Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.2, April 2026, pp. 1282~1290
e-ISSN: 2548-6861

Bayesian-Optimized LSTM Framework for Accurate Stock Price Prediction

Muhammad Idham Maulana[1]*, Syaiful Anam[2]*, Hilmi Aziz Bukhori[3]*
*Mathematics Department, Universitas Brawijaya
Corresponding: syaiful@ub.ac.id[2], idhammaulana12@student.ub.ac.id[1], hilmibukhori@ub.ac.id[3]

Article Info

Article history:

Received 2025-11-12
Revised 2026-03-18
Accepted 2026-04-10

Keyword:

Bayesian Optimization,
Deep Learning,
Financial Forecasting,
LSTM,
Stock Price Prediction.

ABSTRACT

Stock price prediction remains a challenging task due to the highly volatile and non-linear nature of financial market data. Long Short-Term Memory (LSTM) networks have shown remarkable success in modelling temporal dependencies, yet their predictive performance heavily depends on optimal hyperparameter tuning. Conventional methods such as Grid Search and Random Search are often computationally expensive and suboptimal. This study proposes a systematic and data-driven framework that integrates Bayesian Optimization (BO) to enhance the performance of LSTM models for stock price prediction (LSTM+BO Model). Using Amazon (AMZN) daily stock data from 2019 to 2025, the LSTM+BO model was rigorously compared with a standard LSTM and several deep learning and machine learning benchmarks. All models were evaluated over 25 independent runs to ensure statistical reliability. The results demonstrate that the LSTM+BO model achieved the lowest Mean Absolute Percentage Error (2.4413%) and the highest R² score (0.8736), outperforming all benchmarks. Moreover, the optimized model exhibited greater stability and computational efficiency compared to the default configuration. These findings confirm that BO offers an effective and robust approach for systematically developing accurate and efficient forecasting models in financial analytics, providing a strong foundation for the development of automated and adaptive financial forecasting systems.

I. INTRODUCTION

Stock price prediction remains one of the most challenging tasks in financial analytics, mainly due to the complex, non-linear, and non-stationary nature of stock market data [1], [2]. Accurately forecasting stock price movement—particularly the next-day closing price, is essential for investors to design strategic decisions, mitigate financial risks, and gain competitive advantages. The dynamic behaviour of the stock market, influenced by numerous internal and external factors, renders traditional forecasting models insufficient [3].

Time series analysis is a quantitative method that examines the temporal relationships between sequential data points. The primary objective of this analysis is to identify and formulate a precise model that effectively expresses the structured temporal dependencies within a dataset. Such a model can then be used to either evaluate these historical relationships or, more commonly, to forecast future values [4]. Historically, statistical methods such as the Autoregressive Integrated Moving Average (ARIMA) model have been widely employed. While these models offer a solid mathematical foundation, their reliance on assumptions of linearity and stationarity often limits their effectiveness in capturing the volatility inherent in financial time series, prompting a shift toward more flexible, data-driven approaches [5].

Deep learning techniques, particularly the Long Short-Term Memory (LSTM) network, have emerged as powerful tools for modelling sequential financial data. LSTM, a variant of Recurrent Neural Networks (RNNs), was introduced to overcome the vanishing gradient problem that hampers long-term dependency learning[6]. Leveraging its specialized gating mechanisms, LSTM has demonstrated superior performance, outperforming not only traditional methods like ARIMA [7], [8] but also non-sequential models such as

Support Vector Regression (SVR), Random Forests and Deep Neural Networks (DNNs) [9], [10].

While powerful, the effectiveness of LSTM is not guaranteed out-of-the-box. Its architecture involves numerous hyperparameters—such as the number of hidden units, learning rate, and dropout ratio—that must be carefully tuned. Consequently, the predictive capability of the model is highly sensitive to this configuration. Improper tuning can result in underfitting, overfitting, or unstable convergence. Manual tuning or simple search strategies like Grid Search and Random Search are often inefficient and fail to guarantee optimal configurations [11], [12]. To address this, Bayesian Optimization (BO) has gained attention as an effective method for automatic hyperparameter tuning, demonstrating superior performance over Grid Search and Random Search [13]. BO employs a probabilistic surrogate model to intelligently navigate the search space, proving effective in reducing computational cost while improving model performance in deep learning applications [14].

Although several studies have successfully explored the integration of BO with hybrid LSTM models for financial forecasting [15], [16], a significant gap remains in establishing a methodologically rigorous framework that systematically identifies an optimal and reproducible LSTM architecture.

Although several studies have explored the integration of BO with LSTM models for financial forecasting, most existing works primarily focus on improving predictive accuracy. For instance, Huang et al. applied BO to tune a limited set of LSTM hyperparameters, mainly focusing on the number of input units (window size U) [17]. Similarly, other studies have incorporated BO into hybrid deep learning architectures, such as CNN–LSTM models, to enhance stock prediction performance [18]. While these studies demonstrate the effectiveness of optimization techniques, their evaluation protocols are often limited in scope.

