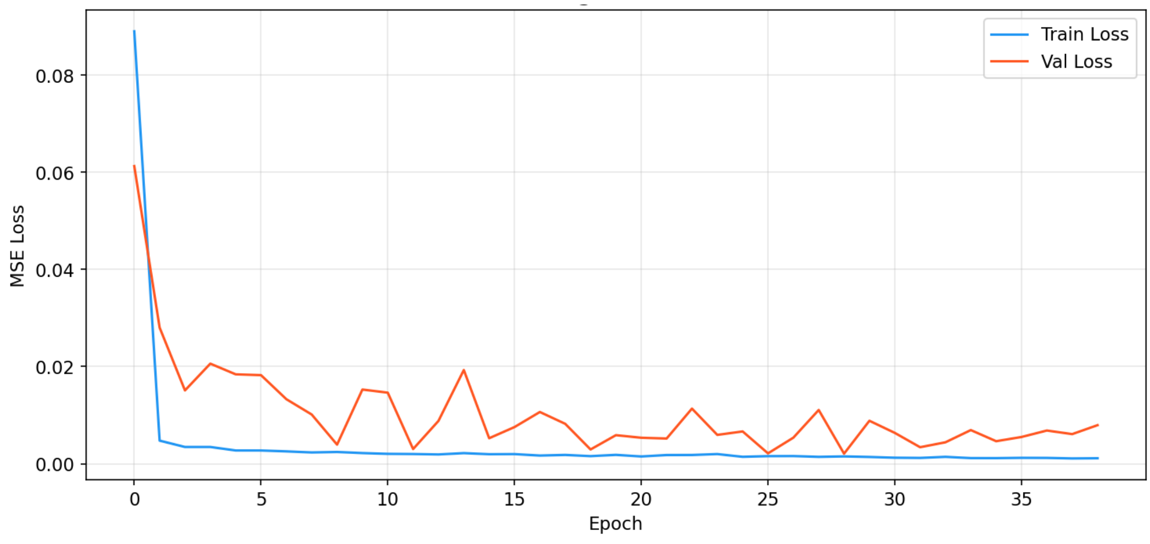

Figure 7. LSTM Training and Validation Loss

5.5. Discussion

The original hypothesis was that the hybrid LSTM–XGBoost architecture would outperform both standalone models. The results do not support this hypothesis.

Several factors may explain the hybrid model's underperformance:

First, the latent features extracted by LSTM may not have been sufficiently informative or well-structured for XGBoost. The final hidden representation does not guarantee effective exploitation by a tree-based regressor.

Second, the second-stage XGBoost may have weakened rather than improved the predictive signal, oversimplifying or distorting the LSTM representation and producing flat predictions.

Third, the selected hyperparameters and preprocessing may have favored standalone LSTM. Hybrid models are sensitive to inter-stage compatibility.

Fourth, residual diagnostics confirm the hybrid model failed to eliminate structured errors — its residuals remained autocorrelated and systematically biased, indicating incomplete modeling of the time-series structure.

Thus, although hybrid models have shown success in previous studies [3], [5], [7], [9], the present results demonstrate that hybridization does not automatically guarantee better performance.

6. Conclusion

This paper proposed a hybrid LSTM–XGBoost framework for stock market forecasting, combining temporal feature extraction through LSTM with nonlinear regression through XGBoost.

The experimental results showed that the **standalone LSTM model outperformed both the standalone XGBoost and the proposed hybrid model** across all metrics (MAE = 15.4081, RMSE = 18.6627, MAPE = 6.1850%, R^2 = 0.501268). Both XGBoost and LSTM–XGBoost produced much larger errors with negative R^2 values.

Graphical analysis confirmed these findings: LSTM followed actual dynamics closely, while the hybrid model generated nearly flat predictions. Residual analysis showed the hybrid model retained strong error structure and substantial autocorrelation.

The main conclusion is that the success of hybrid architectures depends not only on combining strong components, but also on the compatibility between them and the quality of intermediate representations.

Limitations. Experiments were conducted under a single window configuration and specific setup. The hybrid model may perform differently with other feature sets, latent extraction strategies, hyperparameters, or broader datasets. No sentiment, macroeconomic, or cross-market variables were included.

Future research may improve the framework through: different LSTM latent representations; broader hyperparameter optimization; additional technical, sentiment, and macroeconomic features; attention-based encoders or dimensionality reduction before regression; and multi-market walk-forward validation.

REFERENCES

[1] O. Bustos, A. Pomares-Quimbaya, and R. Stellian, "Machine Learning, Stock Market Forecasting, and Market Efficiency: A Comparative Study," *International Journal of Data Science and Analytics*, vol. 20, 2025.

[2] M. Ez-zaiym, Y. Senhaji, M. Rachid, K. El Moutaouakil, and V. Palade, "Fractional Optimizers for LSTM Networks in Financial Time Series Forecasting," *Mathematics*, vol. 13, 2025.

[3] K. He, Q. Yang, L. Ji, J. Pan, and Y. Zou, "Financial Time Series Forecasting with the Deep Learning Ensemble Model," *Mathematics*, vol. 11, 2023.

[4] H. H. Htun, M. Biehl, and N. Petkov, "Survey of Feature Selection and Extraction Techniques for Stock Market Prediction," *Financial Innovation*, vol. 9, 2023.

[5] D. N. Kalange, "Prediction of Stock Prices Using LSTM-ARIMA Hybrid Deep Learning Model," *Asian Journal of Probability and Statistics*, vol. 27, 2025.

[6] Q. Liu, Z. Tao, Y. Tse, and C. Wang, "Stock Market Prediction with Deep Learning: The Case of China," *Finance Research Letters*, vol. 42, 2021.

[7] S. Mehtab and J. Sen, "Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models," in *Proceedings of the International Conference on Decision Aid Sciences and Application*, 2020.

[8] M. Patsiarikas, G. Papageorgiou, and C. Tjortjis, "Using Machine Learning on Macroeconomic, Technical, and Sentiment Indicators for Stock Market Forecasting," *Information*, vol. 16, 2025.

[9] M. Sarıkoç and M. Celik, "PCA-ICA-LSTM: A Hybrid Deep Learning Model Based on Dimension Reduction Methods to Predict S&P 500 Index Price," *Computational Economics*, vol. 65, 2025.

[10] W. M. Shaban, E. Ashraf, and A. E. Slama, "SMP-DL: A Novel Stock Market Prediction Approach Based on Deep Learning for Effective Trend Forecasting," *Neural Computing and Applications*, vol. 36, 2024.

[11] G. Sonkavde, D. S. Dharrao, A. M. Bongale, S. T. Deokate, D. Doreswamy, and S. K. Bhat, "Forecasting Stock Market Prices Using Machine Learning and Deep

Learning Models: A Systematic Review," *International Journal of Financial Studies*, vol. 11, 2023.

[12] Z. Zhan and S.-K. Kim, "Versatile Time-Window Sliding Machine Learning Techniques for Stock Market Forecasting," *Artificial Intelligence Review*, vol. 57, 2024.

[13] M. Patsiarikas, G. Papageorgiou, and C. Tjortjis, "Using Machine Learning on Macroeconomic, Technical, and Sentiment Indicators for Stock Market Forecasting," *Information*, vol. 16, 2025.