

Multi-Level Stacked Ensemble Learning with Multi-Search Hyperparameter Optimization for Stock Price Forecasting

Imam Prasodjo^{1,*}, Suyanto Suyanto², Dian Indiyati³, Imelda Atastina⁴

¹Department of Data Science, School of Computing, Telkom University, Bandung, Indonesia

^{2,4}Department of Informatics, School of Computing, Telkom University, Bandung, Indonesia

³Department of Economics and Business, School of Management, Telkom University, Bandung, Indonesia

(Received: May 15, 2026; Revised: July 15, 2026; Accepted: September 12, 2026; Available online: October 1, 2026)

Abstract

Stock price forecasting remains a challenging task due to the nonlinear, dynamic, and highly volatile nature of financial markets. Traditional forecasting methods often struggle to capture complex relationships among market variables, resulting in limited predictive accuracy. This study proposes a Multi Level Stacked Ensemble Learning (MML-SEL) framework integrated with multiple hyperparameter optimization techniques to improve stock price forecasting performance. The proposed architecture combines heterogeneous machine learning models across several hierarchical stacking layers, including Random Forest, Stochastic Gradient Descent, Multilayer Perceptron, Long Short-Term Memory, eXtreme Gradient Boosting, Gradient Boosting, and Support Vector Regression. To further enhance model performance, three hyperparameter optimization methods—Grid Search, Random Search, and Bayes Search—are employed and compared. Historical stock market data consisting of Open, High, Low, Close, and Volume variables were collected from three different stock exchanges, represented by TLKM (Indonesia Stock Exchange), MSFT (NASDAQ), and BIDU (Hong Kong Stock Exchange). The dataset was chronologically divided into training and testing sets, followed by data preprocessing, feature scaling, model training, stacking integration, and performance evaluation. Forecasting accuracy was assessed using R-Squared (R^2), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The experimental results demonstrate that the proposed MML-SEL framework achieves strong predictive performance across all datasets. The best results were obtained using Grid Search optimization, producing an R^2 value of 0.9996, MAE of 0.0114, and RMSE of 0.0202 on the MSFT dataset. The findings indicate that hierarchical ensemble learning combined with systematic hyperparameter optimization can significantly improve forecasting accuracy and model stability. This study contributes a scalable and effective forecasting framework that can support data-driven investment analysis and financial decision-making in diverse stock market environments.

Keywords: Stock Price Forecasting, Ensemble Learning, Stacked Ensemble, Hyperparameter Optimization, Machine Learning, Financial Prediction.

1. Introduction

Stock price forecasting has attracted significant attention from researchers and practitioners because of its crucial role in supporting investment decisions, risk management, and portfolio optimization. Accurate forecasting remains a challenging task due to the complex, nonlinear, and dynamic characteristics of financial markets. Stock prices are influenced by numerous interacting factors, including historical market behavior, trading volume, investor expectations, and market sentiment, making prediction a highly uncertain process [1]. Traditional statistical approaches, such as autoregressive models and linear regression techniques, often exhibit limited capability in capturing nonlinear dependencies and complex interactions among financial variables [2]. Recent advances in machine learning have transformed the field of financial forecasting by enabling the modeling of high-dimensional and nonlinear relationships that are difficult to capture using conventional methods. Algorithms such as Random Forest (RF), Support Vector Regression (SVR), eXtreme Gradient Boosting (XGBoost), Multilayer Perceptron (MLP), and Long Short-Term Memory (LSTM) networks have demonstrated promising performance in stock market prediction tasks [1], [2].

Despite these developments, no single machine learning model consistently outperforms others across different market conditions. Individual models often suffer from limitations related to overfitting, underfitting, model variance, and generalization capability, which may reduce forecasting reliability when applied to volatile financial environments. To address these limitations, ensemble learning has emerged as an effective strategy for improving predictive performance.

*Corresponding author: Dian Indiyati (dianindiyati@telkomuniversity.ac.id)

DOI: <https://doi.org/10.47738/jads.v7i4.1485>

This is an open access article under the CC-BY license (<https://creativecommons.org/licenses/by/4.0/>).

© Authors retain all copyrights

Ensemble methods combine multiple learning algorithms to generate predictions that are generally more accurate and robust than those produced by individual models [3]. Previous studies have shown that ensemble-based approaches can reduce forecasting errors and improve model stability in financial applications [4], [5].

Among various ensemble techniques, stacking has gained increasing attention because it allows heterogeneous predictive models to collaborate through hierarchical learning structures. Unlike bagging and boosting methods, stacking utilizes meta-learners to integrate predictions from multiple base learners, enabling the model to exploit complementary strengths from different algorithms [5], [6]. Although stacking-based forecasting frameworks have demonstrated encouraging results, most existing studies rely on relatively shallow architectures consisting of a single stacking layer. Such architectures may not fully capture the complexity of financial market behavior, particularly when multiple machine learning algorithms produce diverse predictive patterns.

Financial time-series data often contain hierarchical relationships and nonlinear interactions that may benefit from deeper ensemble structures. However, empirical research investigating multi-level stacked ensemble architectures remains relatively limited in the stock forecasting literature. Consequently, there is a need to explore whether deeper stacking hierarchies can improve predictive performance and enhance model robustness. Another critical factor influencing machine learning performance is hyperparameter optimization. Model performance is highly dependent on the selection of appropriate hyperparameter values, and suboptimal configurations may significantly degrade forecasting accuracy [7].

Various optimization approaches, including Grid Search, Random Search, and Bayesian Search, have been developed to identify optimal parameter combinations. Previous studies have demonstrated that systematic hyperparameter optimization can improve predictive accuracy, convergence efficiency, and model stability [8], [9]. Nevertheless, most stock forecasting studies evaluate optimization techniques separately from ensemble architectures, leaving limited evidence regarding their combined effectiveness within a hierarchical stacking framework. Based on these observations, a significant research gap exists in the integration of multi-level stacked ensemble learning and systematic hyperparameter optimization for stock price forecasting. While ensemble learning and optimization techniques have been extensively studied independently, their combined implementation within a hierarchical forecasting architecture has received comparatively less attention. Furthermore, many previous studies focus on a limited number of algorithms or employ relatively simple ensemble structures, potentially restricting predictive performance.

To address these limitations, this study proposes a Multi-Level Stacked Ensemble Learning (MML-SEL) framework integrated with multiple hyperparameter optimization techniques. The proposed architecture combines heterogeneous machine learning models, including Random Forest, Stochastic Gradient Descent (SGD), Multilayer Perceptron, Long Short-Term Memory, eXtreme Gradient Boosting, Gradient Boosting, and Support Vector Regression, through multiple hierarchical stacking layers. The proposed framework is evaluated using historical stock market data from three representative stocks: TLKM from the Indonesia Stock Exchange (IDX), MSFT from NASDAQ, and BIDU from the Hong Kong Stock Exchange (HKEX). Model performance is assessed using R-Squared (R^2), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). By integrating hierarchical ensemble learning with systematic hyperparameter optimization, this study aims to contribute a scalable forecasting framework capable of improving prediction accuracy and supporting more reliable investment decision-making. The contributions of this study are threefold. First, it proposes a hierarchical multi-level stacked ensemble architecture that integrates multiple heterogeneous machine learning algorithms. Second, it evaluates the effectiveness of three widely used hyperparameter optimization techniques within the proposed framework. Third, it provides empirical evidence regarding the capability of combining ensemble learning and hyperparameter optimization to improve stock price forecasting performance across multiple international stock market datasets.