In many existing studies, model performance is often reported based on a single training run or a limited number of repetitions, which may not adequately capture the stochastic nature of deep learning processes. As a result, critical aspects such as model stability, reproducibility, and computational efficiency remain underexplored, while most studies primarily emphasize predictive accuracy with limited attention to the trade-off between performance and computational cost. This issue is also evident in LSTM-based and Bayesian-optimized models for stock prediction. For example, Huang et al. (2018) evaluate their Bayesian-LSTM using a single experimental setting [17], while Chan et al. (2024) optimize CNN–LSTM architectures but report results based on limited configurations and validation accuracy [18]. Similarly, Chen et al. (2015), rely on a single train–test split and do not report performance variability across multiple runs [19]. Such practices may lead to overestimated or unstable performance evaluations.

Additionally, several studies rely on limited hyperparameter tuning strategies or focus primarily on architectural complexity, such as hybrid neural network structures [20]. As a result, there is still a lack of systematic experimental frameworks that evaluate optimization-based forecasting models from multiple perspectives, including stability across repeated runs, statistical robustness, and computational efficiency. Addressing these aspects is essential to ensure that performance improvements obtained through optimization techniques are not only accurate but also reliable and reproducible in practical financial forecasting applications.

As demonstrated in several studies [21], [22], many applications tend to rely on default parameters or heuristically selected hyperparameters from previous work, which may not be optimal for different datasets or market conditions. Therefore, the primary contribution of this research is the development and empirical validation of a systematic framework for optimizing LSTM models using BO for stock price prediction. This study moves beyond a straightforward application of LSTM by providing a robust experimental framework to identify high-performing and stable model configurations.

To address these limitations, this study proposes a systematic experimental framework for optimizing LSTM models using BO for stock price prediction (LSTM+BO). Unlike many previous works that emphasize architectural complexity or limited hyperparameter tuning, this research focuses on evaluating the effectiveness, robustness, and computational characteristics of LSTM+BO models. The main contributions of this study are summarized as follows:

1. Comprehensive hyperparameter optimization framework. BO is employed to simultaneously optimize multiple critical hyperparameters of the LSTM architecture, including window size, hidden units, hidden layers, dropout rate, learning rate, optimizer type, and batch size [23].
2. Stability-oriented experimental design.
   To account for the stochastic nature of deep learning training, each model is evaluated through 25 independent experimental runs, enabling a more reliable assessment of model robustness and reproducibility.
3. Statistical robustness analysis.
   The study reports both the mean and standard deviation of evaluation metrics across multiple runs to provide a clearer understanding of model stability.
4. Computational efficiency analysis.
   In addition to predictive accuracy, this research evaluates training runtime, allowing an analysis of the trade-off between model performance and computational cost.
5. Comprehensive benchmark comparison.
   The optimized LSTM model is compared with both deep learning architectures (RNN and Bi-LSTM) and classical machine learning models (KNN, Decision Tree, Random Forest, XGBoost, and LightGBM).

II. METHOD

This study employs a quantitative methodology to develop and evaluate the LSTM+BO model for stock price prediction. The research workflow, illustrated in Figure 1, begins with data acquisition and pre-processing, followed by the main experimental phase involving model optimization and comparative analysis, and concludes with a robust performance evaluation.

A. Data Collection and Pre-processing

The dataset used in this study consists of historical daily stock prices of Amazon (AMZN) from January 2, 2019, to January 2, 2025, obtained from Yahoo Finance using the *yfinance* library. It includes five standard financial features: Open, High, Low, Close, and Volume (OHLCV). The next-day closing price is used as the prediction target. Figure 1 illustrates the closing price trend over the observation period, showing notable fluctuations across different market conditions, while Table I summarizes the descriptive statistics. The Volume feature exhibits positive skewness (1.85), indicating a non-normal distribution; therefore, a logarithmic transformation was applied to stabilize variance and reduce skewness.

All features were subsequently normalized to the range [0,1] using Min–Max scaling to ensure numerical stability during model training. The time-series data were then transformed into sequential input samples using a sliding window approach, where each sequence contains a fixed number of past observations to predict the next-day closing price. The window size was treated as a tuneable hyperparameter and optimized using BO.

Given the non-stationary nature of financial time-series data, characterized by changing market conditions and volatility dynamics, this study addresses such properties through normalization and sequential modelling. By structuring the data into temporal sequences, the LSTM model can effectively capture evolving patterns and nonlinear dependencies without requiring strict stationarity assumptions.

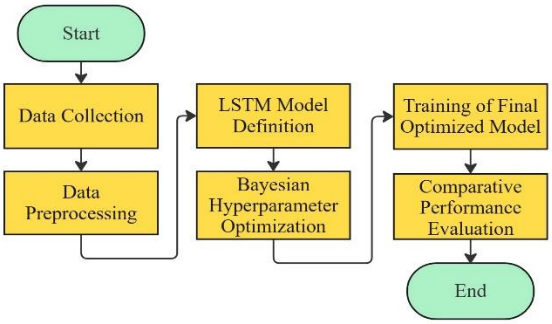

Figure 1. The research methodology workflow

Figure 2. AMZN daily closing price (2019-2025)

TABLE I
DESCRIPTIVE STATISTICS OF THE AMZN STOCK DATASET (2019-2025)

Feature	**Mean**	**Std**	**Min**	**Q1 (25%)**	**Median (50%)**	**Q3 (75%)**	**Max**	**Skew**
Open	137.23	36.35	73.26	99.04	140.86	166.62	232.39	0.06
High	138.73	36.64	76.90	100.65	143.38	168.29	233.00	0.05
Low	135.40	35.98	73.05	97.69	138.95	164.70	228.01	0.07
Close	137.10	36.29	75.01	99.25	140.60	166.40	232.93	0.06
Volume	69,944,091	33,888,940	15,007,500	47,923,150	61,690,000	83,887,000	311,346,000	1.85

Since financial time-series data are inherently sequential, the dataset was partitioned using a chronological split to prevent information leakage. The earlier portion of the dataset was used for model training, the subsequent portion was used as the validation set during the BO process, and the most recent observations were reserved as the test set for final performance evaluation. This temporal splitting strategy ensures that the model is trained only on past data while being evaluated on future observations, thereby preserving the realistic forecasting scenario and avoiding data leakage. The dataset was divided into 70% training, 15% validation, and 15% testing sets following the chronological order of the data [24].

