

Article

Intelligent Financial Forecasting with Granger Causality and Correlation Analysis Using Bayesian Optimization and Long Short-Term Memory

Julius Olaniyan ¹, Deborah Olaniyan ^{1,*}, Ibidun Christiana Obagbuwa ², Bukohwo Michael Esiefarienrhe ³, Ayodele A. Adebisi ⁴ and Olorunfemi Paul Bernard ⁵

¹ Department of Computer Science, Bowen University, Iwo PMB 284, Osun, Nigeria;

julius.olaniyan@bowen.edu.ng

² Department of Computer Science, Sol-Plaatje University, Kimberley 8301, South Africa;

ibidun.obagbuwa@spu.ac.za

³ Department of Computer Science, North-West University, Mafikeng X2046, South Africa; 25840525@nwu.ac.za

⁴ Department of Computer Science, College of Pure and Applied Sciences, Landmark University,

Omu-Aran 251103, Kwara-State, Nigeria; ayo.adebiyi@lmu.edu.ng

⁵ Department of Computer Science, Auchipolytechnic, Auchipolytechnic 312101, Edo State, Nigeria;

bernard.femi@auchipoly.edu.ng

* Correspondence: deborah.olaniyan@bowen.edu.ng

Abstract: Financial forecasting plays a critical role in decision-making across various economic sectors, aiming to predict market dynamics and economic indicators through the analysis of historical data. This study addresses the challenges posed by traditional forecasting methods, which often struggle to capture the complexities of financial data, leading to suboptimal predictions. To overcome these limitations, this research proposes a hybrid forecasting model that integrates Bayesian optimization with Long Short-Term Memory (LSTM) networks. The primary objective is to enhance the accuracy of market trend and asset price predictions while improving the robustness of forecasts for economic indicators, which are essential for strategic positioning, risk management, and policy formulation. The methodology involves leveraging the strengths of both Bayesian optimization and LSTM networks, allowing for more effective pattern recognition and forecasting in volatile market conditions. Key contributions of this work include the development of a novel hybrid framework that demonstrates superior performance with significantly reduced forecasting errors compared to traditional methods. Experimental results highlight the model's potential to support informed decision-making amidst market uncertainty, ultimately contributing to improved market efficiency and stability.

Keywords: Bayesian; financial; smart finance; deep learning; financial forecast



Citation: Olaniyan, J.; Olaniyan, D.; Obagbuwa, I.C.; Esiefarienrhe, B.M.; Adebisi, A.A.; Bernard, O.P. Intelligent Financial Forecasting with Granger Causality and Correlation Analysis Using Bayesian Optimization and Long Short-Term Memory. *Electronics* **2024**, *13*, 4408. <https://doi.org/10.3390/electronics13224408>

Academic Editor: Domenico Ursino

Received: 8 October 2024

Revised: 7 November 2024

Accepted: 8 November 2024

Published: 11 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Financial forecasting plays an important role in modern finance, serving as a cornerstone for decision-making processes across various sectors of the economy [1]. It encompasses a diverse array of methodologies designed to anticipate future market dynamics, asset valuations, and economic indicators [2]. This predictive analysis is crucial for stakeholders ranging from individual investors to multinational corporations and government agencies, as it aids in navigating financial market complexities, mitigating risks, and optimizing investment strategies [3].

At its core, financial forecasting seeks to uncover patterns, trends, and relationships within historical data that offer insights into future market behavior [4]. One primary objective is predicting future market trends, such as changes in asset prices, market indices, and sectoral performance [5]. Accurate market trend forecasts are essential for portfolio managers and fund managers in optimizing asset allocations, diversifying risk, and achieving investment objectives within regulatory frameworks and risk tolerance levels [6].

Predicting asset prices is particularly crucial for securities trading, derivatives markets, and asset management [7]. By analyzing historical price movements, volatility patterns, and market sentiment indicators, forecasters endeavor to predict the future trajectory of individual assets, such as stocks, bonds, commodities, currencies, and cryptocurrencies [8]. These forecasts inform trading strategies, risk management decisions, and investment allocations, empowering market participants to capitalize on price fluctuations and generate favorable returns.

Economic indicator forecasting provides insights into broader macroeconomic trends and their potential impacts on financial markets. Predicting essential economic indicators, such as GDP growth, inflation rates, unemployment levels, consumer expenditure, and interest rates, enables policymakers, central banks, and businesses to anticipate changes in the business cycle, formulate monetary and fiscal policies, and make strategic decisions [9].

Traditional approaches to financial forecasting have historically relied on statistical methods, econometric models, and fundamental analytical techniques [10]. While these methodologies have provided valuable insights, they frequently encounter challenges in capturing the intricate and non-linear dynamics inherent in financial time series data, especially with high-frequency trading, market volatility, and interconnected global markets [11]. These methods involve interpreting historical price movements, trading volumes, interest rates, macroeconomic indicators, and other relevant data points to identify recurring patterns and relationships, providing guidance on potential market trends, investment opportunities, and risk exposures [12].

Recently, with the proliferation of big data, advancements in computational techniques, and the rise of artificial intelligence (AI) and machine learning (ML), there has been a paradigm shift towards more sophisticated, data-driven approaches to financial forecasting [13]. These approaches leverage algorithms and computational models to analyze large volumes of data, extract meaningful patterns and relationships, and generate forecasts with higher accuracy and efficiency.

Despite these advancements, there is a clear research gap in integrating advanced optimization techniques with deep learning models to enhance forecasting accuracy and robustness. One promising avenue is the integration of Bayesian optimization with Long Short-Term Memory (LSTM) networks. Bayesian optimization offers a systematic and efficient method for tuning the hyperparameters of complex forecasting models, enhancing their performance and generalization ability [14]. LSTM networks, specialized forms of recurrent neural networks tailored for sequential data, excel in capturing temporal dependencies and patterns, making them well-suited for time series forecasting tasks [15].

This study proposes a hybrid approach combining Bayesian optimization and LSTM networks for smart financial forecasting. This integration aims to improve the forecasting accuracy, robustness, and adaptability to changing market dynamics. By leveraging AI, ML, and advanced computational techniques, this approach seeks to develop forecasting models that can effectively navigate financial market complexities, provide actionable insights, and support decision-making processes amidst uncertainty and volatility. As the field of smart financial forecasting continues to evolve, driven by advances in technology, data availability, and computational capabilities, innovative approaches like the one proposed in this study are crucial. They can deliver timely and reliable forecasts, empower market participants with actionable insights, and ultimately contribute to the efficiency and stability of financial markets.

This study employed a structured methodology to enhance financial forecasting through the integration of Bayesian optimization and Long Short-Term Memory (LSTM) networks. It began with a comprehensive literature review to identify challenges in existing forecasting methods, establishing the context for the proposed approach. Data were then collected from various financial markets and preprocessed for analysis. The hybrid model was implemented to forecast market trends and economic indicators, followed by a rigorous evaluation against traditional methods to assess its performance. Finally, the results were

analyzed to draw meaningful conclusions, highlighting the model's potential to improve decision-making amid market volatility.

2. Theoretical Background

The increasing integration of machine learning techniques into financial forecasting is driven by their capacity to manage complex, nonlinear relationships within data. Supervised learning algorithms, such as decision trees and random forests, have become prominent tools for predicting stock prices due to their effectiveness in identifying historical patterns. For instance, Tang et al. [16] have demonstrated the efficacy of these algorithms in stock price prediction. Similarly, Chen et al. [17] tackled the issue of fraudulent financial statements by developing predictive models using decision trees, logistic regression, and support vector machines (SVMs). Their study revealed that the decision tree model (DT C5.0) achieved the highest classification accuracy, surpassing logistic regression and SVMs in identifying fraudulent statements. This success underscores the advantages of ensemble methods in reducing overfitting and enhancing generalization, thereby increasing their value for financial analysis.

Deep learning models, particularly Long Short-Term Memory (LSTM) networks, have emerged as pivotal in financial forecasting due to their ability to capture long-term dependencies in sequential data. Van et al. [18] emphasized LSTM networks' capability to retain and learn from extended sequences of past information, which is essential for analyzing financial time series data. Research by Malsa et al. [19] showcased LSTM networks' superior performance in forecasting cryptocurrency prices compared to traditional methods. Wu et al. [20] highlighted LSTM networks' adaptability to volatile cryptocurrency markets, while Siami-Namini and Namin [21] demonstrated that LSTM networks significantly outperform traditional ARIMA models, leading to reduced error rates. Zhang et al. [22] advanced this further with the introduction of an attention-based LSTM (AT-LSTM) model, which improves long-term dependency handling over standard LSTM models. Dixon and London [23] also explored exponential smoothed recurrent neural networks (α -RNNs), finding them effective in big data financial applications with fewer parameters than complex architectures like GRUs and LSTM networks.