2. Literature Review

2.1. Machine Learning-Based Stock Price Forecasting

Stock price forecasting has long been a major topic in financial research because of its importance in supporting investment decisions, portfolio management, and risk assessment. Predicting stock prices is inherently difficult due to the dynamic and uncertain nature of financial markets. Price movements are influenced by various factors, including historical trends, trading volume, investor behavior, and market conditions, which often interact in complex and nonlinear ways. As a result, traditional forecasting approaches frequently encounter limitations when attempting to model the complexity of financial data and generate accurate predictions. In recent years, machine learning has emerged as a promising alternative for stock price forecasting. Unlike conventional statistical methods, machine learning

algorithms are capable of identifying hidden patterns and learning complex relationships directly from large datasets. This capability allows machine learning models to process financial information more effectively and adapt to changing market conditions. Consequently, researchers have increasingly adopted machine learning techniques to improve forecasting performance and provide more reliable predictions in financial markets.

Various machine learning algorithms have been applied to stock market prediction, each offering unique strengths. Random Forest is known for its ability to reduce overfitting and handle high-dimensional data, while Support Vector Regression performs well in modeling nonlinear relationships. Similarly, eXtreme Gradient Boosting has gained popularity because of its strong predictive performance and computational efficiency. Deep learning approaches, particularly Long Short-Term Memory (LSTM) networks, have also demonstrated remarkable success in forecasting stock prices by capturing temporal dependencies within financial time-series data [2]. These developments indicate that machine learning has significantly enhanced the capability of forecasting models compared to traditional approaches.

2.2. Ensemble Learning and Multi-Level Stacking Architecture

The increasing complexity of financial markets has encouraged researchers to move beyond single-model forecasting approaches and adopt ensemble learning techniques. Ensemble learning refers to the integration of multiple predictive models to generate a more accurate and robust prediction than any individual model can achieve independently [3]. In stock market forecasting, where data are often noisy, nonlinear, and highly volatile, ensemble methods have demonstrated superior performance by combining the strengths of different algorithms while reducing their individual weaknesses. As a result, ensemble learning has become one of the most widely adopted approaches in modern financial prediction studies [6], [10].

Ensemble machine learning techniques combined with feature engineering significantly improved stock trend prediction accuracy consistently outperformed standalone models in high-frequency trading environments due to their ability to improve prediction stability and reduce forecasting errors offer a promising solution for addressing the inherent uncertainty and complexity of stock market behavior [4], [5].

Among various ensemble methods, stacking has attracted considerable attention because of its capability to combine heterogeneous learning algorithms within a hierarchical framework. Unlike bagging and boosting techniques, which focus on aggregating or sequentially improving predictions, stacking introduces a meta-learning layer that learns how to optimally combine the outputs generated by multiple base learners. This architecture enables different algorithms to contribute complementary information to the forecasting process. Previous studies have shown that stacking-based frameworks can effectively leverage the strengths of both conventional machine learning and deep learning models, resulting in improved predictive performance [11], [12], [13].

Despite the success of stacking approaches, most existing studies employ relatively shallow architectures consisting of only a single meta-learning layer. While such architectures have demonstrated promising results, they may not fully capture the complex patterns embedded within financial time-series data. Recent developments in artificial intelligence suggest that deeper ensemble structures can provide additional predictive advantages by progressively refining prediction outputs through multiple learning stages. Building upon this perspective, multi-level stacking architecture extends traditional stacking by incorporating several hierarchical ensemble layers, where the outputs of one layer become inputs for subsequent layers. Such an architecture enables information to be processed and refined progressively, allowing the model to capture more sophisticated nonlinear relationships and interactions among financial variables. Given the increasing complexity of stock market dynamics, multi-level stacking architectures provide a promising framework for improving forecasting accuracy, enhancing model robustness, and generating more reliable predictions.

2.3. Hyperparameter Optimization

Hyperparameter optimization has become an essential component of modern machine learning because the performance of predictive models is highly dependent on the selection of appropriate hyperparameter values. Unlike model parameters, which are learned automatically during the training process, hyperparameters must be determined before model development and directly influence how the learning algorithm operates. Examples include the number of trees in Random Forest, learning rates in boosting algorithms, kernel functions in Support Vector Regression, and the number of hidden units in neural networks. Inadequate hyperparameter settings can lead to underfitting, overfitting, poor generalization, and reduced forecasting accuracy. Therefore, optimizing hyperparameters has become a critical step in developing reliable machine learning models, particularly in complex domains such as financial forecasting [7]

The growing complexity of financial time-series data has further increased the importance of hyperparameter optimization. Stock markets exhibit nonlinear behavior, high volatility, and dynamic patterns that require machine learning algorithms to operate under optimal configurations. Previous studies have demonstrated that even highly sophisticated models may produce unsatisfactory results when hyperparameters are not properly tuned. Consequently, researchers have increasingly emphasized the integration of systematic optimization procedures to enhance predictive accuracy and model stability [14], [15], [16]. Effective hyperparameter optimization enables machine learning models to better capture underlying market patterns while reducing prediction errors. Several optimization techniques have been developed to identify optimal hyperparameter configurations. Grid Search is one of the earliest and most widely used approaches because it systematically evaluates all possible combinations within a predefined search space. Although Grid Search can identify highly effective parameter settings, its exhaustive nature often requires substantial computational resources, particularly when dealing with large datasets and complex models. To overcome this limitation, Random Search was introduced as a more computationally efficient alternative. Rather than evaluating every possible combination, Random Search explores randomly selected configurations and has been shown to achieve competitive performance while significantly reducing computational cost.

More recently, advanced optimization approaches have emerged to further improve efficiency and predictive performance. Bayesian Search employs probabilistic modeling to guide the search process toward promising regions of the parameter space, thereby reducing the number of evaluations required to identify optimal configurations. In the context of financial forecasting, several studies have reported substantial performance improvements through advanced optimization techniques. a hybrid deep learning model optimized through the Golden Jackal Optimizer and reported superior forecasting accuracy compared to baseline models [8], [9], [17], [18]. Despite the growing body of literature on hyperparameter optimization, most studies focus on optimizing individual machine learning models rather than investigating optimization strategies within hierarchical ensemble frameworks. Furthermore, limited research has compared multiple optimization techniques in the context of multi-level stacked ensemble architectures for stock price forecasting. This gap is particularly important because the performance of ensemble models depends not only on the diversity of learners but also on the quality of their individual configurations. Therefore, integrating Grid Search, Random Search, and Bayes Search within Multi-Level Stacked Ensemble Learning (MML-SEL) framework offers an opportunity to evaluate the combined effects of hierarchical ensemble learning and systematic hyperparameter optimization on forecasting accuracy and model robustness.

2.4. Integration of Multi-Level Stacking and Hyperparameter Optimization

The rapid advancement of machine learning has led researchers to explore not only individual predictive algorithms but also integrated frameworks that combine multiple learning strategies to improve forecasting performance. Among these developments, ensemble learning and hyperparameter optimization have emerged as two of the most influential approaches in predictive analytics. Ensemble learning enhances prediction accuracy by leveraging the diversity of multiple models, whereas hyperparameter optimization improves model effectiveness by identifying the most suitable parameter configurations. Although both approaches have independently demonstrated strong predictive capabilities, their combined application within a unified forecasting framework remains relatively underexplored, particularly in stock price prediction.

Multi-level stacking architectures have shown considerable potential in addressing the limitations of single-model and shallow ensemble approaches [3], [19], [20]. Previous studies have reported that ensemble-based frameworks generally outperform standalone machine learning models because they reduce prediction variance and improve generalization capability [21], [22] that stacking ensemble structures can significantly improve predictive performance by integrating diverse learning mechanisms within a hierarchical architecture. These findings suggest that deeper stacking structures may provide greater predictive robustness when applied to complex financial datasets. At the same time, numerous studies have highlighted the importance of hyperparameter optimization in maximizing machine learning performance. Model effectiveness is highly sensitive to hyperparameter settings and that optimization procedures are essential for achieving reliable results [7], [23], [24]. Furthermore, automated optimization frameworks could enhance both predictive performance and computational efficiency. These studies collectively indicate that hyperparameter optimization plays a critical role in unlocking the full potential of machine learning models.

