Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 10 No. 3 (2026) 636 - 647 e-ISSN: 2580-0760

Stock Price Prediction Using SVR: A Feature Engineering and Hyperparameter Tuning Approach

Alfian Ramadhan[1], Yoga Pristyanto[2], Anggit Dwi Hartanto[3], Donni Prabowo[4]
[1,2,3,4]Department of Information System, Faculty of Computer Science, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia

alfianramadhan43@students.amikom.ac.id

Abstract

Stock price prediction in Indonesia's volatile mining sector poses significant forecasting challenges driven by commodity price dynamics and structural market shifts. This study proposes a systematic prediction framework for PT Indo Tambangraya Megah Tbk (ITMG.JK) integrating technical and market-derived non-technical feature engineering, LightGBM-based feature selection, multilevel TimeSeriesSplit cross-validation, and hyperparameter optimization. Support Vector Regression (SVR) is benchmarked against LightGBM, XGBoost, and Random Forest under 5-fold, 10-fold, and 15-fold schemes. SVR achieves the best performance at 10-fold, with RMSE of 0.0121, MAE of 0.0090, MAPE of 1.1457%, and R^2 of 0.9249. Generalization experiments across four additional stocks in banking, automotive, and mining sectors confirm SVR's robustness, maintaining R^2 above 0.89 and MAPE below 2.65% in all cases while tree-based models produce negative R^2 on certain datasets. Statistical validation via Wilcoxon signed-rank test ($p < 0.05$) and Cohen's d ($|d| > 0.8$) confirms the significance of SVR's advantage. These findings indicate that SVR consistently outperforms the evaluated models under the proposed experimental framework.

Keywords: cross validation; mining sector; stock price prediction; support vector regression

How to Cite: A. Ramadhan, Y. Pristyanto, A. D. Hartanto, and D. Prabowo, "Stock Price Prediction Using SVR: A Feature Engineering and Hyperparameter Tuning Approach", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 10, no. 3, pp. 636 - 647, Jun. 2026.
Permalink/DOI: https://doi.org/10.29207/resti.v10i3.7249

Received: October 7, 2025
Accepted: May 23, 2026
Available Online: June 14, 2026

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

The mining sector is one of the most important investment sectors in Indonesia because fluctuations in commodity prices have a direct impact on the stock performance of mining companies listed on the Indonesian Stock Exchange [1]. Due to its strong sensitivity to global commodity movements, accurate stock price prediction is essential for supporting investment decisions. High investor interest in this sector also reflects confidence in its profit potential, particularly amid the dynamic nature of global commodity markets [2].

At the same time, global trends toward renewable and sustainable energy sources have begun to influence the long-term outlook of the mining industry [3]. This transition increases the importance of predicting mining stock movements to anticipate market changes and support strategic investment planning. In response to these challenges, previous studies have applied tree-based machine learning algorithms such as Random Forest, LightGBM, and XGBoost because of their ability to handle volatile data, capture complex patterns, and model interactions among variables effectively [4-8].

As the demand for accurate stock price prediction grows, various machine-learning approaches have been combined with parameter optimization techniques to enhance model performance. Study [9] proposed a hybrid BA-SVR (Bat Algorithm Support Vector Regression) model that optimized the core parameters of SVR using the bat algorithm. The results demonstrated that BA-SVR outperformed standard SVR models using polynomial and sigmoid kernels, particularly in short and medium term predictions. In the context of other algorithms, study [6] applied a boosting-based approach to predict AAPL stock prices. In that study, the LightGBM algorithm achieved the best performance, with an average RMSE of 5.12%, while MAE averaged 4.1%. However, different findings were presented in study [4], where an optimized Random Forest algorithm using the D-RF-RS approach outperformed LightGBM. The accuracy of

Random Forest increased by 0.17 points and exceeded LightGBM by 0.18 points in stock price trend direction classification. These findings were further supported by study [10], in which the Random Forest algorithm outperformed the logistic regression model. This indicates that boosting models such as LightGBM are not inherently superior and can be outperformed by other models when appropriate parameter optimization and structural configurations are applied.

Although ensemble models such as Random Forest and LightGBM have frequently demonstrated high performance in various studies, SVR remains a viable alternative, particularly in specific scenarios. For instance, some studies [11], [12] showed that SVR had lower prediction accuracy than LSTM and Random Forest. However, contrasting results were reported in studies [13], where SVR outperformed LSTM, LightGBM, and XGBoost when applied to normalized and stationary datasets. Additional supporting evidence is provided by studies conducted on different stock markets. Supporting evidence from Indonesian and Southeast Asian markets confirms SVR's viability, with reported accuracies consistently exceeding 92% when appropriate preprocessing and tuning are applied [14-16]. By incorporating several technical indicators as input features, these studies collectively reinforce the strong potential of SVR algorithms in effectively modeling stock price dynamics. Such discrepancies observed across different studies are likely attributable to variations in the features used, as well as the absence or presence of comprehensive preprocessing techniques, both of which are crucial for optimizing SVR performance. These observations suggest that the effectiveness of SVR is highly contextual, depending significantly on data quality, preprocessing methods, and parameter tuning strategies. As explained in [17], SVR is known for its strength in handling medium datasets and linear trends, and it performs optimally when the data have been normalized [18].

However, research on stock price prediction in Indonesia's mining sector remains methodologically limited. Previous work applying SVR to PT Indo Tambangraya Megah Tbk (ITMG) [19] used only closing prices as input features and relied on a single train-test split without cross-validation, feature engineering, feature importance analysis, or temporal validation. As a result, the stability and generalization capability of the model were not thoroughly evaluated. Meanwhile, other studies [20] employed traditional time-series approaches such as ARIMA, whose assumptions of linearity and stationarity limit their ability to model the nonlinear behavior commonly found in mining stock movements. This study aims to address existing methodological limitations by developing a stock price prediction framework for ITMG.JK through the integration of technical indicator–based feature engineering and LightGBM feature importance analysis. The research compares the performance of SVR with LightGBM, XGBoost, and Random Forest using the same experimental settings under 5-fold, 10-fold, and 15-fold TimeSeriesSplit validation schemes. In addition, model stability is evaluated using learning curves and residual error distribution analysis, while the Wilcoxon signed-rank test is applied to confirm the statistical superiority of SVR against the strongest competing model. The results are expected to demonstrate that SVR, supported by a structured preprocessing, validation, and optimization process, can provide stable and reliable stock price predictions in volatile mining sector markets.

2. Methods

The research in this study was carried out through several stages, as illustrated in Figure 1.

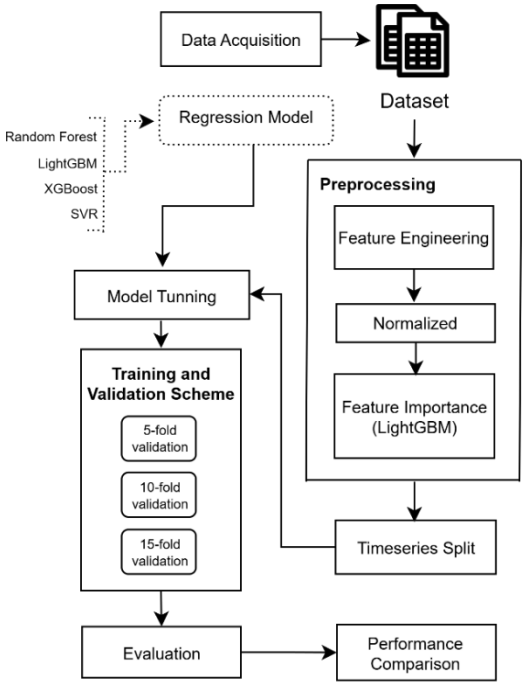


Figure 1. Research Stages

The objective is to develop and evaluate a robust SVR-based stock price prediction framework through systematic feature engineering, feature importance analysis, time-series validation, and comparative model assessment.