In addition, research on deep learning in stock market prediction and financial management highlights the significant improvements offered by advanced neural network models. Zhao et al. [24] integrated CNN, BiLSTM, and attention mechanisms to capture complex patterns in financial data, outperforming traditional models with lower error metrics, such as MAE and RMSE. Staffini [25] demonstrated the advantages of DCGANs in predicting stock prices, showing improvements over conventional methods like ARIMA and LSTM networks. Zheng et al. [26] further explored advanced machine learning models, such as LSTM networks and CNN-BiLSTM, which excel in capturing nonlinear market patterns, achieving a Mean Squared Error (MSE) of 0.01286 and a Mean Absolute Percentage Error (MAPE) of 0.01984. Ghosh et al. [27] assessed the predictive power of MLP, RNN, LSTM, and CNN models, revealing that CNN outperformed the others in predicting stock prices from NSE and NYSE data, with lower MAPE values across companies. Sharma et al. [28] evaluated the ARMA model's effectiveness in forecasting stock prices for leading technology companies, achieving MAPE values between 1.10% and 1.50%, indicating its reliability. Ahammad et al. [29] compared LSTM and ARIMA models, with ARIMA unexpectedly showing superior accuracy, highlighting the need for continued refinement in stock market prediction methodologies. Md et al. [30] explored this challenge by introducing a Multi-Layer Sequential Long Short-Term Memory (MLS LSTM) model, which leverages the Adam optimizer and normalized time series data to predict stock price indices accurately. This model achieved notable prediction accuracies of 95.9% on training data and 98.1% on testing data, outperforming conventional machine learning models and demonstrating feasibility for real-world applications with a low Mean Absolute Percentage Error (MAPE) and Normalized Root Mean Squared Error (NRMSE). Zhang et al. [31] took a different approach by enhancing prediction models through a hybrid architecture com-

binning Convolutional Neural Networks (CNN), Bi-directional Long Short-Term Memory (BiLSTM), and an attention mechanism. This model aimed to improve the prediction accuracy by efficiently extracting temporal features and adjusting weight assignments for input data through the attention mechanism. Chauhan et al. [32] addressed another key aspect of stock price prediction: mitigating noise and overfitting in deep learning models. They proposed an ensemble of LSTM and GRU models optimized with Particle Swarm Optimization (PSO), focusing on price movement prediction for the Nifty 50 index. This ensemble, which combined PSO-hyperparameter-tuned LSTM with GRU models, achieved an accuracy of 57.72% and enhanced precision to 0.5485, surpassing standalone models. Their findings highlighted the potential of integrating metaheuristic algorithms with deep learning ensembles to improve performance in noisy and complex stock data environments.

To illustrate the development and current landscape of financial forecasting methodologies, a chronological literature review table is included in this section. Table 1 lists key studies by publication date, showcasing advancements, trends, and research gaps over time. It provides context for the proposed Bayesian-LSTM model, highlighting its novelty and contributions within the broader research discourse.

Table 1. Chronological overview of financial forecasting models and techniques.

Authors	Study Focus	Key Contributions	Relevance to Current Study
Tang et al. [16]	Demonstrated the efficacy of machine learning algorithms in stock price prediction	Showed that machine learning models can effectively predict stock prices, validating their application in financial forecasting.	Supports the use of machine learning models for enhancing the stock price prediction accuracy in the current study.
Chen et al. [17]	Fraudulent financial statement detection using decision trees, logistic regression, and SVMs	Demonstrated that decision tree models (DT C5.0) achieve a higher classification accuracy compared to logistic regression and SVMs.	Highlights the early success of ensemble methods, providing a foundation for the application of advanced models in financial forecasting.
Van et al. [18]	LSTM networks in financial time series analysis	Emphasized LSTM networks' capacity to learn from extended sequences, which is critical for analyzing financial data.	Reinforced the importance of long-term memory in financial predictions, which is essential for modeling market trends and cycles.
Malsa et al. [19]	LSTM networks' performance in cryptocurrency price forecasting	LSTM networks outperformed traditional forecasting methods.	Demonstrated the practical superiority of LSTM networks, leading to their broader application in financial forecasting.
Wu et al. [20]	LSTM networks in cryptocurrency market forecasting	Demonstrated LSTM networks' adaptability to volatile markets like cryptocurrency.	Showcased the flexibility of LSTM networks in handling high volatility, contributing to their widespread adoption in financial forecasting.
Siarni-Namini and Namin [21]	LSTM vs. traditional ARIMA models for financial forecasting	Showed that LSTM models significantly reduce error rates compared to ARIMA, highlighting the advantages of deep learning models.	Validates the shift from traditional statistical models to deep learning approaches, setting the stage for more complex architectures.
Zhang et al. [22]	Introduction of Attention-based LSTM (AT-LSTM)	Enhanced long-term dependency handling, improving prediction accuracy over standard LSTM models.	Introduced a more sophisticated LSTM variant that directly influences the architecture of modern financial forecasting models.
Dixon and London [23]	Exponential smoothed RNNs (α -RNNs) in big data financial applications	Found α -RNNs effective with fewer parameters compared to GRUs and LSTMs.	Highlighted parameter efficiency, informing the design of more streamlined and scalable deep learning models.

Table 1. Cont.

Authors	Study Focus	Key Contributions	Relevance to Current Study
Zhao et al. [24]	Integrating CNN, BiLSTM, and attention mechanisms for financial time series forecasting	Achieved lower error metrics (MAE and RMSE) compared to traditional models.	Illustrates the progression to more integrated and complex model architectures, reflecting current trends in financial forecasting.
Staffini [25]	DCGANs for stock price predictions	Showed significant improvements over ARIMA and LSTM networks, indicating the potential of generative models in financial forecasting.	Highlights the emergence of generative models, expanding the toolkit for financial predictions beyond conventional methods.
Zheng et al. [26]	Explored advanced machine learning models (LSTM and CNN-BiLSTM) for stock price predictions	Demonstrated the effectiveness of the LSTM and CNN-BiLSTM models in capturing nonlinear market patterns, achieving a low Mean Squared Error (MSE) of 0.01286 and MAPE of 0.01984.	Highlights the potential of combining deep learning architectures to improve the accuracy of market trend predictions.
Ghosh et al. [27]	Assessed the predictive performance of MLP, RNN, LSTM, and CNN models on stock price data from NSE and NYSE	Found that CNN models outperformed the others, achieving lower MAPE values across different companies and indicating superior predictive power for stock prices.	Provides insights into selecting CNN-based models for financial forecasting, supporting the use of effective architectures in experiments in the current study.
Sharma et al. [28]	ARMA model's effectiveness in technology stock price forecasting	MAPE values were between 1.10% and 1.50%, indicating reliable performance.	Provides a benchmark for evaluating newer models like Bayesian-LSTM against traditional approaches in financial forecasting.

3. Materials and Methods

3.1. Materials

This section introduces an innovative methodology aimed at transforming financial forecasting in dynamic markets. The approach integrates advanced techniques such as Bayesian optimization and Long Short-Term Memory (LSTM) networks to create a sophisticated predictive model. By combining the power of Bayesian optimization with the memory capabilities of LSTM networks, the method enhances both the accuracy and robustness of financial predictions. This fusion of techniques represents a significant step forward in revolutionizing financial forecasting, offering improved insights and performance in highly volatile markets.

3.1.1. System Architecture

The architectural design of the financial forecasting model, harnessing the synergy between Bayesian optimization and Long Short-Term Memory (LSTM) networks, as illustrated in Figure 1, is presented in this segment.

3.1.2. Dataset

The dataset utilized in this study is an enhanced version of the Chinese stock market dataset publicly accessible on the Kaggle repository. It encompasses a wide array of financial metrics that go beyond the traditional Open, High, Low, Close (OHLC) prices, and volume data. Additionally, it includes various essential financial ratios computed on a daily basis, such as the Price-to-Earnings (PE) ratio, Price-to-Book (PB) ratio, Price-to-Sales (PS) ratio, and dividend yield, among others, as depicted in Table 2. This enriched dataset offers a comprehensive view of the Chinese stock market landscape, enabling a more thorough analysis of market dynamics and investment opportunities. The dataset covers a significant time span, starting from 4 January 2005, and extending up to 11 May 2022, allowing for a comprehensive examination of long-term trends and patterns within the market. Currently

available in CSV (Comma-Separated Values) format, it comprises 1,048,576 records with 18 features.