B. Model Architecture and Theoretical Framework

This section details the theoretical underpinnings of the primary model and the optimization strategy employed.

1) Long Short-Term Memory (LSTM): The core predictive model used in this research is the LSTM network, an advanced variant of RNNs specifically designed to address the vanishing gradient problem and effectively learn long-term dependencies in sequential data [6]. The key innovation of LSTM is its memory cell (C), which is regulated by three distinct gating mechanisms that control the flow of information. These gates, which consist of a sigmoid neural network layer (σ) and a pointwise multiplication operation, are crucial for the model's ability to retain relevant information over long sequences.

The computational mechanism at each time step t is as follows [24]:

1. Forget Gate ($\vec{f}_t$) : This gate decides which information from the previous cell state ($\vec{c}_{t-1}$) should be discarded. It analyzes the previous hidden state ($\vec{h}_{t-1}$) and the current input ($\vec{x}_t$) to output a number between 0 and 1 for each element in the cell state.
$$\vec{f}_t = \sigma\left(\vec{W}_{xf}^T \vec{x}_t + \vec{W}_{hf}\vec{h}_{t-1} + \vec{b}_f\right) \tag{1}$$
where $\vec{W}_{xf}$ and $\vec{W}_{hf}$ are the weight matrices for the input $\vec{x}_t$ and the hidden state $\vec{h}_{t-1}$, while $\vec{b}_f$ is the bias vector.
2. Input Gate ($\vec{\iota}_t$): This gate determines which new information should be stored in the cell state. It involves two parts: a sigmoid layer that decides which values to update and a tanh layer that creates a vector of new candidate values ($\tilde{c}_t$).
$$\vec{\iota}_t = \sigma\left(\vec{W}_{xi}^T \vec{x}_t + \vec{W}_{hi}\vec{h}_{t-1} + \vec{b}_i\right) \tag{2}$$
$$\vec{\tilde{c}}_t = tanh\left(\vec{W}_{xc}^T \vec{x}_t + \vec{W}_{hc}\vec{h}_{t-1} + \vec{b}_c\right) \tag{3}$$
in these equations, W_{xi} and W_{hi} are the weight matrices for the input and the hidden state with b_i as the corresponding bias vector, while $\vec{W}_{xc}$ and $\vec{W}_{hc}$ are the weight matrices and b_c is the bias vector for the candidate layer.
3. Cell State Update**:** The old cell state ($\vec{c}_{t-1}$) is updated into the new cell state ($\vec{c_t}$) by first forgetting the information deemed irrelevant and then adding the new candidate information.
$$\vec{c}_t = \vec{f}_t \odot \vec{c}_{t-1} + \vec{\iota}_t \odot \vec{\tilde{c}}_t \tag{4}$$
4. Output Gate ($\vec{o}_t$): This gate determines the output based on the updated cell state. The final output, which is the hidden state (h_t), is a filtered version of the cell state.
$$\vec{o}_t = \sigma\left(\vec{W}_{xo}^T \vec{x}_t + \vec{W}_{ho}\vec{h}_{t-1} + \vec{b}_o\right) \tag{5}$$
$$\vec{h}_t = \vec{o}_t \odot tanh(\vec{c}_t) \tag{6}$$
where $\vec{W}_{xo}$ and $\vec{W}_{ho}$ are the weight matrices, and $\vec{b}_o$ is the bias vector for the output gate.

This structured memory control allows the LSTM to selectively remember or forget information, making it highly effective for modelling the complex dynamics of financial time series.

2) Bayesian Hyperparameter Optimization: To determine the optimal architecture of the LSTM model, this study applies BO. BO works by constructing a probabilistic surrogate model specifically a Gaussian Process (GP) to approximate the objective function, such as validation loss. The optimization process is then guided by an acquisition function that balances exploration and exploitation, enabling the identification of the optimum with fewer evaluations [25].

In this research, the acquisition function used is the Upper Confidence Bound (UCB), defined as:

$$\alpha_{UCB}(\vec{x}) = \mu(\vec{x}) + \beta\sigma(\vec{x}) \tag{7}$$

where $\mu(\vec{x})$ denotes the mean prediction of the surrogate model (exploitation), $\sigma(\vec{x})$ represents the predictive uncertainty (exploration), and β is a hyperparameter that regulates the trade-off between the two. The hyperparameter search space adopted in the BO procedure is summarized in Table II.