2.1 Data Acquisition

The data in this study were collected from Yahoo Finance, which provides comprehensive and near real-time stock market information. The main object of analysis was PT Indo Tambangraya Megah Tbk, while additional datasets were included to examine the generalizability of the proposed framework across different sectors. These consisted of PT Bank Central Asia Tbk, PT Bank Rakyat Indonesia Tbk, PT Astra International Tbk, and PT Bukit Asam Tbk. All stock data covered the same observation period, from May 16, 2013, to May 16, 2025, and were processed using a consistent preprocessing pipeline to maintain experimental uniformity.

Each dataset contained standard financial variables, including open, high, low, close, and trading volume. Data quality inspection showed no missing values, indicating that the datasets were ready for further preprocessing and analysis.

2.2 Feature Engineering

Feature engineering in this study was conducted comprehensively by integrating two categories of features, namely technical features and non-technical features, both of which have been widely utilized and validated in previous stock price prediction studies. Technical features were designed to capture trends, momentum, volatility, and short-term price movements through various technical indicators, while non-technical features were incorporated to represent additional market-related information beyond conventional price indicators. The detailed construction and rationale of these two feature groups are presented in this section, respectively. By combining both technical and non-technical information, the proposed framework aims to provide a richer representation of stock market behavior and improve predictive performance.

Technical features are constructed from six indicator families: (1) Moving Averages (MA-5, MA-20, MA-50, MA-95, MA-155, MA-230) and Exponential Moving Averages (EMA-12, EMA-26, EMA-50), from which MACD and its signal line are derived, (2) Average True Range (ATR) for volatility measurement, (3) Relative Strength Index (RSI) for momentum, (4) Rolling Standard Deviation (10 and 20 days) and Bollinger Bands for risk quantification, (5) Stochastic Oscillator (%K and %D); and (6) lagged closing prices at odd intervals from lag-1 to lag-29 to capture temporal dependencies. All formulas follow standard definitions (Equations 1–5). The full feature list is summarized in Table 2.

$$MA_t = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i} \quad (1)$$

This indicator is useful for determining realistic stop loss placements and position sizing strategies [21]. The formula for ATR is presented in Equation 2.

$$ATR_t = \frac{1}{n}\sum_{i=0}^{n-1} TR_{t-i} \quad (2)$$

The third indicator is the Relative Strength Index (RSI), which is used to identify overbought or oversold conditions, detect divergences (discrepancies between price movement and RSI indicating potential reversals), and confirm trend strength [22]. The formula for RSI is presented in Equation 3.

$$RSI = 100 - \left(\frac{100}{1+\frac{EMA_n(Gain)}{EMA_n(Loss)}}\right) \quad (3)$$

The fourth indicator is the Exponential Moving Average (EMA), which functions similarly to the Simple Moving Average (SMA) in identifying trends and support/resistance levels, but with greater responsiveness to recent price changes [23]. The formula for EMA is presented in Equation 4, where EMA_tdenotes the Exponential Moving Average at time t, $P_t$represents the closing price at time t, $\alpha$is the smoothing factor ($0 < \alpha < 1$), and EMA_{t-1}denotes the EMA value at the previous time step.

$$EMA_t = \alpha \cdot P_t + (1-\alpha) \cdot EMA_{t-1} \quad (4)$$

The next indicator is the Rolling Standard Deviation, which measures the volatility or risk of an asset [24]. The formula for the Rolling Standard Deviation is presented in Equation 5, where $RollingSTD_t$denotes the rolling standard deviation at time t, P_{t-i}represents the closing price at time $t-i$, $n$is the rolling window size, and $\bar{P}$denotes the mean closing price over the same rolling window.

$$RollingSTD_t = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}\left(P_{t-i} - \overline{P}\right)^2} \quad (5)$$

Finally, the closing price lag feature (close lag) was employed as one of the crucial supporting features to enhance model performance. A total of 30 lag features were created. This feature is designed to capture the closing price values from previous periods, which are essential for modeling temporal dependencies and recognizing historical price movement patterns that strongly predict future market conditions.

In addition to technical indicators, this study incorporates market-derived non-technical features to broaden the representational capacity of the input space. These features are motivated by evidence that stock price dynamics are influenced not only by historical price behavior, but also by recurring structural patterns in investor activity, temporal regularities in market returns, and the prevailing directional context of the market [25], [26]. Three groups of non-technical features are incorporated to enrich the representation of stock price dynamics.

First, calendar-based features capture temporal trading patterns, including day-of-week, month, quarter, and end-of-period indicators, motivated by well-documented calendar anomalies in financial markets [27]. Second, volume anomaly features quantify deviations in trading activity relative to a 20-day rolling baseline using z-score, volume ratio, and spike indicators, reflecting information-driven market behavior [26]. Third, price regime features encode market trends through binary indicators based on moving average positions, trend direction, and crossover events, which have been shown to enhance predictive performance compared to raw indicator values [28].

2.3 Feature Importance

In this study, feature selection was performed using feature importance scores obtained from the LightGBM algorithm. These scores reflect each feature's contribution to the tree-splitting process and its impact on improving model performance [29]. The split frequency based approach was chosen in this study due to its simplicity and effectiveness in ranking features for

predictive tasks [29], [30]. The feature selection was performed using the split count method, the default method in LightGBM. A threshold score greater than 30 was applied to select the most relevant features. The matrix illustrating the split count concept is shown in Equation 6.

$$Importance(f_i) = \sum_{t=1}^{T} 1[f_i \text{ used in split at node } t] \quad (6)$$

2.4 Regression Model

This study employed four machine learning algorithms: Support Vector Regression (SVR), LightGBM, XGBoost, and Random Forest. These methods were selected for their ability to model nonlinear relationships, handle high-dimensional data, and maintain robust predictive performance. SVR is specifically designed for regression tasks and seeks to achieve accurate predictions while controlling model complexity, making it effective for stock price forecasting [18], [30]. Equation 7 represents the SVR formulation.

$$min \frac{1}{2}|w|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \quad (7)$$

Random Forest is an ensemble learning algorithm that constructs multiple decision trees and aggregates their outputs to enhance accuracy and reduce overfitting [31]. The corresponding equation is presented as Equation 8.

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \quad (8)$$

LightGBM is a gradient boosting algorithm developed for high efficiency and scalability [32]. During the decision tree construction process, LightGBM utilizes the entropy function to determine the optimal split at each node. Entropy measures the level of uncertainty or impurity in the dataset. The entropy formula for discrete variables is presented in Equation 9.

$$Entropy(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \quad (9)$$

XGBoost is an ensemble algorithm derived from Gradient Boosting that builds multiple decision trees and combines their predictions to enhance accuracy and reduce overfitting [33]. It optimizes an objective function that includes a loss term and a regularization term, making it robust and efficient [34]. The objective function is presented in Equation 10.

$$\mathcal{L}(t) = \sum_{i=1}^{n} l\left(y_i, \widehat{y_i^{(t)}}\right) + \sum_{k=1}^{t} \Omega(f_k) \quad (10)$$

2.5 Time Series Split

In the next stage, the dataset was partitioned into training/validation and testing sets, with the last 10% of observations reserved for testing and the remaining 90% used for training and validation. This sequential division preserves temporal order and prevents data leakage [35]. The study then employed the TimeSeriesSplit method for cross-validation, which is specifically designed for time-series data because it maintains temporal dependencies without random shuffling [36]. In each fold, the training data gradually expand from the earliest observations, while the validation set consists of the subsequent time block, ensuring that future information is never used during model training.