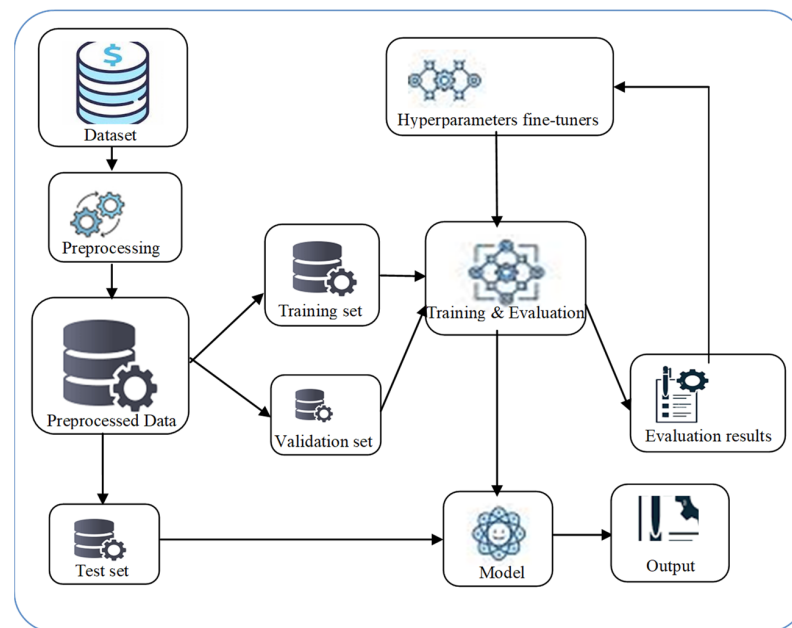


Figure 1. The proposed system architecture. The architecture combines Bayesian optimization and LSTM networks to enhance the accuracy and robustness of financial predictions, offering improved insights and performance in volatile markets.

Table 2. Chinese stock market dataset sample.

Date	Ticker	Open	High	Low	Close	Volume	Outstanding_Share
04/01/2005	sh600000	0.77	0.77	0.75	0.76	3,808,939	900,000,000
05/01/2005	sh600000	0.76	0.76	0.74	0.75	5,225,244	900,000,000
06/01/2005	sh600000	0.75	0.75	0.73	0.74	4,298,099	900,000,000
07/01/2005	sh600000	0.74	0.75	0.73	0.74	4,362,864	900,000,000
10/01/2005	sh600000	0.75	0.77	0.74	0.77	7,115,260	900,000,000
11/01/2005	sh600000	0.77	0.78	0.76	0.77	6,895,175	900,000,000
12/01/2005	sh600000	0.78	0.78	0.76	0.77	2,797,449	900,000,000
13/01/2005	sh600000	0.77	0.78	0.77	0.77	1,284,827	900,000,000
14/01/2005	sh600000	0.77	0.78	0.76	0.76	1,731,048	900,000,000
17/01/2005	sh600000	0.76	0.76	0.75	0.75	2,620,773	900,000,000

3.1.3. Data Preprocessing

The data preprocessing stage consists of several critical steps to prepare the dataset for modeling and analysis. Initially, the dataset is loaded into a pandas Data Frame for easy manipulation, with missing values handled using the forward fill method to ensure data continuity. The ‘Date’ column is converted to a date–time format to facilitate the temporal analysis, while additional features are engineered, such as calculating daily price changes and percentage movements based on the ‘Open’ and ‘Close’ prices. To ensure that features are on a comparable scale, Min–Max scaling is applied, standardizing the data to avoid certain features dominating others during the model’s training process. Following this, the dataset is split into training and testing sets, with the ‘Close’ price designated as the target variable (endogenous variable) and the remaining columns serving as exogenous features. These exogenous variables include Open, High, Low, and Volume.

As illustrated in Figure 2, a correlation matrix was computed to examine the relationships between all variables, particularly to address potential multicollinearity among highly correlated features like the OHLC prices. This analysis confirmed a high degree of

correlation among the Open, High, Low, and Close prices. However, these features were retained due to their direct impacts on stock price movement and their predictive relevance in time series forecasting. To further refine the feature set, the SelectKBest method with $f_{\text{regression}}$ scoring was employed, allowing the selection of the most relevant features for the prediction task. Despite the correlations among some features, those selected, including Open, High, Low, and Volume, were kept based on their overall contributions to capturing market dynamics.

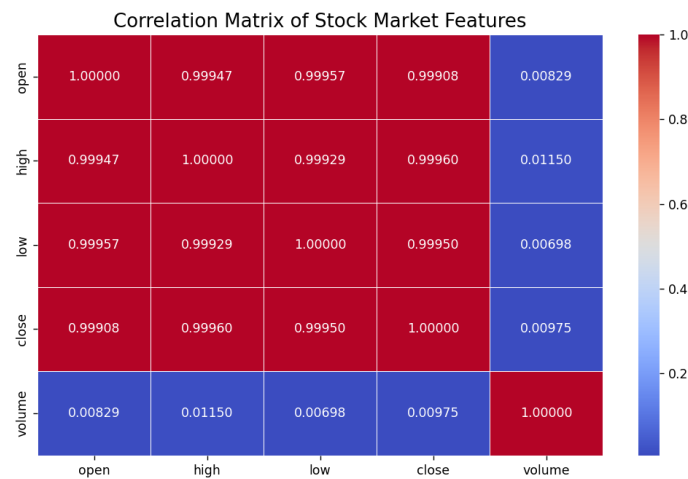


Figure 2. Correlation matrix. Granger causality analysis showing significant predictive relationships among OHLC prices and Volume, highlighting the dynamic interactions and justifying the inclusion of these variables in the model.

Also, a Granger causality analysis was conducted to investigate the predictive relationships among these variables. This method assessed whether past values of one variable could provide significant information about future values of another, thereby enhancing the understanding of the dynamic interactions between the OHLC prices and Volume. As illustrated in Figure 3, the results of the Granger causality test revealed significant predictive relationships, further justifying the inclusion of Open, High, Low, and Volume in the model, despite their correlations.

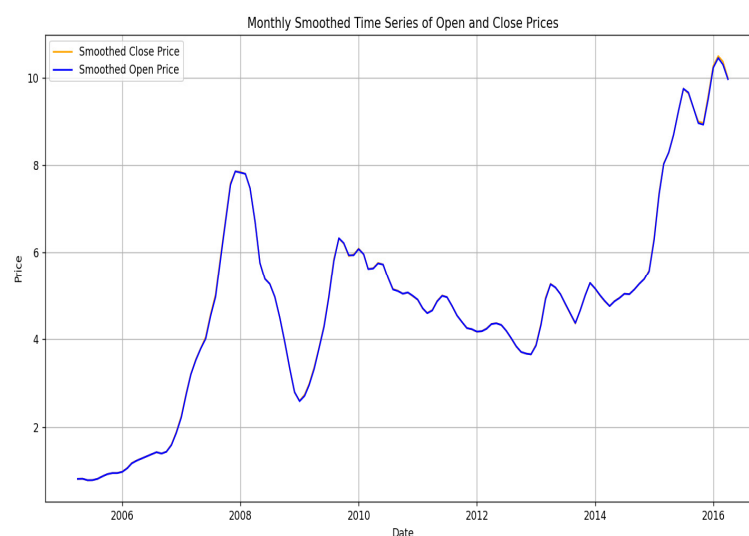


Figure 3. Granger causality analysis. The Augmented Dickey–Fuller (ADF) test indicates that the *Close* price series is non-stationary, with a high p -value of 0.6557, suggesting varying statistical properties over time. In contrast, the *Volume* series is stationary, with a very low p -value of 0.0006, confirming consistent statistical behavior. Table 3 provides detailed results.

Table 3. Granger causality analysis report.

Variable	Statistics	<i>p</i> -Value	Critical Values
			1%
‘Close’	−1.2411	0.6557	−3.4329
‘Volume’	−4.2250	0.0006	−3.4329

The results from the stationarity analysis of the ‘Close’ and ‘Volume’ time series using the Augmented Dickey–Fuller (ADF) test, as shown in Table 4, provide key insights into the behaviors of these variables. For the ‘Close’ price, the ADF test statistic is −1.2411, with a corresponding *p*-value of 0.6557. Given that the *p*-value is well above common significance thresholds (such as 0.05), we fail to reject the null hypothesis of non-stationarity. Additionally, the ADF test statistic is greater than the critical values at the 1%, 5%, and 10% levels, further supporting the conclusion that the ‘Close’ price series is non-stationary. This implies that the ‘Close’ price exhibits changing statistical properties over time, likely influenced by trends or other time-varying factors. Conversely, the ‘Volume’ series shows a markedly different result. The ADF test statistic for ‘Volume’ is −4.2250, with a *p*-value of 0.0006. This very low *p*-value allows us to reject the null hypothesis of non-stationarity, and the test statistic is significantly lower than the critical values at all significance levels, confirming that the ‘Volume’ series is stationary. In summary, while the ‘Volume’ series exhibits consistent statistical properties over time, the ‘Close’ price series is non-stationary. This non-stationarity in the ‘Close’ price suggests that further transformations, such as differencing, may be required before proceeding with advanced time series analysis, such as Granger causality tests. Table 3 highlights these findings in detail.