Despite these advances, existing research has largely examined ensemble learning and hyperparameter optimization as separate research streams. Most studies focus either on developing sophisticated ensemble architectures or on improving individual models through optimization techniques. Consequently, limited attention has been given to understanding how hyperparameter optimization can enhance the effectiveness of hierarchical ensemble frameworks.

This limitation is particularly important because the performance of ensemble systems depends not only on the diversity of constituent models but also on the quality of each model's configuration. Without proper optimization, even advanced ensemble architectures may fail to achieve their maximum predictive potential. The integration of Multi-Level Stacked Ensemble Learning and Hyperparameter Optimization offers a promising solution to this challenge. By combining hierarchical ensemble structures with systematic optimization strategies such as Grid Search, Random Search, and Bayes Search, forecasting models can simultaneously benefit from model diversity and parameter efficiency. Multi-level stacking enables the framework to capture complex nonlinear relationships across different learning layers, while hyperparameter optimization ensures that each learner operates under near-optimal conditions.

3. Methodology

Several crucial phases are covered by the MML-SEL Model research approach, which aims to increase the precision of stock price forecasts. In order to create reliable configuration designs, this methodical approach focuses on using a variety of machine learning approaches, data sources, assessment frameworks, and hyperparameter tuning.

One of the crucial phases in stock price forecasting research is the compilation of data sets. Analyzing how accurate stock price forecasting are is one of them. Each stock has six variables in the data set: trading volume, transaction date, open, high, low, and closed price. Five phases make up the data set construction procedure employed in this study.

Gathering stock transaction data from Google Finance is the first stage. Preprocessing the gathered data is the second phase. The regression model was tested on several training datasets in the third phase using Random Forest (RF), Stochastic Gradient Descent (SGD), Long Short-Term Memory (LSTM), Multilayer Perceptron (MLP), eXtreme Gradient Boosting (XGB), Support Vector Regression (SVR), and Gradient Boosting (GB). In the fourth phase, the model is evaluated using the R-Squared (R^2), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) techniques. Finally, we modify the Hyperparameter in the fifth step using Grid Search (GS), Random Search (RS), and Bayes Search (BS).

3.1. Data Collection

The first phase entails gathering a lot of data. Data on past stock prices, such as closing price and daily trading volume, should be gathered from trustworthy sources. This data approach ensures a rich set of data that can capture the influence on stock price movements. Table 1 displays data for this time period that shows the movement of stock trading transactions on the Nasdaq, Hong Kong Stock Exchange (HKEX), and Indonesia Stock Exchange (IDX).

Table 1. Data Description

Item	Description
Period	January 1, 2010 to September 30, 2025
Indonesia Stock	TLKM (IDX)
US Stock	MSFT (Nasdaq)
China Stock	BIDU (HKEX)
Variables	Date of Transaction, Open, High, Low, Close, and Volume
Number of Observations	11,711

3.2. Data Preprocessing

Following collection, the data is initially cleaned up, which includes identifying missing numbers, eliminating noise and outliers, smoothing out irregular data, and looking for data inconsistencies. After then, the data is categorized by type.

3.2.1. Time-Series Data Splitting Strategy

Unlike conventional machine learning tasks, stock price forecasting involves sequential data in which temporal ordering must be preserved. Randomly shuffling observations prior to model training may introduce look-ahead bias, allowing future information to influence the learning process and resulting in overly optimistic performance estimates.

To maintain the integrity of the forecasting task, the dataset was partitioned chronologically into training and testing subsets. Specifically, the earliest 85% of observations were allocated to the training set, while the most recent 15% were reserved exclusively for testing. No random sampling or shuffling procedures were applied during data partitioning. This approach ensures that the model is trained using only historical information and evaluated on unseen

future observations, thereby providing a more realistic assessment of forecasting performance. The chronological train-test split is consistent with established practices in financial time-series forecasting, where preserving temporal dependencies is essential for avoiding information leakage and maintaining the validity of predictive evaluation.

3.2.2. Experimental Configuration and Reproducibility Settings

To improve reproducibility and transparency, all experiments were conducted using a consistent computational environment and predefined model configurations. The proposed Multi-Level Stacked Ensemble Learning framework was implemented in Python using established machine learning and deep learning libraries. Hyperparameter optimization was performed using Grid Search, Random Search, and Bayes Search to identify the best-performing parameter combinations for each learning algorithm.

The training process followed identical preprocessing, feature generation, and evaluation procedures across all datasets. The dataset was chronologically divided into training (85%) and testing (15%) subsets to preserve temporal ordering and prevent information leakage. Model performance was evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Coefficient of Determination (R^2).

3.3. Stacking Ensemble Method

The MML-SEL model is visualized in figure 1. Proposed Methodology System. Testing models on some of our training datasets, namely stock data from TLKM (Telekomunikasi Indonesia Tbk), MSFT (Microsoft Corporation) and BIDU (Baidu Inc.), using:

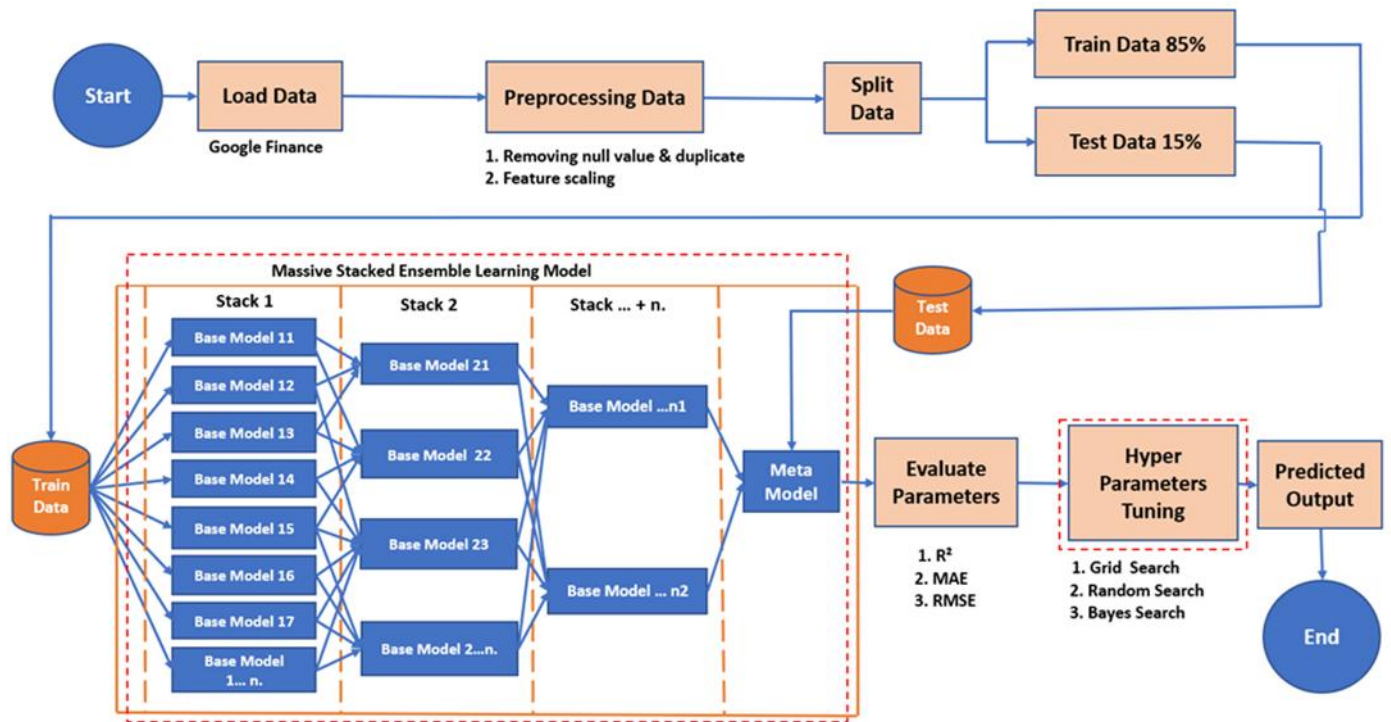


Figure 1. Proposed Methodology System

3.3.1. RF (Random Forest)

RF a group approach that builds numerous decision trees while training. The mean prediction of each individual tree serves as the final prediction for regression tasks, minimizing variance and avoiding overfitting. This algorithm creates a more accurate and stable model by using random feature selection and bagging techniques. The output class is chosen by majority voting, in which each decision tree casts a vote for a certain class; the class with the most votes becomes the final forecast. Able to handle large and complex datasets, reducing the risk of overfitting and providing high accuracy in classification, calculated with the formula:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^B T_b(x) \quad (1)$$