2.6 Hyperparameter Tunning

The hyperparameter tuning process in this study was designed based on commonly used configurations in previous machine learning studies for stock price prediction [4], [13], [18], [37], [38]. The parameter search ranges for SVR, LightGBM, XGBoost, and Random Forest were selected to ensure effective optimization and a fair comparison among algorithms. To preserve the temporal structure of financial time-series data, the GridSearchCV procedure incorporated a time-series-aware cross-validation scheme using predefined n_splits values. This approach allows optimal hyperparameter selection while maintaining temporal dependencies within the dataset [18]. Table 1 summarizes the parameter ranges explored for each algorithm during the GridSearchCV process.

Table 1. Hyperparameters for Machine Learning Algorithms

Algorithm	Param
SVR	Kernel : 'linear', ‘rbf’, ‘poly’
	C : 0.1, 1, 10, 100
	Epsilon : 0.01, 0.1, 0.5, 1
	Gamma : 'scale', ‘auto’
Random Forest	Max_depth : none, 10, 20
	Max_features : ‘sqrt’, 'log2'
	Min_samples_leaf: 1, 2, 4
	Min_samples_split : 2, 5, 10
	n_estimator : 50, 100, 200
LightGBM	Learning_rate: 0.01, 0.05, 0.1
	Max_depth : 5, 7, 10, 12
	Min_data_in_leaf: 10, 20, 30
	Num_leaves : 20, 30, 50, 70
XGBoost	colsample_bytree: 0.8, 1.0
	gamma: 0, 0.1, 0.2
	learning_rate: 0.01, 0.1, 0.3
	max_depth: 3, 5, 7
	n_estimators: 50, 100, 200
	subsample: 0.8, 1.0

2.7 Evaluation Model Performance

The final stage involved evaluating the predictive models using four performance metrics: RMSE, MAE, MAPE, and R^2. These metrics were selected to provide a comprehensive and interpretable assessment of prediction accuracy and model effectiveness [11], [39], [40].

Root Mean Squared Error (RMSE) measures the average magnitude of prediction errors by calculating the square root of the mean squared differences between predicted ($\hat{y}_i$) and actual values (y_i), as shown in Equation 11. Due to the squaring process, RMSE places greater emphasis on larger prediction errors [39].

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (11)$$

where n is the number of observations, $y_i$is the actual value, and $\hat{y}_i$ is the predicted value. The Mean Absolute

Error, shown in Equation 12, computes the average of the absolute differences between predicted and actual values:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (12)$$

The Coefficient of Determination (R^2) quantifies how well the model explains the variance in the actual values relative to a naive baseline model that always predicts the mean $\bar{y}$, where $\bar{y}$ denotes the mean of the actual values. as shown in Equation 13. Higher R^2 values indicate better explanatory power and model fit:

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

Mean Absolute Percentage Error (MAPE) is a scale-independent evaluation metric that quantifies the average magnitude of prediction error as a percentage of the actual values. The formula is defined as Equation 14.

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \times 100 \quad (14)$$

where $A_t$denotes the actual value at time t, $F_t$represents the forecasted value at time t, and $n$is the total number of observations. This property is particularly advantageous in financial forecasting contexts, where stock price magnitudes may differ substantially across instruments and time periods [40].

3. Results and Discussions

3.1 Results

The primary dataset consists of 2,723 daily trading records for ITMG.JK. Additional datasets for BBRI, BBCA, PTBA, and ASII were included, covering the period from May 2013 to May 2025, as illustrated in line chart ITMG in Figure 2.



Figure 2. Graph of Closing Stock Price 2013 - 2025

The initial stage of the data preprocessing process began by checking for the presence of null values and potential outliers in dataset.

The technical and non-technical indicators used as input variables are summarized in Tables 2 and 3, respectively. Table 2 outlines the feature engineering process used in this study. The feature set includes trend indicators (MA, EMA, MACD, and signal line), volatility indicators (Rolling Standard Deviation, ATR, and Bollinger Bands), momentum indicators (RSI and Stochastic Oscillator), as well as lagged prices and price-difference features. Before feature selection using LightGBM, all variables were normalized to a [0,1] range through Min-Max Scaling.

Table 2. Technical Indicators.

Technical Indicators	Full name of index
MA	Moving Average
RSI	Relative Strength Index
ATR	Average True Range
EMA	Exponential Moving Average
Rolling STD	Rolling Standard Deviation
MACD	Moving Average Convergence Divergence
HL Range	High-Low Range
Close-Open	Closing minus Opening Price
BB	Bollinger Band
Stoch	Stochastic
Close Lag	Lagged Closing Price

Table 3. Non-Technical Indicators.

Category	Feature	Description
Calendar	day_of_week	Trading day of the week (0=Monday, 4=Friday)
Calendar	month	Month of the year (1–12)
Calendar	quarter	Calendar quarter (1–4)
Calendar	is_month_end	Binary: 1 if last trading day of the month
Calendar	is_quarter_end	Binary: 1 if last trading day of the quarter
Volume Anomaly	volume_zscore	Z-score of volume relative to 20-day rolling mean
Volume Anomaly	volume_ratio	Ratio of daily volume to 20-day rolling mean
Volume Anomaly	high_vol_flag	Binary: 1 if volume z-score exceeds 2.0
Price Regime	above_MA50	Binary: 1 if close price is above 50-day MA
Price Regime	above_MA200	Binary: 1 if close price is above 230-day MA
Price Regime	trend_strength	Normalized distance of close from 50-day MA
Price Regime	golden_cross	Binary: 1 on day MA50 crosses above MA230
Price Regime	death_cross	Binary: 1 on day MA50 crosses below MA230

The next stage in the data preprocessing pipeline was feature selection. This study, established a significance threshold of 30, meaning only features with an importance score above this threshold were considered for model training. The results of the feature selection process are presented in detail in Table 4.

Table 4 presents the feature importance scores obtained from the LightGBM algorithm for all features generated during the feature engineering process. Out of the total 60 features, 38 were selected as important based on a predefined importance threshold (score > 30), while the remaining features were excluded due to their relatively low contribution. The closing price (Close) achieved the highest importance score (287), followed by Rolling Standard Deviation (rolling_std_10), RSI, and ATR. Other notable features include lag-based variables (e.g., Close_lag_21, Close_lag_27), momentum indicators (e.g., Stoch_K), and price-based transformations such as Close_Open and HL_range. These results indicate that features capturing volatility (rolling_std_10, ATR), momentum (RSI, Stoch_K), and historical price dependencies (lag features) play a dominant role in the

prediction process. The computed importance scores were subsequently used to select the most relevant feature subset for model training.

Table 4. Importance Score.

Technical Indicators	Score
Close	287
Rolling_std_10	217
RSI	175
ATR	172
Vol_price_corr	120
Vol_lag_1	114
Price_accel	112
Close_Open	104
MA_230	102
Close_lag_21	101
Stoch_K	100
Gap_lag_1	98
Low	92
vol_lag_5	91
MA_95	87
Close_lag_27	83
Close_lag_9	82
ATR	81
Close_lag_29	79
High	77
Close_lag_15	75
open_close_gap	74
Close_lag_11	74
MA_155	72
Close_lag_19	70
Volume_zscore	69
HL_range	68
Close_lag_17	67
Close_lag_3	64
Close_lag_13	61
Close_lag_23	60
Trend_strength	54
MA_5	50
BB_width	45
Close_lag_7	44
MA_50	41
Open	40
Month	31

Table 5 presents the data split used for the training and testing processes.

Table 5. Data Distribution Training and Testing.