Table 4. Selected features.

Date	Open	High	Low	Close	Volume
04/01/2005	0.77	0.77	0.75	0.76	3,808,939
05/01/2005	0.76	0.76	0.74	0.75	5,225,244
06/01/2005	0.75	0.75	0.73	0.74	4,298,099
07/01/2005	0.74	0.75	0.73	0.74	4,362,864
10/01/2005	0.75	0.77	0.74	0.77	7,115,260
11/01/2005	0.77	0.78	0.76	0.77	6,895,175
12/01/2005	0.78	0.78	0.76	0.77	2,797,449
13/01/2005	0.77	0.78	0.77	0.77	1,284,827
14/01/2005	0.77	0.78	0.76	0.76	1,731,048
17/01/2005	0.76	0.76	0.75	0.75	2,620,773

Algorithm 1 provides a comprehensive overview of the preprocessing operations conducted.

Algorithm 1: Data Preprocessing

Load the dataset:

Input: Path to the stock data file (china_stock_data.csv)

Output: data

Check for missing values:

If missing values are found, fill them using the forward fill method.

Convert the ‘Date’ column to a date–time format.

Calculate daily price changes:

Compute the difference between the ‘Close’ and ‘Open’ prices and store as ‘Price Change’.

Calculate the daily percentage change: $\text{Price Change \%} = \frac{\text{Price Change}}{\text{Open}} \times 100$

Algorithm 1: *Cont.*

Normalize feature values:

Use Min–Max Scaling to transform the following columns: ‘Open’, ‘High’, ‘Low’, ‘Close’, ‘Volume’.

Prepare training and test datasets:

Define features (X) by excluding the ‘Date’ and ‘Close’ columns.

Define the target variable (y) as the ‘Close’ price.

Split the data into training and test sets (X_train, X_test, y_train, y_test) using an 80–20 ratio.

Feature selection:

Apply SelectKBest to choose the top 5 features based on their f_regression scores from the training data.

Output: X_train_selected (transformed training features)

Following the completion of preprocessing operations, the dataset experienced transformation into an appropriate format for subsequent modeling and analysis, as depicted in Table 4. This facilitated the development of precise predictive models for financial forecasting in dynamic markets, as proposed in this research work.

3.2. Method

3.2.1. Training and Fine Tuning

This section details the training and fine-tuning process of the predictive models using the preprocessed dataset. Bayesian optimization was utilized to finetune the hyperparameters of the Long Short-Term Memory (LSTM) model, aiming to optimize its performance. Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared (R^2), Mean Absolute Percentage Error (MAPE), and Residual Analysis metrics served as the primary performance metrics to assess the effectiveness of the trained models.

3.2.2. Model Training with BO

The training process initiated with the application of Bayesian optimization to finetune the Long Short-Term Memory (LSTM) model. Bayesian optimization utilizes adaptive exploration mechanisms to systematically explore the hyperparameter space and optimize the configuration of the LSTM model. Through iterative adjustments of hyperparameters such as the number of layers, units, learning rate, and dropout rate, Bayesian optimization aimed to enhance the predictive accuracy and robustness of the LSTM model for financial forecasting tasks. During the fine-tuning process, Bayesian optimization systematically traversed the hyperparameter space to pinpoint the optimal combination of hyperparameters that minimized the discrepancy between the predicted and actual values. Hyperparameters were iteratively modified based on their impacts on performance metrics, with the goal of improving model accuracy and generalization. To assess the efficiency of the Bayesian optimization process, the total time taken for the optimization was recorded. This logging helps in understanding the computational demands of hyperparameter tuning, which is crucial for practical implementation in real-world scenarios. The process took approximately 8400 s, indicating the time required to achieve optimal hyperparameter settings for the LSTM model.

3.2.3. Bayesian Optimization

- **Objective Function:** Let $f(x)$ represent the objective function that encapsulates the performance metric of the Long Short-Term Memory (LSTM) model given the hyperparameter configuration x . This function serves as the basis for guiding the optimization process, where the aim is to find the hyperparameter values that maximize model performance.
- **Surrogate Model (Gaussian Process):** The surrogate model is a fundamental component of Bayesian optimization, allowing for the approximation of the objective

function. In this research, a **Gaussian Process (GP)** was used as the surrogate model, which can be mathematically represented as

$$\tilde{f} \sim \text{GP}(m(x), k(x, x')) \quad (1)$$

where $m(x)$ is the **mean function**, which provides a prior belief about the expected value of the objective function at any given point x . In many cases, this mean is assumed to be zero, reflecting no prior bias toward the function values. Additionally, $k(x, x')$ is the **kernel function**, which defines the covariance structure of the Gaussian Process. This function measures the similarity between different input points x and x' , allowing the model to understand the relationship between them. The choice of kernel function significantly influences the behavior and performance of the Gaussian Process, with popular choices including the Radial Basis Function (RBF) kernel and the Matérn kernel.

The Gaussian Process provides a distribution over functions rather than a single deterministic function. This probabilistic approach was employed to express uncertainty about the objective function, enabling effective exploration and exploitation during the optimization process.

- **Acquisition Function:** To decide where to sample next, we employ an **acquisition function**, specifically the **Expected Improvement (EI)**, which is defined as

$$\alpha(x) = E[\max(0, f(x) - f_{\min})] \quad (2)$$

Here, $f_{\min}$ denotes the minimum observed value of the objective function thus far. The Expected Improvement function quantifies the expected gain in performance by evaluating the objective function at a new point x . By maximizing this acquisition function, we focus our search on areas that are likely to yield improvements over the current best-known performance.

- **Next Hyperparameters:** The next set of hyperparameters to explore is determined by maximizing the acquisition function.

$$x_{\text{next}} = \arg\max_x \alpha(x) \quad (3)$$

This step strategically selects hyperparameters that balance exploration (searching unexplored regions) and exploitation (refining the search around known good configurations).

- **Update Surrogate Model:** After evaluating the objective function at the newly selected hyperparameter configuration $f(x_{\text{next}})$, the surrogate model is updated to reflect this new information. The updated surrogate model can be expressed as

$$\tilde{f}_{\text{new}}(x) = \tilde{f}_{\text{old}}(x) + \text{GP}_{\text{update}}((x_{\text{next}}, f(x_{\text{next}}))) \quad (4)$$

This updating mechanism ensures that the surrogate model improves its approximation of the objective function over successive iterations, leading to more informed hyperparameter selections.

3.2.4. LSTM Model

1. **LSTM Output:** Let $\text{LSTM}(x)$ be the output of the LSTM model with the hyperparameter denoted by x .
2. **Training loss:** Let the training loss, $L(x)$, be the objective to minimize during training.
3. **Hyperparameters:** Let x represent a set of hyperparameters for the LSTM model.

3.2.5. Fusion of Bayesian Optimization and LSTM

1. **Optimization Process:**
 - i. Integrate Bayesian optimization with the LSTM training process.

- ii. At each iteration of Bayesian optimization, select x_{next} using the Expected Improvement acquisition function.
 - iii. Train the LSTM model with hyperparameters x_{next} .
2. Overall Objective:
Minimize the final performance metric of the LSTM model by tuning hyperparameters through Bayesian optimization

$$\min_x P(f(x)) = \min_x P(\text{LSTM}(x)) \quad (5)$$

3. Update Process:
Update the surrogate model and LSTM model iteratively based on observed performance, adjusting the mean and variance of the GP predictive distribution.

Algorithm 2 presents a structured outline of the algorithmic approach used for model training with Bayesian optimization.