TABLE II
HYPERPARAMETER SEARCH SPACE FOR BAYESIAN OPTIMIZATION

Hyperparameter	**Range**
Window Size	Integer [5, 60]
Hidden Units	Integer [32, 128]
Hidden Layers	Integer [1, 4]
Dropout Rate	Float [0.1, 0.5]
Learning Rate	Log-uniform [10−4, 10−2]
Optimizer	{'Adam', 'RMSprop'}
Batch Size	{32, 64, 128}

To systematically identify an optimal LSTM configuration, BO was employed to tune several key hyperparameters of the model. The optimization process explores the hyperparameter search space by constructing a probabilistic surrogate model that approximates the objective function. In particular, the BO framework utilizes a Gaussian Process (GP) surrogate model together with the UCB acquisition function to efficiently guide the search process. The acquisition function balances exploration of uncertain regions and exploitation of promising configurations, enabling efficient identification of high-performing hyperparameter settings.

C. Experimental Design and Benchmark Models

To rigorously evaluate the performance of the proposed LSTM+BO, a comprehensive comparative analysis was conducted. The experimental design is cantered on benchmarking the proposed model against a suite of standard models commonly used for time series forecasting. This approach allows for a clear quantification of the performance gains attributable to the BO process.

The benchmark models were selected from two main categories: other deep learning architectures and classic machine learning algorithms. All benchmark models were implemented with their standard (default) configurations to establish a fair and transparent baseline for comparison. The models are listed below:

1. Deep Learning Models: Standard LSTM, Simple RNN and Bi-LSTM.
2. Machine Learning Models: K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Extreme Gradient Boosting (XGBoost) and LightGBM

D. Evaluation Protocol

The evaluation protocol was systematically designed to ensure a robust, reliable, and comprehensive assessment of each model's performance.

- Experimental Setup

To account for the stochastic nature of model training, particularly in deep learning, each model both the LSTM+BO and all benchmark models was independently trained and evaluated 25 times to assess performance consistency [26]. Each run utilized the exact same data splits to ensure a fair and consistent comparison across all iterations and models.

- Evaluation Metrics

The predictive performance of all models was measured on the unseen test set using three complementary metrics:

1. Mean Absolute Percentage Error (MAPE)

   MAPE calculates the average absolute error as a percentage of the actual values, making it highly interpretable and independent of the data's scale [27].

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{Y_i - X_i}{Y_i}\right| \times 100\% \tag{8}$$

(best value = 0; worst value = +∞)

   where X_i is actual value, and Y_i is predicted value, and m is total amount of data.

2. Coefficient of Determination (R^2)

   This metric quantifies the proportion of the variance in the target variable that is predictable from the independent variables, indicating how well the model fits the data [27].

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y} - Y_i)^2} \tag{9}$$

(worst value = −∞; best value = +1)

- Statistical Analysis

The results from the 25 independent runs were aggregated and analysed to provide a comprehensive performance overview. The mean and standard deviation for each evaluation metric were calculated and tabulated. This statistical summary serves two primary purposes: to identify the model with the best average predictive accuracy and to assess the consistency and stability of each model's performance. A lower standard deviation indicates greater robustness and more reliable results across different training runs

III. RESULT AND DISCUSSION

This section presents the empirical results obtained from the hyperparameter optimization process and the subsequent comparative performance evaluation of the models. The findings are further analysed to interpret their significance and implications for financial time-series forecasting.

Compared with several previous BO+LSTM studies that primarily emphasize improvements in predictive accuracy, the present study provides a more comprehensive evaluation by examining model stability across multiple experimental runs and analysing computational efficiency. By conducting repeated experiments and reporting both the mean and standard deviation of the evaluation metrics, this study offers a more reliable assessment of model robustness.

The results demonstrate that BO not only improves predictive accuracy but also identifies model configurations that yield more stable performance and reduced training time compared with the default LSTM configuration. These findings highlight the importance of adopting a systematic experimental design when evaluating optimization-based deep learning models for financial forecasting tasks.

A. Bayesian Optimization Results

The BO process was executed over 50 iterations to refine the LSTM model's hyperparameters, with the objective of minimizing validation loss. A comparative summary of the resulting optimal configuration against the baseline is presented in Table III. The BO-derived configuration indicates a preference for a more complex model architecture, characterized by a substantially larger window size (13) and a greater number of LSTM units (82). Furthermore, the optimization process selected a higher learning rate (0.0062),

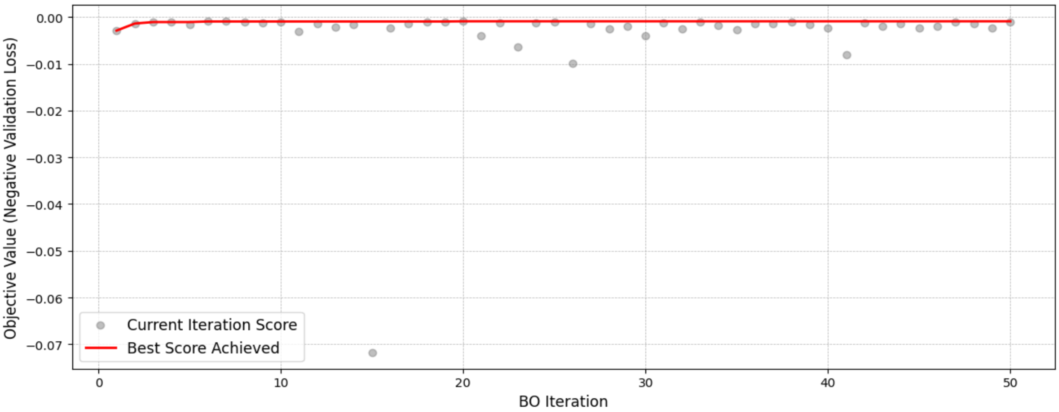

Figure 3. Bayesian optimization convergence plot

a marginally reduced dropout rate (0.1873), and a larger batch size (128). These adjustments suggest that a model with increased capacity for capturing long-term temporal dependencies and a faster convergence rate is optimal for this specific forecasting task.