Percentage	Total Data
90%	2451
10%	272

Table 5 presents the train-validation and test split, maintaining a 90:10 ratio to preserve temporal integrity [37]. Furthermore, during the training and validation phase, the 90% portion was split using time-based cross-validation with an expanding window approach, specifically the TimeSeriesSplit method. This scheme applied several configurations of fold numbers (*n_splits*), which 5, 10, and 15 to examine the model's stability and generalization capability.

Table 6 presents the best parameter results obtained from the Grid Search Cross-Validation (GSCV) stage.

Table 6 presents the optimal hyperparameter configurations obtained through GridSearchCV for each algorithm. These parameter settings were then used consistently in all subsequent cross-validation experiments. Using the optimized configurations, each model was evaluated on the test set under three different TimeSeriesSplit schemes, as summarized in Tables 7–9. Table 7 specifically reports the evaluation results for the 5-fold cross-validation configuration.

Table 6. Hyperparameters for Machine Learning Algorithms

Algorithm	Param
SVR	Kernel = linear, C = 10, epsilon = 0.01, gamma = scale
Random Forest (RF)	max_depth: 10, max_features: log2, min_samples_leaf: 1, min_samples_split: 10, n_estimators: 100
LightGBM	learning_rate: 0.1, max_depth: 10, min_data_in_leaf: 10, num_leaves: 30
XGBoost	colsample_bytree: 0.8, gamma: 0, learning_rate: 0.1, max_depth: 3, n_estimators: 50, subsample: 0.8

Table 7. Model Evaluation using K-fold 5.

Model	RMSE	MAE	MAPE	R^2
SVR	0.0122	0.0090	1.1492%	0.9247
LightGBM	0.0163	0.0127	1.6062%	0.8649
XGBoost	0.0166	0.0127	1.6155%	0.8600
RF	0.0150	0.0114	1.4385%	0.8858

The evaluation results of the four algorithms under the 5-fold cross-validation scheme showed that SVR achieved the best performance with an RMSE of 0.0122, MAE of 0.0090, and an R^2 score of 0.9247. LightGBM followed with an RMSE of 0.0163 and an R^2 of 0.8649, while Random Forest obtained an RMSE of 0.0150, R^2 of 0.8858, and MAPE 1.4385%. Meanwhile, XGBoost demonstrated the lowest performance, with an RMSE of 0.0166, MAE of 0.0127, MAPE 1.6155% and an R^2 score of 0.8600.

Table 8 presents the model evaluation results under the 10-fold cross-validation scheme. SVR again recorded the best performance, with an RMSE of 0.0121, MAE of 0.0090, lowest MAPE 1.1457%, and an R^2 score of 0.9249. Following SVR, XGBoost achieved an RMSE of 0.0138, R^2 of 0.9032, and second lowest MAPE after SVR 1.3575%, while RF recorded an RMSE of 0.0150 and an R^2 of 0.8847. LightGBM showed the lowest performance, with an RMSE 0.0165 and an R^2 score of 0.8626.

Table 8. Model Evaluation using K-fold 10.

Model	RMSE	MAE	MAPE	R^2
SVR	0.0121	0.0090	1.1457%	0.9249
LightGBM	0.0165	0.0128	1.6200%	0.8626
XGBoost	0.0138	0.0107	1.3575%	0.9032
RF	0.0150	0.0113	1.4402%	0.8847

Table 9 presents the evaluation results under the 15-fold cross-validation scheme. SVR demonstrated consistent performance, with an RMSE of 0.0133, MAE of 0.0092, MAPE 1.1357% and an R^2 score of 0.9233. XGBoost and Random Forest showed competitive results, with R^2 values of 0.8929 and 0.8880, respectively. LGBM achieving an R^2 score of 0.8781.

Figure 3 compares the actual stock prices and the predicted prices by the SVR model using the 10-fold.

Table 9. Model Evaluation using K-fold 15.

Model	RMSE	MAE	MAPE	R^2
SVR	0.0133	0.0092	1.1357%	0.9233
LightGBM	0.0155	0.0118	1.4954%	0.8781
XGBoost	0.0145	0.0111	1.4102%	0.8929
RF	0.0148	0.0113	1.4259%	0.8880

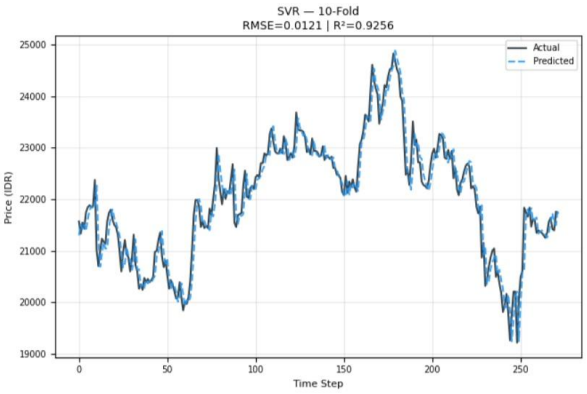


Figure 3. Comparison of Actual and Predicted Closing Prices

This visualization illustrated the closeness between the predicted results and the actual data, highlighting the model's ability to accurately and consistently capture the patterns of stock price movements.

Figures 4 and 5 present scatter plots of the residual distributions for the SVR and XGBoost models under the 10-fold cross-validation scheme, which were used to observe the differences in prediction error patterns between the algorithms.

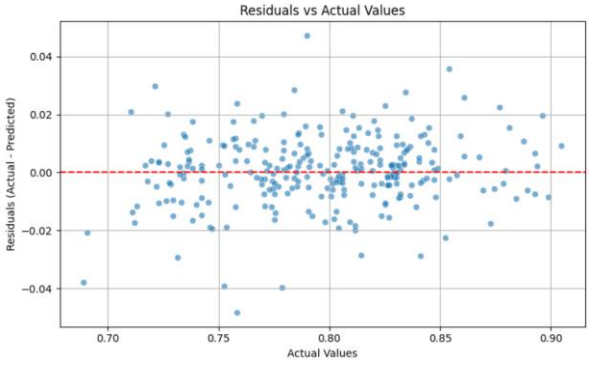


Figure 4. Scatter Plot Residual vs. Actual SVR.

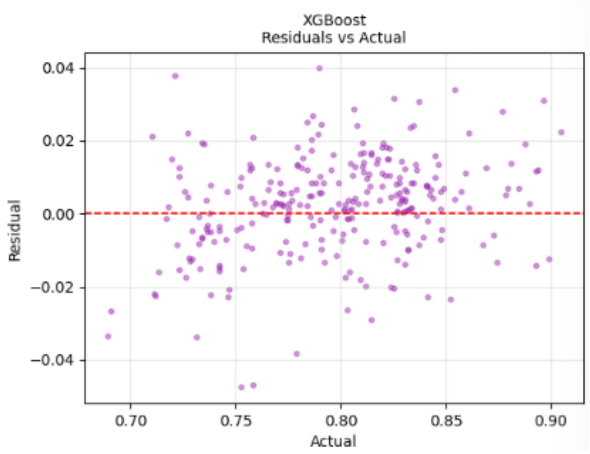


Figure 5. Scatter Plot Residual vs. Actual XGboost.

The plot showed that the residuals (the differences between actual and predicted values) were distributed relatively symmetrically and randomly around the zero line (red dashed line). No specific linear or curved patterns were observed. Most points clustered near the zero residual area, indicating that the SVR model's prediction errors were controlled and stable across the range of actual values.

The residual plot for the XGBoost model exhibited a broader and less uniform spread than SVR. Some residual areas appeared asymmetrically distributed, with a biased concentration in a certain range. Figures 6 and 7 present histograms of the residual distributions for the SVR and XGBoost models.