Algorithm 2: Model Training with Bayesian Optimization

1. Load the dataset from the CSV file.
 2. Preprocess the dataset:
 - a. Drop missing values;
 - b. Convert the 'Date' column to date-time format;
 - c. Set 'Date' as the index;
 - d. Extract 'Close' prices.
 3. Normalize the dataset using MinMaxScaler.
 4. Create sequences and labels:
 - a. Define the sequence length;
 - b. For each time step in the dataset:
 - i. Append sequence of 'sequence_length' to sequences list;
 - ii. Append the next value to the label list.
 5. Convert sequences and labels to numpy arrays.
 6. Split the data into training and testing sets:
 - a. Define the split ratio (e.g., 80–20);
 - b. Split sequences and labels based on the split index.
 7. Define the hyperparameter search space:
 - a. Set ranges for LSTM units, optimizers, batch sizes, epochs, learning rates, and dropout rates.
 8. Define the objective function:
 - a. Initialize an empty Sequential model;
 - b. Add LSTM layers with specified units and dropout;
 - c. Compile the model with selected optimizer and loss function;
 - d. Fit the model on training data with validation data and early stopping;
 - e. Predict on test data and calculate the Mean Squared Error (MSE).
 9. Perform Bayesian optimization:
 - a. Initialize Trials object;
 - b. Run the optimization to find the best hyperparameters using the TPE algorithm.
 10. Retrieve the best hyperparameters.
 11. Build the final model with the best hyperparameters:
 - a. Initialize a new Sequential model;
 - b. Add LSTM layers with the best units and dropout;
 - c. Compile the model with best optimizer and loss function.
 12. Train the final model on the training data with validation data and early stopping.
 13. Evaluate the final model on the test data:
 - a. Calculate residuals;
 - b. Calculate performance metrics (MSE, MAE, RMSE, R^2 , and MAPE).
 14. Plot results:
 - a. Plot residuals;
 - b. Plot training and validation accuracy;
 - c. Plot training and validation loss.
-

3.2.6. Performance Evaluation

In evaluating the performance of the trained models, we utilized several key metrics, including Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), Mean Absolute Percentage Error (MAPE), and Residual Analysis. Among these, MAE was selected as the primary metric due to its straightforward nature and ease of interpretation, offering a clear measure of predictive accuracy without being disproportionately influenced by outliers. While RMSE is more sensitive to large errors, which can skew the results, MAE provides a more balanced view, making it particularly suitable for our analysis. This approach allows for a more accurate comparison of model performance across different scenarios. The equations for the evaluation metrics are presented as follows:

$$\text{MAE} = \frac{\sum |y_i - \hat{y}_i|}{n} \quad (6)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (8)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (9)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (10)$$

$$\text{ResidualAnalysis} = y_i - \hat{y}_i \quad (11)$$

The hyperparameter space defined for Bayesian optimization examined a learning rate within a uniform range of 1×10^{-4} to 1×10^{-2} , while the drop-out rate was adjusted between 0.0 and 0.5. The batch size was chosen from three options: 32, 64, and 128. The number of epochs was considered at either 10, 20, or 30. For the model architecture, the first LSTM layer ('units_lstm1') had options of 32, 64, or 128 units, and the second LSTM layer ('units_lstm2') had options of 64, 128, or 256 units. Throughout the Bayesian optimization process, these hyperparameters were initially set within the defined ranges. Ultimately, the optimization process determined that the best-performing hyperparameters included a batch size of 64, a drop-out rate of 0.12896, 30 epochs, and a learning rate of 0.00976. The Adam optimizer was selected, with the first LSTM layer configured with 64 units and the second LSTM layer with 256 units. These settings were found to deliver the optimal performance for the model.

The training and fine-tuning processes were carried out in Python, utilizing TensorFlow and Keras for LSTM model development. Bayesian optimization for hyperparameter tuning was integrated through the Hyperopt library. Model performance was evaluated using several metrics, including RMSE, MAE, R^2 , MAPE, MSE, and Residual Analysis, all computed with built-in implementations from the scikit-learn library.

4. Results

The results of an experimental study aimed at revolutionizing financial predictions in dynamic markets through the combination of Bayesian optimization and Long Short-Term Memory (LSTM) models utilizing historical Chinese stock market data for experimentation and findings are hereby presented in detail.

4.1. Performance of the Predictive Model

In this section, the performance of the predictive model was analyzed using various evaluation metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R^2), and Mean Absolute Percentage Error (MAPE). To ensure a robust evaluation, the dataset was split into training and testing sets

using an 80–20 split, where 80% of the data was utilized for training the model and the remaining 20% for testing. This division allows the assessment of the model's performance on unseen data, addressing potential overfitting concerns. The Mean Squared Error (MSE) achieved a value of 0.00000535, indicating the average squared difference between the actual and predicted values. This low MSE suggests a high level of precision in predictions, with minimal variance. The Mean Absolute Error (MAE) was measured at 0.00102352, representing the average magnitude of errors in predictions. The small MAE value indicates that the model's predictions are, on average, very close to the actual values. The Root Mean Squared Error (RMSE) was calculated as 0.00231278, providing an accurate measure of the differences between predicted and observed values. RMSE helps gauge the spread of predictions around the actual values. The R-squared (R^2) value of 0.99752901 signifies the proportion of the variance in the dependent variable that can be predicted from the independent variables. This high R^2 value suggests that the model explains a significant portion of the variability in the data. The Mean Absolute Percentage Error (MAPE) stood at 0.02930441, indicating the average percentage difference between the predicted and actual values. This low MAPE suggests that the model's predictions are, on average, very accurate relative to the actual values. Collectively, these performance evaluation metrics demonstrate the effectiveness of the predictive model in accurately capturing and forecasting market trends. The consistently low values across these metrics affirm the reliability and precision of the model's predictions, providing confidence in its ability to inform financial decision-making processes.

4.2. Impact and Performance of the Fusion Algorithm

The performance and impact of the hybrid model (improved Bayesian optimization and LSTM networks) was analyzed. By leveraging the inherent strengths of the model, such as its ability to effectively explore complex search spaces and adapt to evolving market dynamics, the proposed model exhibited enhanced convergence rates and minimized prediction errors. This shows the importance of incorporating nature-inspired optimization techniques with deep learning models in financial forecasting domains. Figure 4 visually illustrates the performance of the hybrid predictive model by presenting a comparison between the actual stock prices and the predicted prices. The visualization demonstrates how the model effectively captures the underlying patterns in the data and generates accurate predictions.

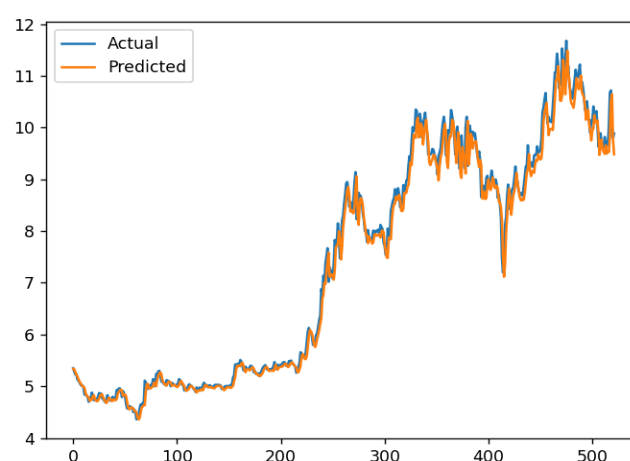


Figure 4. Predicted vs. actual stock prices. It compares actual and predicted stock prices, showcasing the hybrid model's ability to capture data patterns accurately and demonstrating the effectiveness of combining Bayesian optimization with LSTM networks for enhanced convergence and minimized prediction errors in financial forecasting.

Figure 5 depicts the Mean Absolute Error (MAE) metric, which quantifies the average absolute disparity between the actual and predicted values. The x -axis denotes the epochs

or iterations of training, while the y -axis represents the magnitude of the MAE. The training MAE curve portrays the evolution of the model's performance throughout training on the training dataset across numerous epochs. Initially, the training MAE tends to be high as the model commences learning from the data. However, with the progression of training, it typically diminishes, signifying an enhancement of the model's predictive accuracy on the training data. Conversely, the validation MAE curve evaluates the model's performance on a distinct validation dataset, serving as an indicator of its real-world performance and its capability to generalize to novel, unseen data. The objective is for the validation MAE to initially decrease as the model learns, stabilizing or experiencing a slight increase if it begins to overfit the training data. A comparative analysis of the training and validation MAE curves facilitates the assessment of potential overfitting or underfitting: a scenario where training MAE decreases while validation MAE increases indicates overfitting, while stable or decreasing curves suggest effective learning and generalization. This comparison aids in measuring the model's capacity to perform well on unseen data and informs adjustments to training strategies to enhance overall performance.

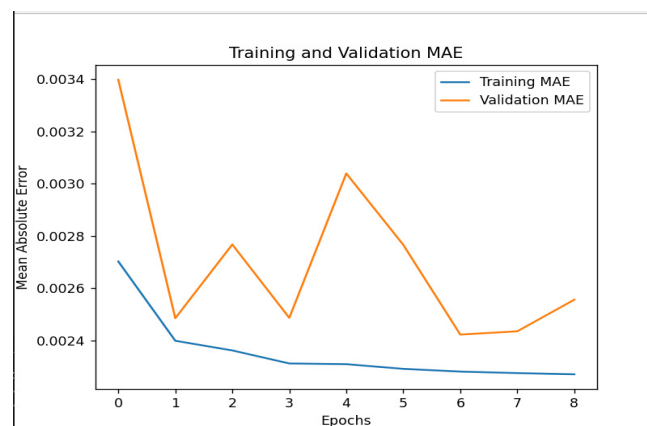


Figure 5. Training and validation MAEs. It illustrates the Mean Absolute Error (MAE) trends over training epochs, comparing the model's performance on training and validation datasets. Initially, the training MAE is high, decreasing as the model learns, indicating improved accuracy. The validation MAE reflects how well the model generalizes to new data, with a stable or slightly increasing trend suggesting overfitting. This comparison helps assess the model's effectiveness and informs training adjustments to optimize performance on unseen data.