TABLE III
COMPARASION OF LSTM MODEL CONFIGURATION

Hyperparameter	Optimization	Default
Window Size	13	7
Hidden Units	82	16
Hidden Layers	1	1
Dropout Rate	0.1873	0.2
Learning Rate	0.0062	0.001
Optimizer	Adam	Adam
Batch Size	128	32

The convergence behaviour of the BO process, depicted in Figure 3, demonstrates its efficacy in efficiently navigating the hyperparameter space. The best observed score exhibited rapid improvement during the initial iterations, indicating a swift transition from broad exploration to focused exploitation of promising regions. The trajectory of the best-score curve plateaued in later stages, with only minor fluctuations. These subsequent fluctuations represent controlled exploratory steps that failed to surpass the identified optimum, confirming the convergence of the search. The stable plateau suggests that additional iterations would yield diminishing returns. In conclusion, the BO process successfully and efficiently identified a robust hyperparameter set that enhances the LSTM model's predictive performance for stock price forecasting.

B. Comparative Performance Analysis

This section evaluates the Bayesian-Optimized LSTM (LSTM+BO) against a baseline LSTM (LSTM Default) and the benchmark models. Performance is assessed based on predictive accuracy (MAPE and R^2) and computational efficiency (runtime), with all metrics representing averages over 25 independent runs.

As summarized in Table IV, the LSTM+BO model demonstrated superior predictive performance, achieving the lowest average MAPE (2.4413%) and highest average R^2 (0.8736). The visual comparison in Figure 4 clearly illustrates the LSTM+BO's superior R^2 score, significantly outperforming all other models. Similarly, Figure 5 demonstrates its lowest MAPE value among all evaluated approaches. The standard LSTM (LSTM Default) performed credibly, ranking third in both metrics, which underscores the inherent strength of the LSTM architecture. Notably, all deep learning models substantially outperformed classical machine learning approaches, which exhibited R^2 values near or below zero, indicating their inability to effectively model this time series data.

TABLE IV
MEAN PERFORMANCE METRICS OVER 25 RUNS

Model	MAPE (%)	R^2	Runtime (s)
LSTM+BO	2.4413	0.8736	7.3590
RNN	2.7504	0.8376	14.8659
LSTM Default	2.7914	0.8306	23.4524
Bi-LSTM	3.4174	0.7266	20.4704
Decision Tree	5.1930	0.0307	0.0081
Random Forest	5.2343	0.0067	0.5788
KNN	5.3042	-0.0120	0.0038
XGBoost	5.4499	-0.0540	0.2388
LightGBM	5.4304	-0.0557	0.1559

The consistency and robustness of each model are quantified by the standard deviations presented in Table 4. The LSTM+BO model exhibited the lowest standard deviation in MAPE (0.2675) among all deep learning models, signifying highly stable and reliable predictive accuracy. Its stability in R^2 (0.0260) was also strong, being superior to that of the RNN and Bi-LSTM models. In contrast, the minimal variance observed in the classical models is attributed to their deterministic nature under fixed random seeds, rather than superior generalization.