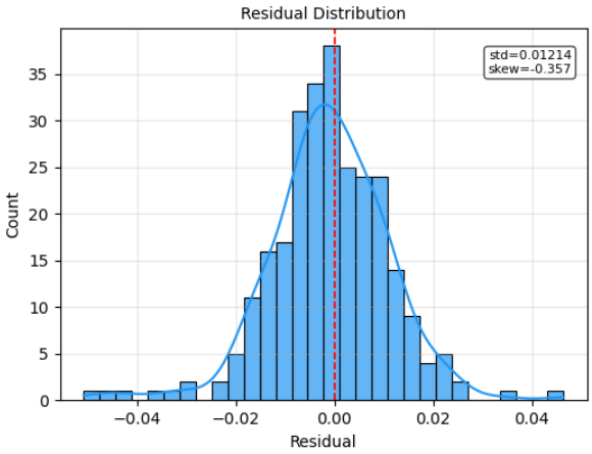


Figure 6. Histogram of SVR Residual Distribution

The residual distribution histogram of the SVR model in Figure 6 shows an approximately normal pattern with most prediction errors concentrated near zero, indicating that the majority of errors are relatively small. This pattern is reinforced by the kernel density curve, which forms a smooth and nearly symmetric bell-shaped distribution. Although the residuals exhibit a slight negative skewness (−0.357), suggesting occasional overestimation of actual prices, the deviation remains limited. In addition, the residual standard deviation of 0.01214 indicates a narrow error spread, reflecting the strong stability and generalization capability of the SVR model.

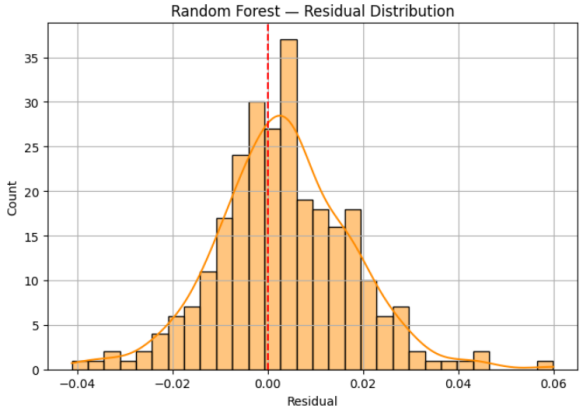


Figure 7. Histogram of XGBoost Residual Distribution

Conversely, the distribution peak, while remaining close to zero, is slightly shifted to the right of center, suggesting a systematic tendency toward underprediction. A more pronounced negative skewness of −0.469 indicates a heavier left tail, implying that XGBoost produces larger overprediction errors with greater frequency than SVR. This is further corroborated by a higher standard deviation of 0.01362 approximately 12.2% larger than that of SVR reflecting greater variability in prediction error and reduced consistency in capturing the nonlinear dynamics of the stock price series. Figures 8 and 9 present the per fold RMSE distribution and fold-by-fold error trajectory derived from the 10-fold TimeSeriesSplit cross-validation for the SVR and XGBoost models, respectively. These visualizations serve a dual purpose: to evaluate the stability and consistency of each model's predictive performance across varying temporal training windows, and to assess the degree to which each model is susceptible to overfitting as the validation fold progresses forward in time.

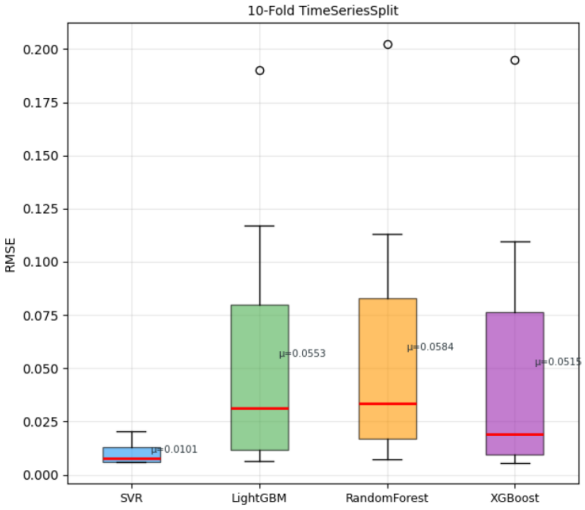


Figure 8. RMSE Distribution — SVR vs XGBoost (10-Fold TimeSeriesSplit Boxplot)

Figure 8 presents the distribution of per fold RMSE, SVR exhibits a remarkably compact and stable distribution, with a mean RMSE of 0.01012 and a narrow interquartile range, reflecting high consistency in predictive accuracy across all temporal validation folds. In contrast, XGBoost yields a substantially wider distribution with a mean RMSE of 0.05150, an expansive interquartile range, and a notable outlier approaching ± 0.190 indicating considerable instability in generalization performance as the training window shifts forward in time.

Figure 9 illustrates the RMSE performance of SVR and XGBoost across all 10 temporal folds, with dashed lines representing the average RMSE of each model. SVR demonstrates a consistently low and stable error pattern throughout the validation process, remaining close to its mean RMSE of 0.0101 with only minor deviations. This stability indicates strong generalization capability under time-series-aware validation and reflects the Structural Risk Minimization principle of SVR, which controls model complexity and supports consistent performance across changing data distributions [18]. In contrast, XGBoost shows substantial RMSE fluctuations across several folds, with errors increasing sharply in folds 4, 8, and 9. Such instability suggests higher sensitivity to temporal distribution shifts, a limitation commonly observed in ensemble-based models applied to non-stationary financial time-series data [4].

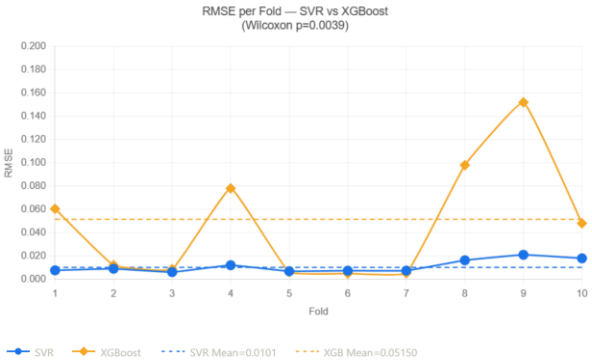


Figure 9. RMSE per Fold SVR vs XGBoost (Wilcoxon p = 0.0039)

These findings are further supported by the learning curve analysis, where SVR exhibits a very small generalization gap (~0.00048), while XGBoost shows a much larger gap (~0.07157), indicating a stronger tendency toward overfitting. Figures 10 and 11 present the comparative learning curve analysis of both models.

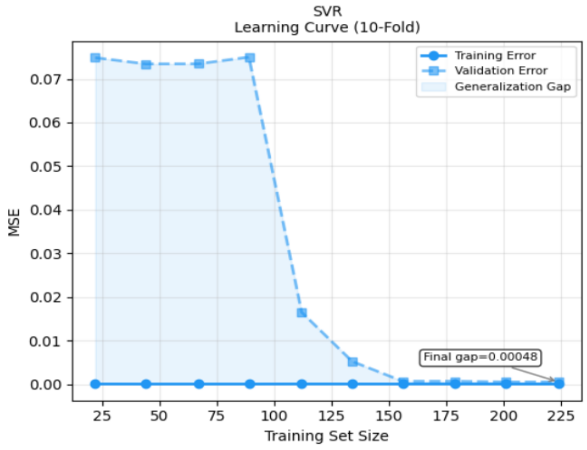


Figure 10. Learning Curve SVR

The learning curve of SVR shows a strong convergence pattern that reflects good model generalization. At smaller training sizes, the validation error starts at around 0.080 MSE and gradually decreases as more training data are introduced, while the training error remains consistently close to zero. As the dataset approaches its full training size, the training and validation errors converge to a low and stable value, resulting in a very small generalization gap of 0.00048. This indicates that SVR is able to balance bias and variance effectively without showing signs of overfitting or underfitting, while its predictive performance continues to improve as additional training data become available.