Figure 6 presents the residual plot, a critical visualization aiding in the assessment of regression model performance. In this plot, the x -axis represents the predicted values, while the y -axis displays the corresponding residuals, which are the differences between the observed and predicted values. In an ideally fitted model, the residuals should be randomly distributed around the horizontal line at $y = 0$, indicating that predictions are unbiased and accurately capture the data's underlying patterns. Notably, upon examination of Figure 3.4, it was observed that nearly all data points align along the $y = 0$ line. This alignment signifies that the model's predictions closely match the actual observed values across the dataset. The absence of discernible patterns or trends in the residuals suggests that the model adequately captures the relationship between the independent and dependent variables, and its predictions are unbiased. To conclude, the residual plot's visualization confirms the model's robustness and suggests that it effectively captures the dataset's variability, contributing to its reliability in making accurate predictions.

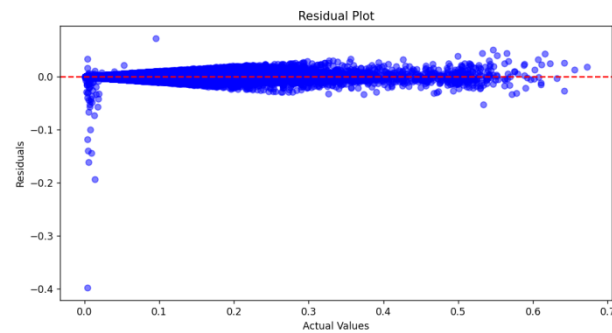


Figure 6. Residual plot. It displays the residual plot, showing the differences between observed and predicted values. Ideally, residuals should be randomly scattered around $y = 0$, indicating unbiased predictions. In this case, most data points align closely along the $y = 0$ line, suggesting that the model accurately captures the data patterns. The lack of visible trends in the residuals confirms the model's robustness and reliability in making accurate predictions.

The plot presented in Figure 7 illustrates the training and validation loss of the proposed hybrid predictive model across epochs or iterations of training. The x -axis denotes the number of epochs or iterations, while the y -axis portrays the magnitude of the loss function. The curve representing training loss depicts how the loss on the training dataset evolves as the model undergoes training over numerous epochs. Initially, the training loss tends to be high, as the model's parameters are randomly initialized. However, as training progresses, the model adjusts its parameters to minimize this loss, leading to a decline in training loss over time. This reduction signifies the model's improvement in fitting the training data. Similarly, the curve reflecting validation loss assesses the loss on a separate validation dataset that the model has not encountered during training. This dataset serves as an indicator of real-world performance and aids in evaluating the model's generalization to new, unseen data. Comparing the training loss and validation loss curves allows for an assessment of whether the model is overfitting or underfitting. If the training loss continues to decrease while the validation loss increases, it suggests overfitting of the training data and potential limitations in generalization to new data. Conversely, if both curves decrease and stabilize at similar levels, it indicates effective learning and successful generalization, as depicted in Figure 5.

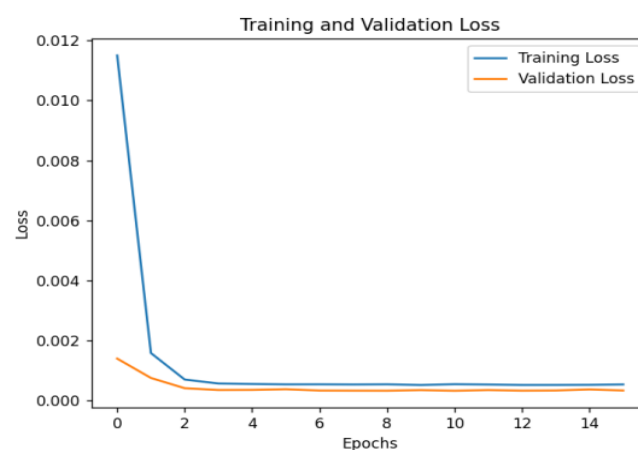


Figure 7. Training and validation losses. It shows the training and validation loss curves across epochs, illustrating the model's learning progress, where decreasing and stabilizing losses indicate effective learning and good generalization, while divergence suggests overfitting.

4.3. Comparative Analysis

To evaluate the performance of the Bayesian-optimized Long Short-Term Memory (BOLSTM) and standard LSTM models, a paired t -test was conducted on several perfor-

mance metrics following 10-fold cross-validation of the dataset. This approach ensures that the results are robust and not dependent on a single training–test split. The analysis revealed that the BOLSTM model significantly outperformed the LSTM model across all evaluated metrics. Specifically, the BOLSTM model exhibited significantly lower values for the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The *t*-statistics recorded for these metrics were -91.5478 , -80.4320 , and -76.8281 , respectively, with corresponding *p*-values of 0.0000 for each comparison. Furthermore, the BOLSTM model achieved a higher R-squared (R^2) value, indicated by a *t*-statistic of 54.3696 and a *p*-value of 0.0000, reflecting the superior explanatory power of the model. Additionally, the Mean Absolute Percentage Error (MAPE) was notably lower for the BOLSTM model, with a *t*-statistic of 69.3377 and a *p*-value of 0.0000. These results collectively underscore the substantial advantages of the BOLSTM model over the LSTM model in terms of predictive accuracy and performance metrics.

Researchers aiming to replicate this analysis using different time series data must ensure that the dataset is clean and formatted properly. Handle missing values and normalize the data using techniques such as Min–Max scaling. Create sequences and corresponding labels for the model inputs, and implement a hyperparameter optimization technique such as Bayesian Optimization. Define the search space for hyperparameters, including the number of LSTM units, learning rates, dropout rates, and batch sizes. Utilize the objective function to minimize the prediction error on a validation set, and adopt the same performance evaluation metrics used in this study: MSE, MAE, RMSE, R^2 , and MAPE. Utilize statistical tests like paired *t*-tests to compare the performance of models.

The performance comparison between the BOLSTM and LSTM models is illustrated in Table 5 and Figure 8. A summary of the performance metrics for both models is also presented.

Table 5. Comparison of BO-LSTM and LSTM-only.

	BOLSTM_MSE	BOLSTM_MAE	BOLSTM_RMSE	BOLSTM_R2	BOLSTM_MAPE	LSTM_MSE	LSTM_MAE	LSTM_RMSE	LSTM_R2	LSTM_MAPE
0	0.000005	0.001	0.002482	0.967518	0.030386	0.000154	0.014848	0.024132	0.744425	0.010763
1	0.000005	0.001	0.002287	1.089914	0.029556	0.000149	0.014846	0.02223	0.838599	0.010469
2	0.000006	0.001036	0.002321	0.996856	0.029135	0.000155	0.015384	0.02256	0.766998	0.01032
3	0.000006	0.000926	0.002148	0.944774	0.028863	0.000161	0.013746	0.020882	0.726925	0.010224
4	0.000005	0.000935	0.00225	1.038555	0.027138	0.000148	0.013889	0.021872	0.799082	0.009613
5	0.000005	0.000995	0.002326	0.936638	0.02825	0.000148	0.014773	0.022609	0.720665	0.010006
6	0.000006	0.000972	0.00218	1.007946	0.028629	0.000162	0.01443	0.02119	0.775531	0.010141
7	0.000006	0.00104	0.002356	0.899788	0.030853	0.000156	0.015439	0.022906	0.692312	0.010929
8	0.000005	0.000977	0.002243	0.931284	0.029808	0.000146	0.01451	0.021809	0.716546	0.010558
9	0.000005	0.000951	0.002279	1.007348	0.026721	0.000154	0.014127	0.022156	0.775071	0.009465

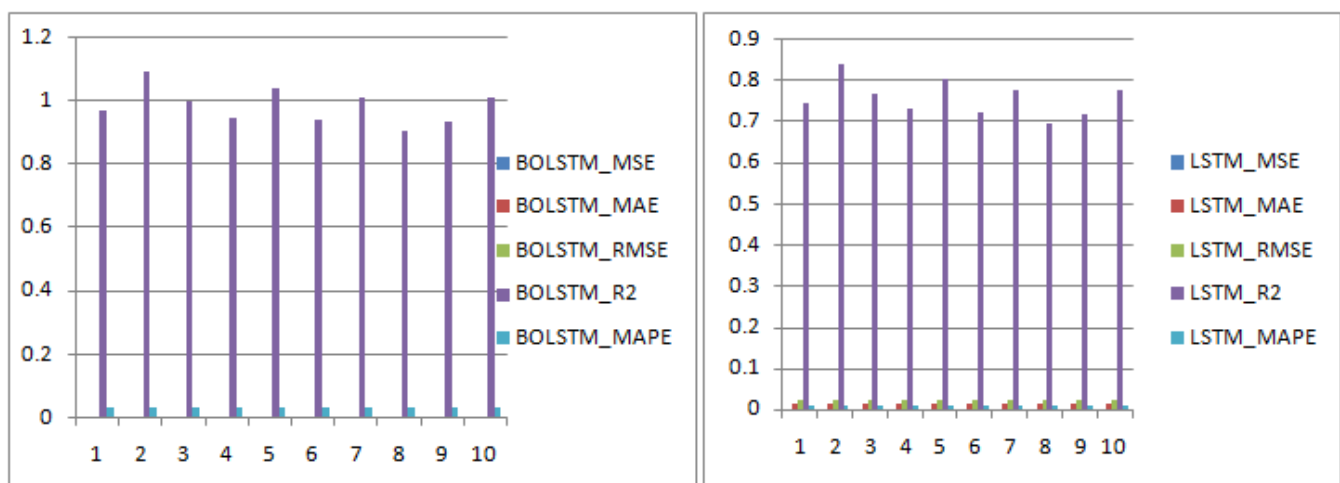


Figure 8. Plots comparing BO-LSTM and LSTM-only.