3.3.2. XGB (eXtreme Gradient Boosting)

XGB is a machine learning approach that builds hundreds or thousands of decision trees sequentially in order to achieve better speed and accuracy. The goal of every new tree that is produced is to fix the errors that the ones that came before it, calculated with the formula:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F} \quad (2)$$

3.3.3. SVR (Support Vector Regression)

SVR is a regression variant of the widely used Support Vector Machine (SVM) classification algorithm. Finding the lines that best match the data while maintaining a specific level of fault tolerance is the aim of the SVR if the SVM's objective is to identify the lines that best divide the data into classes, calculated with the formula:

$$f(x) = \langle w, x \rangle + b \quad (3)$$

3.3.4. SGD (Stochastic Gradient Descent)

SGD Machine learning models are trained using the optimization process known as stochastic gradient descent. Finding the ideal model parameters (weights) involves repeatedly modifying them little by little until the smallest errors or prediction errors are achieved, calculated with the formula:

$$\theta = \theta - \eta \cdot \nabla_{\theta J(\theta; x^{(i)}, y^{(i)})} \quad (4)$$

3.3.5. MLP (Multilayer Perceptron)

One of the most basic forms of artificial neural network architecture is the multilayer perceptron. It is a simple deep learning model that recognizes intricate patterns in the data by connecting multiple layers of neurons, calculated with the formula:

$$\mathbf{a}^{(l)} = f(\mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}) \quad (5)$$

3.3.6. LSTM (Long Short-Term Memory)

One kind of sophisticated Recurrent Neural Network (RNN) architecture called (Long Short-Term Memory) is made especially to remember information for an extended amount of time. Because of this, it is very dependable for processing sequential data, including time series data, text, and speech, calculated with the formula:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (7)$$

$$\{C\}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (8)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \quad (9)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (10)$$

$$h_t = o_t \odot \tanh(C_t) \quad (11)$$

3.3.7. GB (Gradient Boosting)

GB is a machine learning strategy that combines multiple weak prediction models into a single, stronger model to increase prediction accuracy. This method is renowned for its capacity to handle complex data, especially data with noise, and it can yield excellent accuracy, calculated with the formula:

$$F_{M(x)} = F_{0(x)} + \sum_{m=1}^M h_{m(x)} \quad (12)$$

3.4. Evaluation Model

Some of our training datasets whose analysis results were acquired utilizing the RF, XGB, SVR, SGD, MLP, LSTM, and GB Stacked Ensemble Models were evaluated during this evaluation step. The R^2 , MAE, and RMSE methods were employed.

3.4.1. R-squared (R^2)

The percentage of the dependent variable's variance that the independent variables in the regression model can account for is measured by the coefficient of determination. Higher R^2 values show that the model is more effective in explaining data variability. R^2 values range from 0 to 1. The formula used to compute R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (13)$$

3.4.2. Mean Absolute Error (MAE)

The average absolute difference between the expected and actual numbers is measured by the {Mean Absolute Error (MAE)}. Without accounting for the error's direction, MAE provides a clear image of the size of the forecast error. The model performs better when the MAE value is less. The formula for MAE is:

$$AE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (14)$$

3.4.3. Root Mean Squared Error (RMSE)

Measure that illustrates how distant the projected value is from the actual value by computing the square root of the average square of the difference between the two. RMSE lends more weight to larger errors since the errors are squared before being averaged. The model's quality increases with a decreased RMSE value. The formula for RMSE is:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (15)$$

3.5. Tuning Hyperparameters

In machine learning or deep learning, the following procedure is used to determine or set a model's optimal value.

3.5.1. Grid Search (GS)

Is a technique for determining a machine learning model's optimal hyperparameter values. In order to determine which combination of the given hyperparameters yields the greatest model performance, we will work in a very thorough and scientific manner, trying every potential combination. The formula for GS is:

$$G = \Lambda_1 \times \Lambda_2 \times \dots \times \Lambda_p \quad (16)$$

$$\theta_{\text{best}} = \arg \max_{\theta^* \in G} \mathcal{S}(A_{\theta^*}, D_{\text{val}}) \quad (17)$$

3.5.2. Random Search (RS)

Is a method to find the best hyperparameter setting for a machine learning model by trying random combinations and testing only a randomly selected number of combinations from a given search space. The RS formula is:

$$\theta^{(i)} \sim \mathcal{D}, \quad \text{for } i = 1, 2, \dots, n \quad (18)$$

$$\theta_{\text{best}} = \arg \max_{\theta^{(i)} \in \{\theta^{(1)}, \dots, \theta^{(n)}\}} \mathcal{S}(A_{\theta^{(i)}}, D_{\text{val}}) \quad (19)$$

3.5.3. Bayes Search (BS)

Is a smart method for hyperparameter optimization that learns from each experiment to make better guesses in the next experiment and uses probabilities to guide its search to the most promising areas. The BS formula is:

$$P(A|B) \propto P(B|A) \cdot P(A) \quad (20)$$

$$\hat{y}_{\text{final}} = M_{\text{meta}}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N) \quad (21)$$

4. Results and Discussion

4.1. Results

The following is an in-depth explanation the analysis with a MML-SEL structure that uses a specific combination of algorithms at each level. The process configuration model contrasted in [figure 2](#). The Base Layer (Stack 1) uses five base models, namely Random Forest, SGD Regressor (used twice as m2 and m4), LSTM, and MLP Regressor. Intermediate Layer Stack 2, the results of the base layer are further processed using two units of XGBoost (A and B) and one unit of Gradient Boosting (C) and Intermediate Layer Stack 3 using SVR (X). Final Model (Meta Model) Using a Random Forest Regressor as the final decision maker to minimize residual prediction errors.

4.1.1. Relationship Between Figure 1 and Figure 2

[Figure 1](#) presents the conceptual architecture of the proposed Multi-Level Stacked Ensemble Learning (MML-SEL) framework. It illustrates the overall workflow, including data pre processing, hierarchical ensemble construction, hyperparameter optimization, model training, and forecasting evaluation. The figure is intended to provide a high-level representation of the proposed forecasting methodology.