In contrast, the learning curve of XGBoost reveals a persistent and severe generalization gap across all training sizes. The training error remains effectively zero regardless of data volume, while the validation error stabilizes at approximately 0.070 MSE from the outset and shows no meaningful reduction as training size increases resulting in a final gap of 0.07024. This pattern is a textbook indication of overfitting: the model memorizes training data with near-perfect accuracy but fails to generalize to unseen temporal segments.

To assess whether the performance advantages observed on ITMG.JK reflect genuine model robustness rather than dataset-specific behavior, the proposed framework was applied to four additional stocks spanning two distinct market sectors: the banking sector, represented by BBCA.JK and BBRI.JK, and the automotive and consumer goods sector, represented by ASII.JK, alongside PTBA.JK as a second mining sector reference. All stocks were evaluated under identical preprocessing, feature engineering, and validation conditions as the primary experiment, using the 10-fold TimeSeriesSplit scheme. The comparative results are presented in Table 10.

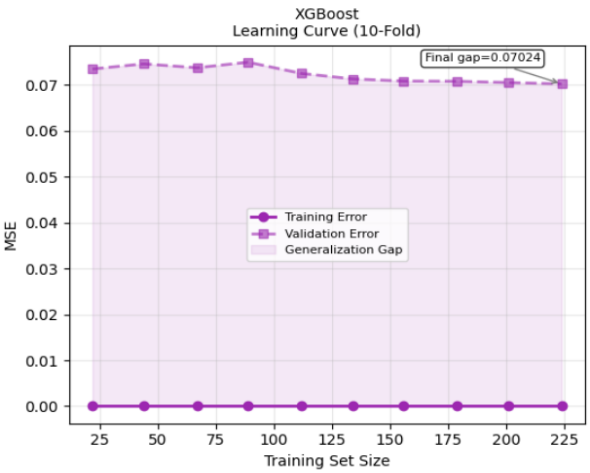


Figure 11. Learning Curve XGBoost

Table 10. Generalization Performance of Models Across Different Stocks (10-Fold).

Stocks	Model	RMSE	MAE	MAPE	R^2
ITMG	SVR	0.0133	0.0092	1.1357%	0.9277
	LightGBM	0.0167	0.0133	1.6802%	0.8571
	XGBoost	0.0143	0.0110	1.4010%	0.8959
	RF	0.0142	0.0106	1.3417%	0.8973
BBCA	SVR	0.0174	0.0134	1.5768%	0.9313
	LightGBM	0.0459	0.0363	4.0671%	0.5225
	XGBoost	0.0477	0.0393	4.4517%	0.4844
	RF	0.0485	0.0388	4.3669%	0.4694
BBRI	SVR	0.0200	0.0152	2.3379%	0.9609
	LightGBM	0.0355	0.0284	4.3394%	0.8770
	XGBoost	0.0306	0.0254	3.9027%	0.9084
	RF	0.0556	0.0491	7.6587%	0.6987
ASII	SVR	0.0212	0.0158	2.6416%	0.8904
	LightGBM	0.0240	0.0183	3.0587%	0.8596
	XGBoost	0.0228	0.0174	2.8955%	0.8731
	RF	0.0220	0.0165	2.7561%	0.8823
PTBA	SVR	0.0166	0.0117	1.3963%	0.9207
	LightGBM	0.0684	0.0505	5.7294%	-0.3410
	XGBoost	0.0660	0.0488	5.5407%	-0.2481
	RF	0.0858	0.0691	7.9153%	-1.1104

Across all evaluated stocks and sectors, SVR consistently achieved the best predictive performance, maintaining R^2 values above 0.89 and MAPE values below 2.65%. The performance gap between SVR and the benchmark models was especially evident in BBCA.JK and PTBA.JK. For BBCA.JK, the tree-based models produced substantially lower R^2 values, with Random Forest reaching only 0.469, while SVR maintained an R^2 of 0.931. An even larger difference appeared in PTBA.JK, where LightGBM, XGBoost, and Random Forest generated negative R^2 values, indicating performance worse than a simple mean-based prediction and suggesting severe overfitting to training data patterns. In contrast, SVR still achieved an R^2 of 0.921 with a MAPE of 1.396% on the same dataset. These findings demonstrate the stronger robustness and generalization capability of SVR when facing varying data distributions. This stability is closely related to the Structural Risk Minimization (SRM) principle underlying SVR, which balances model complexity and prediction accuracy rather than focusing solely on minimizing training error [18].

3.2 Discussion

The superior performance of SVR across all evaluation metrics and validation schemes can be explained by its Structural Risk Minimization (SRM) principle [19], which controls model complexity through margin maximization instead of solely minimizing training error. This characteristic is particularly beneficial for financial time-series data, where data distributions often change over time. In contrast, tree-based ensemble methods rely on split-based learning structures that are more sensitive to training distributions, resulting in the RMSE instability observed across multiple validation folds. The combination of SRM-based learning in SVR and consistent Min-Max normalization throughout the

framework also contributes to its strong cross-sector generalization capability.

Among the benchmark models, XGBoost showed the closest performance to SVR on the test datasets. However, the Wilcoxon signed-rank test on the RMSE values obtained from the 10-fold TimeSeriesSplit confirmed that the advantage of SVR was statistically significant (W = 1.000, p = 0.0039). SVR consistently outperformed XGBoost in all validation folds without exception. As illustrated in Figure 9, XGBoost also demonstrated larger RMSE fluctuations across temporal folds, especially in later validation periods, indicating greater sensitivity to distribution shifts in financial time-series data. Table 11 summarizes the per-fold RMSE comparison and Wilcoxon signed-rank test results between SVR and XGBoost under the 10-fold TimeSeriesSplit configuration.

Table 11. ITMG Wilcoxon Signed-Rank Test SVR vs. XGBoost

Fold	SVR RMSE	XGBoost RMSE	Superior
1	0.0076	0.0150	SVR
2	0.0089	0.0104	SVR
3	0.0060	0.0068	SVR
4	0.0119	0.0220	SVR
5	0.0068	0.0073	SVR
6	0.0072	0.0070	XGBoost
7	0.0073	0.0070	XGBoost
8	0.0162	0.0350	SVR
9	0.0209	0.0450	SVR
10	0.0179	0.0150	SVR
Mean ± Std	0.01012 ± 0.0049	0.05150 ± 0.0590	-
W Statistic	-	-	1.000
p-value	-	-	0.0039
Significance	-	-	P<0.05

SVR achieves lower RMSE in all 10-folds. Wilcoxon signed-rank test ($\alpha = 0.05$) confirms the performance difference is statistically significant.

To further contextualize these findings and position the proposed model within existing research, a comparative evaluation against prior studies was conducted. The detailed comparison of performance metrics is presented in Table 12 of the study [19], [38].

Table 12 compares the results of this study with previous studies [19] and [38], both of which examined stock price prediction in the Indonesian mining sector using machine learning approaches.

Table 12. Research comparison and validation.

Research	RMSE	MAE	MAPE	R^2
Proposed Method	0.0133	0.0091	1.1192%	0.9277
Previous Research [19]	-	-	7.09%	0.9063
Previous Research [38]	0.0199	0.0133	2.713%	0.99388

This comparison was conducted to evaluate the effectiveness of the proposed framework and position the findings within existing research. Compared to [19], which reported a MAPE of 7.09% and R^2 of 0.9063 for ITMG stock using a single train-test split without feature engineering or cross-validation, the proposed framework achieved a significantly lower MAPE of 1.1192% and a higher R^2 of 0.9277. These improvements indicate that the integration of systematic feature engineering, LightGBM-based feature selection, and TimeSeriesSplit validation substantially enhances prediction performance.