As observed in Table 5, for all metrics (MSE, MAE, RMSE, R^2 , and MAPE), the BOLSTM model significantly outperforms the LSTM model, as indicated by the very low

p -values (0.0000) and large absolute values of the t -statistics. This suggests that the observed differences in performance are statistically significant and not due to random chance, as visually depicted in Figure 8.

The proposed financial forecasting scheme demonstrated superior performance compared to other models in the literature, as shown by the metrics in the comparison table. Specifically, our scheme achieved very low values for key metrics: an MSE of 0.00000535, MAE of 0.00102352, RMSE of 0.00231278, R^2 of 0.99752901, and MAPE of 0.02930441. These results highlight its accuracy and robustness. For instance, Mudassir et al. reported an MAE of 39.50 and an RMSE of 74.10 for next-day forecasts, and their 7-day forecast performance (MAE of 16.32 and RMSE of 37.32) was also significantly higher than our scheme's metrics. Mallqui and Fernandes' model showed an MAE of 14.32 and an RMSE of 25.47, while Tripathi and Sharma's 7-day forecast had an MAE of 171.97 and RMSE of 192.33, both of which were substantially less accurate than our results. Liwei et al.'s LSTMBO-XGBoost model had a notably higher RMSE of 610.35 and MAE of 15.60. Additionally, Bhambu et al.'s RedRVFL model, although innovative, showed an RMSE of 2.63 and MAE of 3.27, again trailing behind our proposed scheme.

5. Discussion

This study contributes to the existing body of literature by demonstrating the superior predictive accuracy of the Bayesian-LSTM model in financial forecasting compared to traditional and contemporary methods. The Bayesian-LSTM model exhibits exceptionally low values for Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), surpassing other models such as ARIMA, CNN, and CNN-BiLSTM in handling the complexities of financial data. These findings highlight the model's potential to effectively capture nonlinear patterns and uncertainties in financial time series, providing a more robust and reliable forecasting tool.

To enhance the validity of model comparisons, additional experiments were conducted to evaluate the performance of the Bayesian-LSTM model against three additional state-of-the-art (SOTA) models, specifically LSTM with Attention Mechanisms, the CNN-BiLSTM-Attention model and the PSO-based ensemble of LSTM and GRU models.

All models were assessed using the same datasets under identical conditions when compared with the proposed model, emphasizing the importance of controlling variables and maintaining consistency in experimental setups. As depicted in Table 6, key performance metrics were standardized across all models, including MSE, MAE, RMSE, R^2 , and MAPE, offering a clear framework for assessing predictive accuracy.

The findings reinforce the growing trend of utilizing advanced machine learning and deep learning techniques in financial forecasting. The Bayesian-LSTM model demonstrates adaptability across multiple datasets, including the China Stock Price dataset and various stock exchanges such as the National Stock Exchange of India (NSE) and the New York Stock Exchange (NYSE). The model's performance consistently outperforms traditional methods, indicating its ability to capture complex patterns and long-term dependencies in financial data.

While significant strides have been made in advancing the field of financial forecasting through the implementation of the Bayesian-LSTM model, certain limitations remain. The model currently relies solely on historical price data, which may limit its responsiveness to external factors such as news sentiment or economic indicators. Additionally, the computational demands of the model could hinder scalability for real-time applications.

Future research should focus on testing the Bayesian-LSTM model on a broader range of real-world data to validate its robustness across various market conditions. Expanding SOTA comparisons to include a wider array of leading forecasting methods will refine the competitive analysis and identify areas for improvement. Exploring the model's performance in volatile or atypical market environments could also enhance its robustness and versatility.

Table 6. SOTA comparison.

Authors	Methods	Dataset	Results
Proposed Model	Bayesian–LSTM	China Stock Price	MSE:0.00000535
			MAE: 0.00102352
			RMSE: 0.00231278
			R ² : 0.99752901
			MAPE: 0.02930441
		CAMBRIA Dataset	MAE: 7.60
			MAPE: 2.06%
			RMSE: 1.065
			MSE: 2.28
		Financial Times Stock Exchange Milano Indice di Borsa	MAE: 0.0515
			RMSE: 0.0437
			MAPE: 0.0881
		(National Stock Exchange of India) and NYSE (New York Stock Exchange)	MAPE 1.484% from MARUTI
		GOOGL Stocks	MAE: 0.66%
			RMSE: 0.44%
			MAPE: 0.90%
		Dhaka Stock Exchange	RMSE: 0.78048
			MAE: 0.6226668
[24]	CNN, BiLSTM, Attention Mechanism	CAMBRIA Dataset, KRIRAN dataset, SHARMA dataset, JAMES dataset	MAE: 15.20
			MAPE: 4.12%
			RMSE: 2.13
[25]	DCGAN	Financial Times Stock Exchange Milano Indice di Borsa	MSE: 4.56
			RMSE: 0.112
			MAE: 0.132
[27]	CNN	(National Stock Exchange of India) and NYSE (New York Stock Exchange)	MAPE: 0.226
			MAPE values across companies like MARUTI, HCL, and AXIS BANK were 5.30%, 6.33% and 7.32, respectively.
			The MAE, RMSE, and MAPE values range from 1.10% to 1.50%.
[28]	ARIMA	AAPL, MSFT, NFLX, and GOOGL Stocks	RMSE:4.336
[29]	ARIMA	Dhaka Stock Exchange	MAE: 3.45926
			MAPE: 1.79%
[30]	Multi-Layered Sequential LSTM with the Adam Optimizer	GOOGL Stocks	RMSE:0.019
			MAPE: 1.56%
[31]	CNN–BiLSTM–Attention model	China Stock Price	RMSE:0.0076
			MAE: 0.00102352
[32]	PSO and ensemble of LSTM + GRU models	China Stock Price	RMSE: 0.00231278
			R ² : 0.83752901

By addressing these aspects and integrating additional SOTA comparisons under controlled conditions, future studies can provide more comprehensive insights and solidify the Bayesian–LSTM model’s contribution to financial forecasting, thereby advancing both theoretical knowledge and practical applications in the industry.

6. Computational Setup

To ensure reproducibility and clarity, the computational setup utilized in this study is detailed in this section. The hardware configuration consisted of an Intel Core i7 processor, 8 GB of RAM, a 1 TB HDD, and a 250 GB SSD. The software environment included Visual Studio Code and Python 3.8, along with several libraries, specifically TensorFlow, pandas, scikit-learn, Keras, Hyperopt, Matplotlib, Stats models, and Seaborn, all operating on a Windows 10 platform. The execution times for the proposed method were evaluated across multiple test cases, as summarized in Table 7. These execution times provide valuable insights into the method’s performance characteristics and efficiency.

Table 7. Execution times for test cases.

Test Case	Execution Time (Seconds)
Test Case 1	12.5
Test Case 2	15.3
Test Case 3	10.8
Test Case 4	18.6
Test Case 5	14.2

As illustrated in Table 7, the method completed Test Case 1 in approximately 12.5 s, Test Case 2 in around 15.3 s, Test Case 3 in 10.8 s, Test Case 4 in 18.6 s, and Test Case 5 in about 14.2 s.

7. Conclusions

In conclusion, this study has demonstrated the effectiveness and potential of the proposed approach in enhancing financial predictions within dynamic markets. Through the integration of nature-inspired optimization techniques and deep learning methodologies, our hybrid framework has showcased promising results with very low MSE, MAE, RMSE, R^2 and MAPE results of 0.00000535, 0.00102352, 0.00231278, 0.99752901 and 0.02930441, respectively, indicating its capability to accurately capture and forecast market trends. Despite the notable advancements achieved, it is imperative to acknowledge the limitations and areas for future exploration, such as further refinement of the hybrid approach and exploration of novel techniques for feature engineering and model interpretability. Overall, this research represents a significant step forward in the realm of predictive analytics, offering valuable insights and opportunities for continued innovation in financial forecasting.