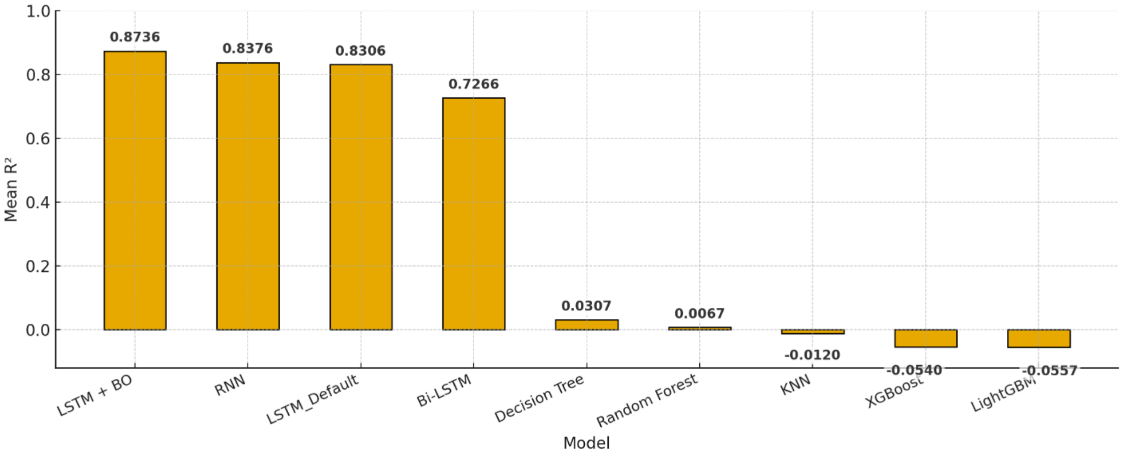

Figure 4. Average R² comparison across models

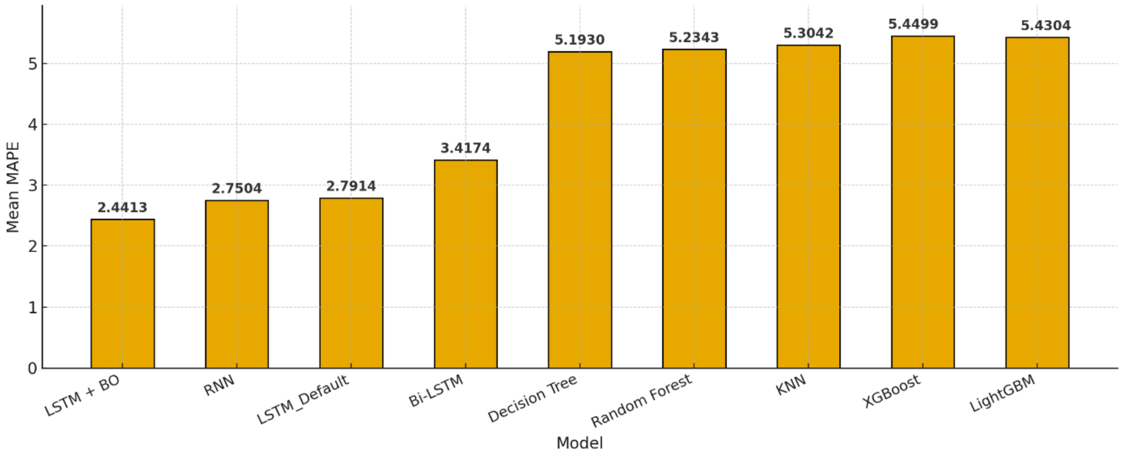

Figure 5. Average MAPE score comparison across models

Finally, the qualitative analysis in Figure 6 reveals that the LSTM+BO model successfully captures both the overall trend and major fluctuations in actual stock prices. Although minor deviations occur during periods of high volatility, the close alignment between predicted and actual values visually corroborates the model's strong quantitative performance.

In summary, the LSTM+BO model achieves superior accuracy with remarkable consistency, establishing it as the most effective approach for this stock prediction task among all evaluated models.

C. *Discussion*

This section discusses the interpretation of the experimental results presented earlier, contextualizes the findings within the research objectives, and elaborates on their implications and limitations.

Interpretation of key findings the experimental results consistently demonstrate that the LSTM model optimized using BO (LSTM+BO) delivered superior predictive performance compared to all benchmark models on the AMZN stock price dataset. This superiority is evident across both primary accuracy metrics: LSTM+BO achieved the lowest average MAPE (2.4413%) and the highest average R² (0.8736). These findings underscore the significant potential of systematic hyperparameter tuning to maximize the predictive capabilities of LSTM models. The BO convergence plot (Figure 3) further confirms the efficiency of the optimization process, which rapidly identified a high-performing hyperparameter region within a relatively small number of iterations.

Impact of BO direct comparison between LSTM+BO and LSTM Default highlights the quantitative impact of BO. The LSTM+BO model showed a marked improvement in accuracy, with an average MAPE approximately 12.5% lower (2.4413% vs 2.7914%) and an average R² approximately 5.2% higher (0.8736 vs 0.8306) than LSTM Default. Interestingly, optimization not only boosted accuracy but also computational efficiency; LSTM+BO was significantly faster in average execution time compared to LSTM Default (7.36 seconds vs 23.45 seconds). This is likely attributable to the optimized configuration (e.g., a larger batch size) enabling more efficient training. These results empirically validate that BO is not merely a tool for enhancing accuracy but can also identify configurations that are more computationally efficient.

TABLE V
STANDARD DEVIATION OF PERFORMANCE METRICS OVER 25 RUNS

Model	MAPE (%)	R^2	Runtime (s)
LSTM + BO	0.2675	0.0260	1.9729
RNN	0.7928	0.0896	2.3118
LSTM Default	0.4206	0.0525	3.0331
Bi-LSTM	0.7917	0.1085	3.0923
Decision Tree	0.0000	0.0000	0.0003
Random Forest	0.0102	0.0031	0.1838
KNN	0.0000	0.0000	0.0003
XGBoost	0.0098	0.0026	0.1855
LightGBM	0.0000	0.0000	0.0154

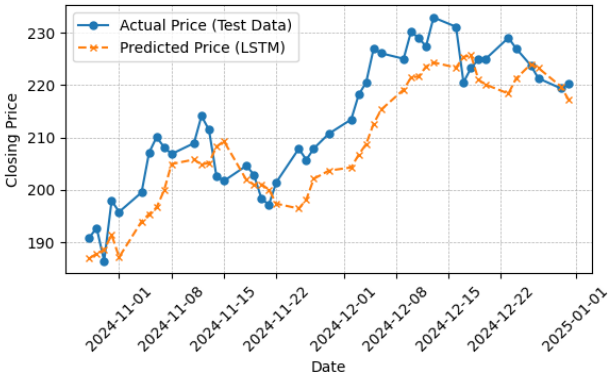

Figure 6. Actual vs. predicted price comparison for LSTM+BO model

Comparison with benchmark models among the DL models, LSTM+BO outperformed RNN and Bi-LSTM. Although RNN showed competitive performance and slightly better stability in terms of R², LSTM+BO remained superior in overall MAPE and R² accuracy. Bi-LSTM, despite its theoretical complexity, exhibited the lowest performance among the DL models, suggesting that increased complexity does not always translate to better accuracy for this specific dataset.