Although [38] reported lower RMSE and MAE values with a higher R^2 of 0.99388, the comparison should be interpreted carefully because the study used a different stock dataset and relied on a single experimental split without temporal cross-validation or feature importance analysis. The very high R^2 reported in [38] may therefore indicate overfitting rather than strong generalization ability. In contrast, the results of this study remained consistent across 5-fold, 10-fold, and 15-fold TimeSeriesSplit evaluations, with statistical significance confirmed through the Wilcoxon signed-rank test (W = 1.000, p = 0.0039). To further validate the framework, cross-sector experiments were conducted by comparing SVR with the best-performing competing models for each stock. As presented in Table 13, SVR consistently achieved higher R^2 values across all evaluated stocks, demonstrating its robustness and adaptability in handling different market characteristics.

Table 13. Best-evaluation for model generalization.

Stock	Sektor	SVR R^2	Best Competitor R^2	Gap
ITMG	Mining	0.9249	0.9032 (XGB)	+0.022
BBCA	Banking	0.9313	0.5226 (LGB)	+0.409
BBRI	Banking	0.9610	0.9084 (XGB)	+0.053
ASII	Consumer /Auto	0.8904	0.8823 (RF)	+0.008
PTBA	Mining	0.9207	−0.248 (XGB)	+1.169

The superiority of SVR was further confirmed through statistical analysis. Across all five stocks, the Wilcoxon signed-rank test consistently produced p-values below 0.05 for every comparison between SVR and the benchmark models, indicating that the lower RMSE achieved by SVR was statistically significant rather than caused by random variation. This result was strengthened by Cohen's d effect sizes, which ranged from 0.940 to 2.216, exceeding the threshold for a large effect in all cases. In contrast, comparisons among the tree-based models, particularly between LightGBM and XGBoost, mostly produced non-significant results with very small effect sizes below 0.1, suggesting similar generalization capabilities among them. These findings indicate that SVR consistently outperformed the competing models across different market conditions.

Beyond its methodological contribution, the proposed framework also offers practical value for stock price forecasting in the Indonesian capital market. The ability of SVR to maintain MAPE below 2.65% across multiple sectors provides reliable next-day price estimates that can support trading decisions, particularly in volatile sectors such as mining and banking. In addition, the use of TimeSeriesSplit validation ensures that the reported performance reflects genuine out-of-sample prediction capability,

increasing its suitability for real-world implementation. The modular design of the framework, which integrates feature engineering, LightGBM-based feature selection, and time-series validation, can also be adapted to other financial assets while maintaining interpretability through feature importance analysis.

4. Conclusions

This study proposes a stock price prediction framework for PT Indo Tambangraya Megah Tbk using 2,723 daily observations from May 2013 to May 2025. The framework combines technical and non-technical feature engineering, LightGBM-based feature selection, multilevel TimeSeriesSplit cross-validation, and hyperparameter optimization to address methodological limitations found in previous studies.

The performance comparison among SVR, LightGBM, XGBoost, and Random Forest shows that SVR consistently delivers the best results across all validation schemes, achieving an RMSE of 0.0121, MAPE of 1.1457%, and R^2 of 0.9249 under the 10-fold configuration. To evaluate model robustness and generalizability, the framework was also tested on four additional stocks from the banking, automotive, and mining sectors, namely PT Bank Central Asia Tbk, PT Bank Rakyat Indonesia Tbk, PT Astra International Tbk, and PT Bukit Asam Tbk. Across all datasets, SVR maintained R^2 values above 0.89 and MAPE below 2.65%, while several tree-based models produced negative R^2 values, indicating overfitting tendencies. Statistical validation using the Wilcoxon signed-rank test and Cohen's d further confirmed the superiority of SVR, with significant differences ($p < 0.05$) and large effect sizes across all comparisons.

The main contribution of this study lies in providing a reproducible and statistically validated evaluation framework that demonstrates the robustness and generalizability of SVR under varying market conditions. By emphasizing methodological rigor and stability-oriented validation, the study strengthens empirical evidence regarding model reliability for financial time-series forecasting. Nevertheless, several limitations remain. The framework does not incorporate external macroeconomic variables such as commodity prices or interest rates, focuses only on single-step forecasting, and relies solely on end-of-day data. Future studies may improve the framework by integrating macroeconomic and sentiment-related features, extending forecasting horizons, and exploring hybrid architectures such as SVR-LSTM or SVR-Transformer.

Acknowledgment

Thanks to the Department of Research and Community Service, Universitas Amikom Yogyakarta for supporting this research.

References

[1] A. R. Putra and R. Robiyanto, "The effect of commodity price changes and USD/IDR exchange rate on Indonesian mining companies' stock return," *Jurnal Keuangan dan Perbankan*, vol. 23, no. 1, Jan. 2019, doi: 10.26905/jkdp.v23i1.2084.

[2] G. Putra, S. Mulyantini, and D. Arieftiara, "Business diversification of coal mining companies as a strategy facing coal price volatility: The effect on company performance and share prices," *International Journal of Business Ecosystem & Strategy (2687-2293)*, vol. 3, pp. 38–50, Feb. 2021, doi: 10.36096/ijbes.v3i4.299.

[3] M. K. G. Deshmukh, M. Sameeroddin, D. Abdul, and M. Abdul Sattar, "Renewable energy in the 21st century: A review," *Mater. Today Proc.*, vol. 80, pp. 1756–1759, Jan. 2023, doi: 10.1016/j.matpr.2021.05.501.

[4] L. Yin, B. Li, P. Li, and R. Zhang, "Research on stock trend prediction method based on optimized random forest," *CAAI Trans. Intell. Technol.*, vol. 8, no. 1, pp. 274–284, Mar. 2023, doi: 10.1049/cit2.12067.

[5] A. U. Haq, A. Zeb, Z. Lei, and D. Zhang, "Forecasting daily stock trend using multi-filter feature selection and deep learning," *Expert Syst. Appl.*, vol. 168, Apr. 2021, doi: 10.1016/j.eswa.2020.114444.

[6] A. D. Hartanto, Y. Nur Kholik, and Y. Pristyanto, "Stock Price Time Series Data Forecasting Using the Light Gradient Boosting Machine (LightGBM) Model," *JOIV : International Journal on Informatics Visualization*, vol. 7, no. 4, p. 2270, Dec. 2023, doi: 10.62527/joiv.7.4.1740.

[7] E. H. A. Rady, H. Fawzy, and A. M. A. Fattah, "Time series forecasting using tree based methods," *J. Stat. Appl. Probab.*, vol. 10, no. 1, pp. 229–244, 2021, doi: 10.18576/JSAP/100121.

[8] X. Zhao and Q. Zhao, "Stock Prediction Using Optimized LightGBM Based on Cost Awareness," in *2021 5th IEEE International Conference on Cybernetics (CYBCONF)*, IEEE, Jun. 2021, pp. 107–113. doi: 10.1109/CYBCONF51991.2021.9464148.

[9] J. Zheng, Y. Wang, S. Li, and H. Chen, "The stock index prediction based on svr model with bat optimization algorithm," *Algorithms*, vol. 14, no. 10, Oct. 2021, doi: 10.3390/a14100299.

[10] P. Sadorsky, "A Random Forests Approach to Predicting Clean Energy Stock Prices," *Journal of Risk and Financial Management*, vol. 14, no. 2, Feb. 2021, doi: 10.3390/jrfm14020048.