Author Contributions: Conceptualization, J.O. and D.O.; methodology, J.O. and D.O.; software, J.O.; validation, J.O. and D.O.; formal analysis, J.O. and D.O.; investigation, I.C.O. and O.P.B.; resources, B.M.E.; data curation, J.O.; writing—original draft, J.O. and D.O.; writing—review and editing, D.O.; visualization, J.O. and D.O.; supervision, I.C.O., B.M.E. and A.A.A.; project administration, I.C.O., B.M.E., A.A.A. and O.P.B.; funding acquisition, B.M.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by Bukohwo Michael Esiefarienrhe.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available at <https://www.kaggle.com/datasets/franciscofeng/chinese-stock-data-from-20052022>, accessed on 7 November 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, Y.; Yu, Y.; Shakeel, P.M.; Montenegro-Marin, C.E. Research on operational research-based financial model based on e-commerce platform. *Inf. Syst. E-Bus. Manag.* **2021**, *21*, 17. [\[CrossRef\]](#)
2. Thakkar, A.; Chaudhari, K. Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions. *Inf. Fusion* **2021**, *65*, 95–107. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Patel, K.R. Enhancing Global Supply Chain Resilience: Effective Strategies for Mitigating Disruptions in an Interconnected World. *Bullet J. Multidisiplin Ilmu* **2023**, *2*, 257–264.
4. Sarker, I.H. Data science and analytics: An overview from data-driven smart computing, decision-making and applications perspective. *SN Comput. Sci.* **2021**, *2*, 377. [\[CrossRef\]](#)
5. Kumar, G.; Jain, S.; Singh, U.P. Stock market forecasting using computational intelligence: A survey. *Arch. Comput. Methods Eng.* **2021**, *28*, 1069–1101. [\[CrossRef\]](#)
6. Doering, J.; Kizys, R.; Juan, A.A.; Fito, A.; Polat, O. Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Oper. Res. Perspect.* **2019**, *6*, 100121. [\[CrossRef\]](#)
7. Bartram, S.M.; Branke, J.; Motahari, M. *Artificial Intelligence in Asset Management*; CFA Institute Research Foundation: Charlottesville, VA, USA, 2020.
8. Ghosh, I.; Jana, R.K. A granular machine learning framework for forecasting high-frequency financial market variables during the recent black swan event. *Technol. Forecast. Soc. Chang.* **2023**, *194*, 122719. [\[CrossRef\]](#)
9. Bhandari, H.N.; Rimal, B.; Pokhrel, N.R.; Rimal, R.; Dahal, K.R.; Khatri, R.K. Predicting stock market index using LSTM. *Mach. Learn. Appl.* **2022**, *9*, 100320. [\[CrossRef\]](#)
10. Gandhmal, D.P.; Kumar, K. Systematic analysis and review of stock market prediction techniques. *Comput. Sci. Rev.* **2019**, *34*, 100190. [\[CrossRef\]](#)
11. Cavalcante, R.C.; Brasileiro, R.C.; Souza, V.L.; Nobrega, J.P.; Oliveira, A.L. Computational intelligence and financial markets: A survey and future directions. *Expert Syst. Appl.* **2016**, *55*, 194–211. [\[CrossRef\]](#)
12. Gourieroux, C. *Financial Econometrics: Problems, Models and Methods*; Princeton University Press: Princeton, NJ, USA, 2018.
13. Strielkowski, W.; Vlasov, A.; Selivanov, K.; Muraviev, K.; Shakhnov, V. Prospects and Challenges of the Machine Learning and Data-Driven Methods for the Predictive Analysis of Power Systems: A Review. *Energies* **2023**, *16*, 4025. [\[CrossRef\]](#)
14. Zulfiqar, M.; Gamage, K.A.; Kamran, M.; Rasheed, M.B. Hyperparameter optimization of bayesian neural network using bayesian optimization and intelligent feature engineering for load forecasting. *Sensors* **2022**, *22*, 4446. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Bharadiya, J.P. Exploring the Use of Recurrent Neural Networks for Time Series Forecasting. *Int. J. Innov. Sci. Res. Technol.* **2023**, *8*, 2023–2027.
16. Tang, Y.; Song, Z.; Zhu, Y.; Yuan, H.; Hou, M.; Ji, J.; Tang, C.; Li, J. A survey on machine learning models for financial time series forecasting. *Neurocomputing* **2022**, *512*, 363–380. [\[CrossRef\]](#)
17. Chen, S.; Goo, Y.J.J.; Shen, Z.D. A hybrid approach of stepwise regression, logistic regression, support vector machine, and decision tree for forecasting fraudulent financial statements. *Sci. World J.* **2014**, *2014*, 968712. [\[CrossRef\]](#)
18. Van Houdt, G.; Mosquera, C.; Nápoles, G. A review on the long short-term memory model. *Artif. Intell. Rev.* **2020**, *53*, 5929–5955. [\[CrossRef\]](#)
19. Malsa, N.; Vyas, V.; Gautam, J. RMSE calculation of LSTM models for predicting prices of different cryptocurrencies. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–9. [\[CrossRef\]](#)
20. Wu, D.; Wang, X.; Su, J.; Tang, B.; Wu, S. A labeling method for financial time series prediction based on trends. *Entropy* **2020**, *22*, 1162. [\[CrossRef\]](#)
21. Siarni-Namini, S.; Namin, A.S. Forecasting economics and financial time series: ARIMA vs. LSTM. *arXiv* **2018**, arXiv:1803.06386.
22. Zhang, X.; Liang, X.; Zhiyuli, A.; Zhang, S.; Xu, R.; Wu, B. At-lstm: An Attention-Based Lstm Model for Financial Time Series Prediction. In Proceedings of the 5th Annual International Workshop on Materials Science and Engineering, Changsha, China, 17–18 May 2019; IOP Publishing: Bristol, UK; Volume 569.
23. Dixon, M.; London, J. Financial forecasting with α -RNNs: A time series modeling approach. *Front. Appl. Math. Stat.* **2021**, *6*, 551138. [\[CrossRef\]](#)
24. Zhao, R.; Lei, Z.; Zhao, Z. Research on the application of deep learning techniques in stock market prediction and investment decision-making in financial management. *Front. Energy Res.* **2024**, *12*, 1376677. [\[CrossRef\]](#)
25. Staffini, A. Stock price forecasting by a deep convolutional generative adversarial network. *Front. Artif. Intell.* **2022**, *5*, 837596. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Zheng, H.; Wu, J.; Song, R.; Guo, L.; Xu, Z. Predicting financial enterprise stocks and economic data trends using machine learning time series analysis. *Appl. Comput. Eng.* **2024**, *87*, 26–32. [\[CrossRef\]](#)
27. Ghosh, B.P.; Bhuiyan, M.S.; Das, D.; Nguyen, T.N.; Jewel, R.M.; Mia, M.T.; Shahid, R. Deep Learning in Stock Market Forecasting: Comparative Analysis of Neural Network Architectures Across NSE and NYSE. *J. Comput. Sci. Technol. Stud.* **2024**, *6*, 68–75. [\[CrossRef\]](#)

28. Sharma, R.; Sharma, R.; Hariharan, S. Stock Price Prediction Using ARIMA: A Study on AAPL, MSFT, NFLX, and GOOGL Stocks. In Proceedings of the 2024 International Conference on Computational Intelligence and Computing Applications (ICCICA), Panipat, India, 23–24 May 2024; Volume 1, pp. 47–50.
29. Ahammad, I.; Sarkar, W.A.; Meem, F.A.; Ferdus, J.; Ahmed, M.K.; Rahman, M.R.; Sultana, R.; Islam, M.S. Advancing Stock Market Predictions with Time Series Analysis Including LSTM and ARIMA. *Cloud Comput. Data Sci.* **2024**, *5*, 226–241. [[CrossRef](#)]
30. Md, A.Q.; Kapoor, S.; Juni, A.V.C.; Sivaraman, A.K.; Tee, K.F.; Sabireen, H.; Janakiraman, N. Novel optimization approach for stock price forecasting using multi-layered sequential LSTM. *Appl. Soft Comput.* **2023**, *134*, 109830. [[CrossRef](#)]
31. Zhang, J.; Ye, L.; Lai, Y. Stock price prediction using CNN-BiLSTM-Attention model. *Mathematics* **2023**, *11*, 1985. [[CrossRef](#)]
32. Chauhan, A.; Shivaprakash, S.J.; Sabireen, H.; Md, A.Q.; Venkataraman, N. Stock price forecasting using pso hypertuned neural nets and ensembling. *Appl. Soft Comput.* **2023**, *147*, 110835. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Reproduced with permission of copyright owner. Further reproduction
prohibited without permission.