The performance disparity between DL models and classic machine learning models (KNN, Decision Tree, Random Forest, XGBoost, LightGBM) was substantial. The classic ML models demonstrated significantly higher MAPE (>5%) and R² scores near or below zero. A negative R² score, in particular, indicates that these ML models performed worse than a baseline model predicting the mean value, highlighting their inability to capture the complex patterns in the AMZN stock price data compared to sequential architectures like LSTM and RNN.

Model Stability and Robustness The standard deviation analysis (Table V) provides insights into model consistency. LSTM+BO demonstrated the highest MAPE stability among the DL models, evidenced by the lowest standard deviation (0.2675). While its R² standard deviation (0.0260) was slightly higher than RNN's, it remained relatively low, indicating reliable and reproducible performance. This stability is crucial for practical applications where consistent prediction quality is highly valued. The very low or zero standard deviations observed for the classic ML models are likely due to their deterministic nature (given a fixed random_state), rather than an indication of good predictive performance.

Computational trade-offs the runtime results confirm the inherent trade-off between accuracy and computational efficiency. Classic ML models were extremely fast (runtime < 0.6 seconds) but offered poor accuracy. Conversely, DL models required significantly more time but delivered far superior accuracy. Among the DL models, LSTM+BO presented an attractive balance: it achieved the best accuracy while also being the most computationally efficient (average runtime 7.36 seconds), outperforming RNN, LSTM Default, and Bi-LSTM.

Practically, this study offers a validated, data-driven methodology that financial analysts can employ to develop superior stock price forecasting systems, moving beyond default parameters. Theoretically, this research reinforces the effectiveness of BO as a robust hyperparameter tuning framework for deep learning models in the complex and non-linear financial domain. The findings confirm that systematic optimization is critical not only for accuracy but also for achieving computational efficiency and model stability.

Limitations despite the promising results, this study has several limitations. First, the evaluation was conducted solely on one stock asset (AMZN); generalization to other stocks with different characteristics (e.g., sector, market capitalization, volatility) requires further validation. Second, the model exclusively used historical price data and did not incorporate external factors such as market news, social media sentiment, or macroeconomic indicators, which are known to influence stock price movements. Third, the data period (2019-2025) covers specific market conditions (including the COVID-19 pandemic); model performance might differ under different market regimes.

Future work based on the findings and limitations, several avenues for future research can be explored. First, applying and validating this optimization framework on a more diverse portfolio of stocks, covering various sectors and markets. Second, integrating additional features, such as sentiment data or fundamental indicators, into the model to investigate potential accuracy improvements. Third, exploring alternative hyperparameter optimization algorithms or newer DL architectures as alternatives or benchmarks.

One limitation of this study is that the experimental evaluation was conducted using a single stock (AMZN). While Amazon provides a representative example of a highly traded stock with complex price dynamics, the findings may not fully generalize to other stocks with different market characteristics, sectors, or volatility patterns. Future research could extend this framework by applying the proposed BO–based LSTM approach to a broader range of assets and market conditions to further evaluate its robustness and generalizability.

IV. CONCLUSION

This study developed and evaluated a Bayesian BO framework for LSTM models in stock price prediction using Amazon (AMZN) data from 2019–2025. The experimental results demonstrate that the optimized LSTM+BO model achieves superior predictive performance, with a MAPE of 2.4413% and an R^2 value of 0.8736. In addition to improved accuracy, the optimized configuration also exhibits greater stability and computational efficiency compared with the default LSTM configuration and other benchmark models. The findings confirm that BO provides an effective and systematic approach for tuning LSTM hyperparameters in financial time-series forecasting tasks. By exploring a broader hyperparameter space and evaluating model performance across multiple independent runs, the proposed framework offers a more robust and reproducible methodology for developing predictive models in financial analytics. Nevertheless, this study has certain limitations. The experimental evaluation was conducted using a single stock (AMZN), which may limit the generalizability of the findings to other assets or market conditions. Future research may extend the proposed framework by applying it to multiple stocks from different sectors or markets, as well as incorporating additional financial indicators or external information sources to further improve forecasting performance. Overall, the proposed BO-optimized LSTM framework provides a practical and reliable approach for improving stock price forecasting models and contributes to the development of more robust AI-driven financial analytics systems.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Brawijaya, for providing research facilities and academic support during the completion of this study.

REFERENCES

[1] A. Sarkar and G. Vadivu, “An Advanced Ensemble Deep Learning Framework for Stock Price Prediction Using VAE, Transformer, and LSTM Model,” 2025.

[2] Z. Hu, Y. Zhao, and M. Khushi, “A survey of forex and stock price prediction using deep learning,” *Applied System Innovation*, vol. 4, no. 1, pp. 1–30, 2021, doi: 10.3390/ASI4010009.

[3] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018, doi: 10.1016/j.ejor.2017.11.054.