[Figure 2](#) operationalizes the conceptual framework presented in [figure 1](#) by specifying the actual machine learning models used at each stacking level. In the implementation stage, Random Forest (RF), Stochastic Gradient Descent (SGD), Multi-Layer Perceptron (MLP), and Long Short-Term Memory (LSTM) serve as first-level learners. Their outputs are subsequently processed by XGBoost (XGB) and Gradient Boosting (GB) models in the second stacking layer, followed by Support Vector Regression (SVR) in the third layer. Finally, a Random Forest meta-learner generates the final stock price prediction. Therefore, [figure 1](#) represents the conceptual design of the proposed framework, whereas [figure 2](#) illustrates its specific implementation and experimental configuration. The two figures should be interpreted as complementary components of the same forecasting architecture rather than independent frameworks.

```
{'layers': ['stack1', 'stack2', 'stack3', 'meta'],
 'models': {'stack1': {'m1': {'model': 'RandomForestRegressor',
                               'm2': {'model': 'SGDRegressor',
                               'm3': {'model': 'LSTM',
                               'm4': {'model': 'SGDRegressor',
                               'm5': {'model': 'MLPRegressor',
                               'stack2': {'A': {'model': 'XGBRegressor',
                               'B': {'model': 'XGBRegressor',
                               'C': {'model': 'GradientBoostingRegressor',
                               'stack3': {'X': {'model': 'SVR',
                               'meta': {'final': {'model': 'RandomForestRegressor',
                               'mappings': {'stack2': {'A': ['m1', 'm2'],
                               'B': ['m2', 'm3'],
                               'C': ['m4', 'm5']},
                               'stack3': {'X': ['A', 'B', 'C']},
                               'meta': {'final': ['X']}}}}
```

Figure 2. Process Configuration Model

The description of the results of the research on the MML-SEL model for stock price forecasting is This study shows that the proposed stacked ensemble model has very high performance and surpasses single models and previous studies. The average results of the evaluation metrics achieved are as follows R-Squared (R2) 0.9996, Mean Absolute Error (MAE) 0.0114, Root Mean Square Error (RMSE) 0.0202. The model uses a tiered stacking architecture that combines various machine learning algorithms to capture complex data patterns and reduce the risk of overfitting. The algorithms used include Base Learners Stochastic Gradient Descent (SGD), Multilayer Perceptron (MLP), Random Forest (RF), Support Vector Regression (SVR), eXtreme Gradient Boosting (XGB), and Gradient Boosting (GB). Meta-Models are used to combine forecasting from basic models to minimize final prediction errors. This study proves that the use of Hyperparameter Tuning consistently improves model accuracy compared to models without optimization. The methods used are Grid Search (GS), Random Search (RS), and Bayes Search (BS). Best Results The Grid Search (GS) method proved to be the most successful in producing the lowest MAE and RMSE values for most of the stock codes tested.

The reliability of the model has been tested using historical data from various international stock exchanges to ensure its effectiveness, including the Indonesia Stock Exchange (IDX), TLKM Shares (Telekomunikasi Indonesia), Nasdaq (US): MSFT Shares (Microsoft), Hong Kong Stock Exchange (HKEX): BIDU Shares (Baidu).

The study process model proposed MML-SEL is described in [figure 1](#). The research uses the results of a historical stock transaction dataset, which only uses the status "close" to carry out the prediction process; features for the results include the statuses "Open," "High," "Low," and "Volume." There are two parts to the data set: training data (0.85) and testing data (0.15). The data is divided into train and test data in order to determine the optimal accuracy value. The use of standard scaling for feature scaling ensures that all features in the dataset are at the same scale, which is crucial for machine learning algorithms that are sensitive to data scale. Additionally, it was used to merge the steps into one. The actual and anticipated stock values of the TLKM, MSFT & BIDU stock code's are contrasted in [figure 3](#). The scatter plot actual and anticipated stock values of the TLKM, MSFT & BIDU stock code's are contrasted in [figure 4](#). The R^2 , MAE & RMSE Comparison Result are contrasted in [figure 5](#). The results of the research can be seen in [table 2](#).