[4] S. Deviana, Nusyirwan, D. Azis, and P. Ferdias, “Analisis Model Autoregressive Integrated Moving Average Data Deret Waktu Dengan Metode Momen Sebagai Estimasi Parameter,” *Jurnal Siger Matematika*, vol. 02, no. 02, pp. 57–67, 2021.

[5] A. Gifty and Dr. Yang Li, “A Comparative Analysis of LSTM, ARIMA, XGBoost Algorithms in Predicting Stock Price Direction,” *Engineering and Technology Journal*, vol. 09, no. 08, pp. 4978–4986, 2024, doi: 10.47191/etj/v9i08.50.

[6] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[7] Z. Zhang, *Comparison of LSTM and ARIMA in Price Forecasting: Evidence from Five Indexes*, no. Esfct 2023. Atlantis Press International BV, 2024. doi: 10.2991/978-94-6463-268-2_6.

[8] B. Gautam, S. Kandel, M. Shrestha, and S. Thakur, “Comparative Analysis of Machine Learning Models for Stock Price Prediction: Leveraging LSTM for Real-Time Forecasting,” *Journal of Computer and Communications*, vol. 12, no. 08, pp. 52–80, 2024, doi: 10.4236/jcc.2024.128004.

[9] S. Mehtab and J. Sen, “Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models,” *2020 International Conference on Decision Aid Sciences and Application, DASA 2020*, pp. 447–453, 2020, doi: 10.1109/DASA51403.2020.9317207.

[10] S. Mehtab, J. Sen, and A. Dutta, “Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models | Papers With Code”.

[11] A. Flavia and C. Mio, “Performance Comparison of Standard LSTM and LSTM with Random Search Optimization for Spark New Zealand Limited Stock Price Prediction,” *International Journal Artificial Intelligent and Informatics*, vol. 3, no. 2, pp. 60–66, 2025, doi: 10.33292/ijarlit.v3i2.48.

[12] A. Parkin, M. Herrera, and D. A. Coley, “Energy or carbon? Exploring the relative size of universal zero carbon and zero energy design spaces,” *Building Services Engineering Research and Technology*, vol. 40, no. 3, pp. 319–339, May 2019, doi: 10.1177/0143624418815780.

[13] T. Yu and H. Zhu, “Hyper-Parameter Optimization: A Review of Algorithms and Applications,” pp. 1–56, 2020.

[14] P. I. Frazier, “A Tutorial on Bayesian Optimization,” no. Section 5, pp. 1–22, 2018.

[15] B. Dinda, “Gated recurrent neural network with TPE Bayesian optimization for enhancing stock index prediction accuracy,” 2022.

[16] B. K. Chan, O. V. Johnson, X. Chew, K. W. Khaw, M. H. Lee, and A. Alnoor, “Proposed Bayesian Optimization Based LSTM-CNN Model for Stock Trend Prediction,” *Computing and Informatics*, vol. 43, no. 1, pp. 38–63, 2024, doi: 10.31577/cai_2024_1_38.

[17] W. Huang, Y. Nakamori, and S. Wang, “Stock prediction based on Bayesian-LSTM,” in *International Conference on Machine Learning and Cybernetics*, 2018. doi: 10.1109/ICMLC.2018.8527025.

[18] B. K. Chan, O. V Johnson, X. Chew, K. W. Khaw, M. H. Lee, and A. Alnoor, “Proposed Bayesian Optimization Based LSTM-CNN Model for Stock Trend Prediction,” *Computing and Informatics*, vol. 43, no. 1, pp. 38–63, 2024, doi: 10.31577/cai_2024_1_38.

[19] K. Chen, Y. Zhao, and F. Dai, “A Critical Review of Recurrent Neural Networks for Sequence Learning,” in *2015 IEEE International Conference on Big Data (Big Data)*, Oct. 2015. [Online]. Available: http://arxiv.org/abs/1506.00019

[20] T. Yu and H. Zhu, “Hyper-parameter optimization: A review of algorithms and applications,” *arXiv preprint arXiv:2003.05689*, 2020.

[21] T. Fischer and C. Krauß, “Deep Learning With Long Short-Term Memory Networks for Financial Market Predictions,” *Eur. J. Oper. Res.*, 2018, doi: 10.1016/j.ejor.2017.11.054.

[22] A. Moghar and M. Hamiche, “Stock market prediction using LSTM recurrent neural network,” *Procedia Comput. Sci.*, vol. 170, pp. 1168–1173, 2020, doi: 10.1016/j.procs.2020.03.049.

[23] P. I. Frazier, “A Tutorial on Bayesian Optimization,” 2018. [Online]. Available: https://arxiv.org/abs/1807.02811

[24] M. Martínez-Ramón, M. Ajith, and A. Rajendra Kurup, *Deep Learning: A Practical Introduction.*, 1st ed. Chichester.

[25] R. Garnett, *Bayesian Optimization*. Cambridge, UK: Cambridge University Press, 2023. doi: 10.1017/9781108348973.

[26] S. Anam, H. A. Bukhori, A. Maulana, M. I. Maulana, and H. Rasikhun, “An Enhanced Particle Swarm Optimization with Mutation for Mean-Value-at-Risk Portfolio Optimization in the Indonesian Banking Sector,” Jurnal Teknik Informatika (Jutif), vol. 6, no. 4, pp. 2648–2666, 2025, doi: 10.52436/1.jutif.2025.6.4.5191.

[27] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Comput. Sci.*, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.