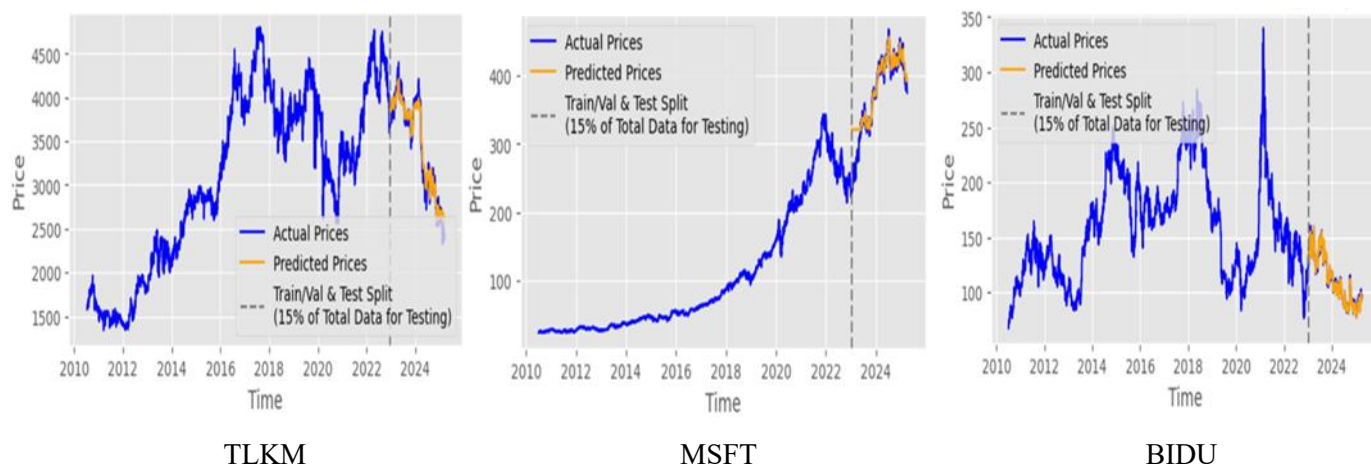


Figure 3. Actual vs Forecast Stock Price Result

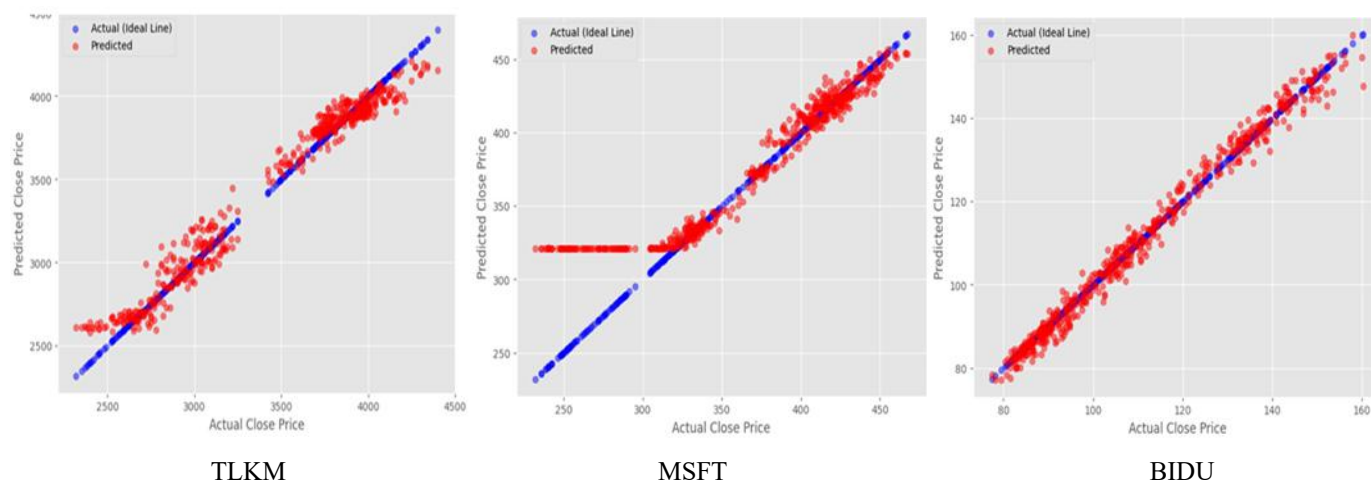


Figure 4. Scatter Plot Actual vs Forecast Stock Price Result

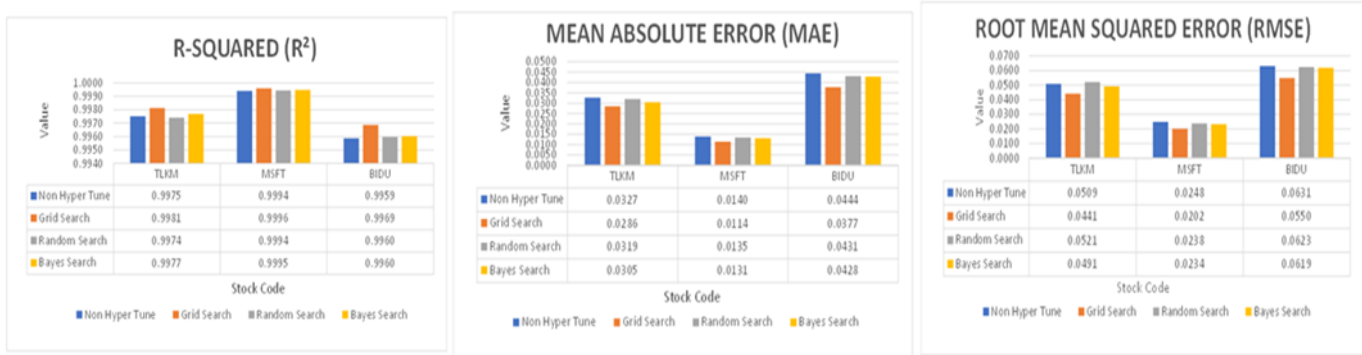


Figure 5. R^2 , MAE & RMSE Comparison Result

Table 2 presents the stock price forecasting results for the TLKM, MSFT, and BIDU stock codes using four hyperparameter-tuning approaches, namely NH, GS, RS, and BS. The performance is evaluated using three metrics: R^2 , MAE, and RMSE. Overall, all approaches produced high R^2 values, ranging from 0.9959 to 0.9995, indicating a strong agreement between the predicted and actual stock prices. For TLKM, the highest R^2 value was obtained using RS at 0.9981, while GS achieved the lowest MAE of 0.0286 and RMSE of 0.0441. For MSFT, BS produced the highest R^2 value of 0.9995 and the lowest MAE and RMSE values of 0.0131 and 0.0234, respectively. For BIDU, GS achieved the highest R^2 value of 0.9969 and the lowest MAE and RMSE values of 0.0377 and 0.0550, respectively.

Table 2. Table Stock Price Forecast Result

Stock Code	HT	R2	MAE	RMSE
TLKM	NH	0.9975	0.0327	0.0509
	GS	0.9974	0.0286	0.0441
	RS	0.9981	0.0319	0.0521
	BS	0.9977	0.0305	0.0491
MSFT	NH	0.9994	0.0140	0.0248
	GS	0.9994	0.0140	0.0248
	RS	0.9981	0.0319	0.0521
	BS	0.9995	0.0131	0.0234
BIDU	NH	0.9959	0.0444	0.0631
	GS	0.9969	0.0377	0.0550
	RS	0.9960	0.0431	0.0623
	BS	0.9960	0.0428	0.0619

Note: HT → Hyperparameter Tuning, NH → Non Hyperparameter Tuning, GS → Grid Search, RS → Random Search, BS → Bayes Search

As summarized in table 2, the results show that the performance of the hyperparameter-tuning approaches varies across the three stock codes. The MSFT forecasting results generally show the smallest prediction errors, with MAE values between 0.0131 and 0.0140 and RMSE values between 0.0234 and 0.0248. In comparison, BIDU exhibits relatively higher prediction errors, while TLKM produces intermediate error values. These differences indicate that the forecasting performance is influenced by the characteristics of each stock dataset. The consistently high R^2 values, together with relatively low MAE and RMSE values, demonstrate that the proposed MML-SEL model can closely approximate the observed stock prices across the evaluated

4.2. Ensemble Learning and Hyperparameter Optimization for Stock Price Forecasting

4.2.1. The Role of Ensemble Learning in Stock Price Forecasting

Ensemble learning has become one of the most effective approaches for stock price forecasting because it combines the strengths of multiple predictive models while reducing the limitations of individual learners. Financial markets exhibit complex nonlinear patterns, high volatility, and dynamic interactions among market variables, making accurate forecasting particularly challenging. By integrating diverse learning algorithms, ensemble methods can improve predictive stability, reduce model variance, and enhance generalization performance.

The effectiveness of stacking lies in its hierarchical learning mechanism. Rather than relying on a single algorithm, stacking enables multiple base learners to capture different characteristics of financial data. The outputs of these learners are subsequently combined through intermediate and meta-learning layers, allowing the framework to exploit complementary predictive information. This hierarchical structure is particularly beneficial for stock price forecasting, where market behavior is influenced by complex and evolving patterns. The results obtained in this study support these findings. The proposed Multi-Level Stacked Ensemble Learning framework integrates Random Forest, SGD, MLP, LSTM, XGBoost, Gradient Boosting, and Support Vector Regression within a hierarchical architecture. Experimental results show that the ensemble framework achieves high predictive accuracy across TLKM, MSFT, and BIDU datasets, outperforming non-optimized configurations. The combination of diverse learning mechanisms enables the model to capture both linear and nonlinear relationships present in historical market data. These findings suggest that forecasting performance can be significantly enhanced through the integration of multiple learning algorithms within a structured stacking architecture. The proposed framework demonstrates that ensemble diversity and hierarchical learning contribute substantially to prediction accuracy and model robustness, making Multi-Level Stacked Ensemble Learning a promising approach for stock price forecasting applications.

Recent studies consistently demonstrate that ensemble learning outperforms single-model approaches in stock price forecasting because it reduces prediction variance and improves generalization capability. Ensemble architectures combine multiple learners with different predictive characteristics, enabling the model to capture complex nonlinear relationships commonly observed in financial time-series data. Previous reviews have identified ensemble learning as one of the most promising directions for financial forecasting due to its ability to integrate heterogeneous learning mechanisms and improve predictive stability across different market conditions. Although ensemble methods generally outperform standalone algorithms, significant differences exist among ensemble architectures. Traditional ensemble approaches such as bagging and boosting primarily focus on reducing variance or bias at a single learning level. For example, Random Forest improves stability through bootstrap aggregation, while Gradient Boosting and XGBoost sequentially correct prediction errors generated by preceding learners. However, these methods rely on a relatively homogeneous learning structure and may be limited when modeling highly complex market dynamics.

To address these limitations, recent studies have explored multi-stage and stacked ensemble frameworks that a two-stage ensemble architecture improves forecasting accuracy by combining signal decomposition and machine learning models [5], [25] and integrating feature engineering with ensemble learning significantly enhances stock trend prediction. These findings suggest that hierarchical integration of diverse learners can generate more robust predictive performance than conventional ensemble approaches [3], [26].

Despite these advances, most existing studies focus either on optimizing individual models or constructing shallow ensemble architectures. Research investigating deeper hierarchical stacking frameworks remains limited. Furthermore, previous studies rarely examine how different hyperparameter optimization strategies influence the performance of multi-level ensemble systems. As a result, it remains unclear whether improvements in forecasting accuracy originate primarily from ensemble diversity, hyperparameter optimization, or the interaction between both mechanisms.

This research addresses this gap by integrating Multi-Level Stacked Ensemble Learning (MML-SEL) with multiple hyperparameter optimization techniques, namely Grid Search, Random Search, and Bayes Search. Unlike previous studies that evaluate optimization methods separately or apply them to individual learners, the proposed framework investigates their effectiveness within a hierarchical ensemble architecture. This approach enables a more comprehensive evaluation of how ensemble depth and hyperparameter optimization jointly contribute to stock price forecasting performance.

4.2.2. Hybrid Hyperparameter Optimization Techniques

The performance of machine learning models in stock price forecasting is highly dependent on appropriate hyperparameter selection. Hyperparameters control model behavior during training and significantly influence predictive accuracy, generalization capability, and computational efficiency. Financial time-series data are characterized by nonlinearity, volatility, and temporal dependencies, making hyperparameter optimization a critical component of forecasting model development.

Previous studies have demonstrated that model performance can vary substantially depending on parameter configurations of systematic hyperparameter optimization is essential for obtaining reliable machine learning results, particularly in complex prediction tasks. Similarly, recent stock forecasting studies have shown that optimized machine learning and deep learning models consistently outperform non-optimized configurations in terms of prediction accuracy and error reduction [7]. Among the most widely adopted optimization approaches are Grid Search, Random Search, and Bayesian Optimization. Grid Search performs an exhaustive exploration of predefined parameter combinations and often provides highly reliable solutions when the search space is manageable. Random Search improves computational efficiency by evaluating randomly selected parameter combinations, while Bayesian Optimization uses probabilistic modelling to guide the search toward promising regions of the parameter space, reducing the number of evaluations required.

The findings of this study support previous evidence regarding the importance of hyperparameter optimization. Across the three stock datasets evaluated, optimized models consistently achieved better predictive performance than non-optimized configurations. Among the evaluated approaches, Grid Search produced the most stable and accurate forecasting results, achieving the highest R^2 values and the lowest prediction errors in most experimental scenarios.

These findings suggest that the effectiveness of ensemble learning frameworks depends not only on model diversity but also on the quality of parameter configurations assigned to individual learners. Therefore, integrating multiple hyperparameter optimization strategies within a Multi-Level Stacked Ensemble Learning framework provides a systematic approach for improving forecasting accuracy and model robustness in stock price prediction.

4.3. Discussion of Predictive Performance and Data Leakage Prevention

The proposed Multi-Level Stacked Ensemble Learning framework achieved exceptionally high predictive performance, with R^2 values approaching 0.9996 across several experimental scenarios. While such results may appear unusually high in financial forecasting studies, several factors should be considered when interpreting these findings.

First, the forecasting framework was trained using historical market variables consisting of Open, High, Low, Close, and Volume (OHLCV) data. These variables exhibit strong temporal dependencies and inherent correlations, enabling machine learning models to learn recurring price movement patterns effectively. The integration of multiple learning algorithms through hierarchical stacking further enhances the model's capability to capture both linear and nonlinear relationships within the data.

Second, the evaluation procedure was designed to minimize the risk of information leakage. The dataset was divided chronologically into training and testing subsets, with the earliest 85% of observations used for model training and the most recent 15% reserved exclusively for testing. No random shuffling was applied during data partitioning to preserve temporal ordering and avoid look-ahead bias. Consequently, the model was evaluated only on unseen future observations.

Third, the observed performance gains may also be attributed to the combined effects of multi-level ensemble learning and systematic hyperparameter optimization. By integrating diverse learners and optimizing model configurations through Grid Search, Random Search, and Bayes Search, the framework is able to reduce prediction errors and improve generalization performance across different datasets.

Nevertheless, high predictive performance should be interpreted cautiously. Future research may employ additional validation procedures, such as walk-forward validation, rolling-window evaluation, and out-of-sample testing across broader market conditions, to further verify the robustness and generalizability of the proposed framework.

4.3.1. Benchmark Comparison with Previous Studies

To provide a broader perspective on the effectiveness of the proposed framework, the forecasting performance of the Multi-Level Stacked Ensemble Learning (MML-SEL) model was compared with results reported in previous stock forecasting studies. Although direct comparisons should be interpreted cautiously due to differences in datasets,

forecasting horizons, and evaluation protocols, such benchmarking provides useful insights into the relative performance of the proposed approach. The comparison indicates that the proposed MML-SEL framework achieves competitive predictive performance relative to existing machine learning, deep learning, and ensemble-based forecasting models. The integration of hierarchical stacking and multi-search hyperparameter optimization contributes to improved prediction accuracy and reduced forecasting errors across the evaluated datasets. Overall, the results suggest that combining heterogeneous learners within a multi-level ensemble architecture can provide substantial performance gains compared with conventional single-model and shallow ensemble approaches.

Although the proposed Multi-Level Stacked Ensemble Learning framework achieved strong forecasting performance across the evaluated datasets, the results should not be interpreted as evidence of robustness under crisis conditions or geopolitical shocks. The experimental design focused exclusively on predictive accuracy using historical market data and did not include stress testing, event-based analysis, regime-switching evaluation, or crisis-period validation. Consequently, the findings demonstrate the effectiveness of the proposed framework under the observed market conditions represented in the datasets but do not establish its resilience during periods of extreme market turbulence, financial crises, or major geopolitical disruptions. Additional empirical investigations are required to assess model stability and forecasting reliability under such conditions.

4.4. Studi Limitation

Another limitation of this study is the exclusive use of structured historical market data. Alternative information sources, including investor sentiment, financial news, social media content, and textual market disclosures, were not incorporated into the forecasting framework. As a result, the study does not evaluate the potential contribution of sentiment-based or textual features to stock price prediction performance. Future research may explore the integration of these alternative data sources within the proposed ensemble architecture. Several limitations should be acknowledged when interpreting the findings of this study. First, the empirical evaluation was conducted using only three representative stocks, namely TLKM, MSFT, and BIDU. Although these companies originate from different industries and market environments, the dataset may not fully capture the diversity of global financial markets. Consequently, the generalizability of the proposed Multi-Level Stacked Ensemble Learning framework to other sectors, market capitalizations, or geographic regions should be interpreted with caution. Second, the study relies exclusively on historical market variables, including Open, High, Low, Close, and Volume (OHLCV) data. External factors such as macroeconomic indicators, and firm-specific fundamental information were not incorporated into the forecasting process. Therefore, the reported predictive performance reflects the model's ability to learn patterns from historical trading data rather than its capability to capture broader market influences

Another limitation of this study is the absence of an ablation analysis. Although the proposed Multi-Level Stacked Ensemble Learning framework achieved strong predictive performance, the current experimental design does not isolate the individual contribution of each stacking layer, base learner, or hyperparameter optimization technique. Consequently, it is not possible to determine the relative importance of specific components within the overall architecture. Future research should conduct systematic ablation experiments by sequentially removing individual learners, stacking levels, and optimization strategies to quantify their respective contributions to forecasting performance. Such analyses would provide deeper insights into the mechanisms driving the effectiveness of the proposed framework and support more efficient model design. Finally, the experimental evaluation focuses on a limited number of stocks and a single forecasting setting. Future research should validate the proposed framework using larger and more diverse datasets, including stocks from multiple exchanges, sectors, and market conditions. Expanding the empirical scope would provide stronger evidence regarding the robustness, scalability, and generalizability of the proposed forecasting framework

5. Conclusion

This study proposed a Multi-Level Stacked Ensemble Learning (MML-SEL) framework integrated with multiple hyperparameter optimization techniques, including Grid Search, Random Search, and Bayes Search, for stock price forecasting. The framework combines heterogeneous machine learning and deep learning models through a hierarchical stacking architecture consisting of Random Forest, SGD, MLP, LSTM, XGBoost, Gradient Boosting, Support Vector Regression, and a Random Forest meta-learner.

Experimental evaluations conducted on TLKM, MSFT, and BIDU stock datasets demonstrate that the proposed framework achieves high forecasting accuracy and consistently outperforms non-optimized model configurations. The

results indicate that combining multi-level ensemble learning with systematic hyperparameter optimization can significantly improve predictive performance and reduce forecasting errors.

Among the evaluated optimization methods, Grid Search produced the most consistent performance across the tested datasets, while the hierarchical stacking mechanism effectively leveraged the complementary strengths of different learning algorithms. These findings highlight the importance of both ensemble diversity and hyperparameter tuning in developing accurate stock forecasting models.

Nevertheless, the findings should be interpreted within the scope of the evaluated datasets and experimental settings. The study was conducted using three representative stocks and historical OHLCV market data. Therefore, further validation using larger and more diverse datasets is required to assess the generalizability of the proposed framework across different market environments. Future research may investigate the scalability of the proposed architecture using additional financial assets, alternative market conditions, and broader validation strategies to further examine its forecasting effectiveness.

6. Declarations

6.1. Author Contributions

Conceptualization: I.P., S.S., D.I. and I.A.; Methodology: S.S.; Software: I.P.; Validation: S.S., D.I., and I.A.; Formal Analysis: S.S., D.I., and I.A.; Investigation: I.P.; Resources: S.S.; Data Curation: S.S.; Writing Original Draft Preparation: I.P., S.S., and D.I.; Writing Review and Editing: S.S., I.P., and D.I.; Visualization: I.P.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

Telkom University all provided funding for this work and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," *Elsevier Ltd.* vol. 2022, no. 7, pp. 1-41, doi: 10.1016/j.eswa.2022.116659.
- [2] A. W. Li and G. S. Bastos, "Stock Market Forecasting Using Deep Learning and Technical Analysis : A Systematic Review," *IEEE Access*, vol. 8, no. 10, pp. 1-11, 2020, doi: 10.1109/ACCESS.2020.3030226.
- [3] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Engineering Applications of Artificial Intelligence Ensemble deep learning : A review," *Eng. Appl. Artif. Intell.*, vol. 115, no. 3, pp. 1-18, p. 105151, 2022, doi: 10.1016/j.engappai.2022.105151.
- [4] E. M. Ferrouhi and I. Bouabdallaoui, "A comparative study of ensemble learning algorithms for high-frequency trading," *Sci. African*, vol. 24, no. 2, pp. 1-16, 2024, doi: 10.1016/j.sciaf.2024.e02161.
- [5] J. Zhang and X. Chen, "A two-stage model for stock price prediction based on variational mode decomposition and ensemble machine learning method," *Soft Comput.*, vol. 28, no. 3, pp. 2385–2408, 2024, doi: 10.1007/s00500-023-08441-0.

-
- [6] Y. Lin, S. Liu, H. Yang, and H. Wu, "Stock Trend Prediction Using Candlestick Charting and Ensemble Machine Learning Techniques with a Novelty Feature Engineering Scheme," *IEEE Access*, vol. 9, no. 7, pp. 101433–101446, 2021, doi: 10.1109/ACCESS.2021.3096825.
 - [7] L. Yang and A. Shami, "Neurocomputing On hyperparameter optimization of machine learning algorithms : Theory and practice," *Neurocomputing*, vol. 415, no. 7, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.
 - [8] V. Gkonis and I. Tsakalos, "A hybrid optimized deep learning model via the Golden Jackal Optimizer for accurate stock price forecasting," *Expert Syst. Appl.*, vol. 278, no. 2, pp. 1-17, p. 127287, 2025, doi: 10.1016/j.eswa.2025.127287.
 - [9] J. Timko, R. El Shawi, and S. Tomasiello, "Optimizing stock price forecasting : a hybrid approach using fuzziness and automated machine learning," *Expert Syst. Appl.*, vol. 295, no. 6, pp. 1-12, 2025, p. 128844, 2026, doi: 10.1016/j.eswa.2025.128844.
 - [10] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; A Comparative Analysis," *IEEE Access*, vol. 8, pp. 150199–150212, 2020, doi: 10.1109/ACCESS.2020.3015966.
 - [11] N. Alamsyah, Saparudin, and A. P. Kurniati, "Event Detection Optimization Through Stacking Ensemble and BERT Fine-tuning for Dynamic Pricing of Airline Tickets," *IEEE Access*, vol. 12, no. 9, pp. 145254–145269, 2024, doi: 10.1109/ACCESS.2024.3466270.
 - [12] Y. Li and Y. Pan, "A novel ensemble deep learning model for stock prediction based on stock prices and news," *Int. J. Data Sci. Anal.*, vol. 13, no. 2, pp. 139–149, 2022, doi: 10.1007/s41060-021-00279-9.
 - [13] N. G. Ramadhan, W. Maharani, and A. A. Gozali, "Enhancing SMOTE Using Euclidean Weighting for Imbalanced Classification Dataset," *J. Appl. Data Sci.*, vol. 6, no. 3, pp. 2207–2220, 2025.
 - [14] B. Gülmez, "Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm," *Expert Syst. Appl.*, vol. 227, no. 1, pp. 1-16, 120346, 2023, doi: 10.1016/j.eswa.2023.120346.
 - [15] L. Chen, Z. Qiao, M. Wang, C. Wang, R. Du, and H. E. Stanley, "Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?," *IEEE Access*, vol. 6, no. 5, pp. 48625–48633, 2018, doi: 10.1109/ACCESS.2018.2859809.
 - [16] H. Cao, T. Lin, Y. Li, and H. Zhang, "Stock Price Pattern Prediction Based on Complex Network and Machine Learning," *Complexity*, vol. 2019, no. 5, pp. 1-12, doi: 10.1155/2019/4132485.
 - [17] M. Sharma and H. S. Shekhawat, "Portfolio optimization and return prediction by integrating modified deep belief network and recurrent neural network," *Knowledge-Based Syst.*, vol. 250, no. 8, pp. 1-19, 2022, doi: 10.1016/j.knosys.2022.109024.
 - [18] W. Khan, M. A. Ghazanfar, M. A. Azam, A. Karami, K. H. Alyoubi, and A. S. Alfakeeh, "Stock market prediction using machine learning classifiers and social media, news," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 7, pp. 3433–3456, 2022, doi: 10.1007/s12652-020-01839-w.
 - [19] B. Y. Dwiandiyanta, R. Hartanto, and R. Ferdiana, "Optimization of Stock Predictions on Indonesia Stock Exchange: A New Hybrid Deep Learning Method," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 1, pp. 19370–19379, 2025, doi: 10.48084/etasr.9363.
 - [20] S. Samal and R. Dash, "Developing a novel stock index trend predictor model by integrating multiple criteria decision-making with an optimized online sequential extreme learning machine," *Granul. Comput.*, vol. 8, no. 3, pp. 411–440, May 2023, doi: 10.1007/s41066-022-00338-x.
 - [21] A. Gul, "A Novel Hybrid Ensemble Framework for Stock Price Prediction: Combining Bagging, Boosting, Dagging, and Stacking," no. 0123456789, Springer US, vol. 2025, no. 4, pp 1-31, doi: 10.1007/s10614-025-10979-0.

- [22] S. Wang, "A Stock Price Prediction Method Based on BiLSTM and Improved Transformer," *IEEE Access*, vol. 11, no. 5, pp. 104211–104223, 2023, doi: 10.1109/ACCESS.2023.3296308.
- [23] Z. Ye, Y. Wu, H. Chen, Y. Pan, and Q. Jiang, "A Stacking Ensemble Deep Learning Model for Bitcoin Price Prediction Using Twitter Comments on Bitcoin," *Mathematics*, vol. 10, no. 8, 2022, doi: 10.3390/math10081307.
- [24] R. Singhal and R. Kashef, "A Weighted Stacking Ensemble Model With Sampling for Fake Reviews Detection," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 2, pp. 2578–2594, 2024, doi: 10.1109/TCSS.2023.3268548.
- [25] A. Sulaiman, T. B. Kurniawan, D. A. Dewi, and M. Alqudah, "Utilizing Sentiment Analysis for Reflect and Improve Education in Indonesia," *J. Appl. Data Sci.*, vol. 6, no. 1, pp. 189–200, 2025.
- [26] B. H. Hayadi, S. Sofiana, and D. Setiyadi, "Unsupervised Learning Methods for Topic Extraction and Modeling in Large-scale Text Corpora using LSA and LDA," *J. Appl. Data Sci.*, vol. 4, no. 3, pp. 103–118, 2023.