[11] M. Y. Urrochman, H. Asy'ari, and F. A. Hizham, "Performance Comparison Of Random Forest Regression, Svr Models In Stock Price Prediction," *Jurnal Pilar Nusa Mandiri*, vol. 21, no. 1, pp. 16–23, Mar. 2025, doi: 10.33480/pilar.v21i1.6072.

[12] A. Arfan and L. ETP, "Perbandingan Algoritma Long Short-Term Memory dengan SVR Pada Prediksi Harga Saham di Indonesia," *PETIR*, vol. 13, no. 1, pp. 33–43, Mar. 2020, doi: 10.33322/petir.v13i1.858.

[13] M. Chlebus, M. Dyczko, and M. Woźniak, "Nvidia's Stock Returns Prediction Using Machine Learning Techniques for Time Series Forecasting Problem," *Central European Economic Journal*, vol. 8, no. 55, pp. 44–62, Jan. 2021, doi: 10.2478/ceej-2021-0004.

[14] B. T. Khoa and T. T. Huynh, "Forecasting stock price movement direction by machine learning algorithm," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 6625–6634, Dec. 2022, doi: 10.11591/ijece.v12i6.pp6625-6634.

[15] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "Efficient Stock-Market Prediction Using Ensemble Support Vector Machine," *Open Computer Science*, vol. 10, no. 1, pp. 153–163, Jan. 2020, doi: 10.1515/comp-2020-0199.

[16] D. A. Puspitasari and Z. Rustam, "Application of SVM-KNN using SVR as feature selection on stock analysis for Indonesia stock exchange," 2018, p. 020207. doi: 10.1063/1.5064204.

[17] Y. Zhang, Y. Song, and G. Wei, "A feature-enhanced long short-term memory network combined with residual-driven ν support vector regression for financial market prediction," *Eng. Appl. Artif. Intell.*, vol. 118, p. 105663, 2023, doi: 10.1016/j.engappai.2022.105663.

[18] R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, "Fine-tuned support vector regression model for stock predictions," *Neural Comput. Appl.*, vol. 35, no. 32, pp. 23295–23309, Nov. 2023, doi: 10.1007/s00521-021-05842-w.

[19] D. H. Yudhawan and T. Purwaningsih, "Developing support vector regression model to forcast stock prices of mining companies in Indonesia," *Jurnal Informatika*, vol. 14, no. 2, p. 43, May 2020, doi: 10.26555/jifo.v14i2.a17283.

[20] D. Nuryadin, I. B. P. C. P. Sarayuda, and D. Q. Nada, "Forecasting the Stock Price of Coal and Coal Commodity Companies using the ARIMA and ARCH/GARCH Models for 2011-2022," *Jurnal Samudra Ekonomi dan Bisnis*, vol. 16, no. 1, pp. 29–45, Jan. 2025, doi: 10.33059/jseb.v16i1.10797.

[21] A. Ntakaris, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Mid-price prediction based on machine learning methods with technical and quantitative indicators," *PLoS One*, vol. 15, no. 6, p. e0234107, Jun. 2020, doi: 10.1371/journal.pone.0234107.

[22] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, "Predicting stock market index using LSTM," *Machine Learning with Applications*, vol. 9, p. 100320, Sep. 2022, doi: 10.1016/j.mlwa.2022.100320.

[23] S. Y. Kuo and Y. H. Chou, "Building Intelligent Moving Average-Based Stock Trading System Using Metaheuristic Algorithms," *IEEE Access*, vol. 9, pp. 140383–140396, 2021, doi: 10.1109/access.2021.3119041.

[24] E. S. Gunnarsson, H. R. Isern, A. Kaloudis, M. Risstad, B. Vigdel, and S. Westgaard, "Prediction of realized volatility and implied volatility indices using AI and machine learning: A review," *International Review of Financial Analysis*, vol. 93, p. 103221, May 2024, doi: 10.1016/j.irfa.2024.103221.

[25] D. Stosic, D. Stosic, I. Vodenska, H. E. Stanley, and T. Stosic, "A New Look at Calendar Anomalies: Multifractality and Day-of-the-Week Effect," *Entropy*, vol. 24, no. 4, p. 562, Apr. 2022, doi: 10.3390/e24040562.

[26] V. Azevedo and C. Hoegner, "Enhancing stock market anomalies with machine learning," *Review of Quantitative Finance and Accounting*, vol. 60, no. 1, pp. 195–230, Jan. 2023, doi: 10.1007/s11156-022-01099-z.

[27] K. Aggarwal and M. K. Jha, "Stock returns seasonality in emerging asian markets," *Asia-Pacific Financial Markets*, vol. 30, no. 1, pp. 109–130, Mar. 2023, doi: 10.1007/s10690-022-09370-y.

[28] A. Y. Wiiava, C. Fatichah, and A. Saikhu, "Stock Price Prediction with Golden Cross and Death Cross on Technical Analysis Indicators Using Long Short Term Memory," in *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, IEEE, Aug. 2022, pp. 278–283. doi: 10.1109/icoiact55506.2022.9971844.

[29] M. Wu, Q. Wan, and Y. Wang, "Academic performance evaluation based on knowledge sharing theory: feature importance analysis of dimensions in LightGBM model," in *Proceedings of 2025 6th International Conference on Education, Knowledge and Information Management, ICEKIM 2025*, Association for Computing Machinery, Inc, Sep. 2025, pp. 302–308. doi: 10.1145/3756580.3756629.

[30] D. Wang, Y. Huang, and C. Ren, "Prediction and contribution analysis of CSI 300 Stock Index Futures based on KPCA-LightGBM," *BCP Business & Management*, vol. 28, pp. 1–11, Oct. 2022, doi: 10.54691/bcpbm.v28i.2135.

[31] P. K. Illa, B. Parvathala, and A. K. Sharma, "Stock price prediction methodology using random forest algorithm and support vector machine," *Mater. Today Proc.*, vol. 56, pp. 1776–1782, 2022, doi: 10.1016/j.matpr.2021.10.460.

[32] G. Ke *et al.*, "LightGBM: a highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, in NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 3149–3157. doi: 10.5555/3294996.3295074

[33] A. Performa *et al.*, "Performance Analysis of XGBoost Algorithm to Determine the Most Optimal Parameters and Features in Predicting Stock Price Movement," *Jurnal Informatika dan Teknologi Informasi*, vol. 20, no. 1, pp. 91–102, 2023, doi: 10.31515/telematika.v20i1.9329.

[34] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[35] V. Cerqueira, L. Torgo, and I. Mozetic, "Evaluating time series forecasting models: An empirical study on performance estimation methods," May 2019, doi: 10.1007/s10994-020-05910-7.

[36] I. R. Parray, S. S. Khurana, M. Kumar, and A. A. Altalbe, "Time series data analysis of stock price movement using machine learning techniques," *Soft comput.*, vol. 24, no. 21, pp. 16509–16517, Nov. 2020, doi: 10.1007/s00500-020-04957-x.

[37] T. N. Tran, B. M. Lam, A. T. Nguyen, and Q. B. Le, "Load forecasting with support vector regression: influence of data normalization on grid search algorithm," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 4, pp. 3410–3420, Aug. 2022, doi: 10.11591/ijece.v12i4.pp3410-3420.

[38] N. P. S. Yuli Artini, I. W. Sumarjaya, and D. P. E. Nilakusmawati, "Penerapan Metode Support Vector Regression (Svr) Dengan Algoritma Grid Search Dalam Peramalan Harga Saham," *E-Jurnal Matematika*, vol. 13, no. 2, p. 94, May 2024, doi: 10.24843/mtk.2024.v13.i02.p447.

[39] H. Yao, "Research on Stock Price Prediction Based on Machine Learning Techniques," in *Proceedings of the 2nd International Conference on Data Science and Engineering*, SCITEPRESS - Science and Technology Publications, 2025, pp. 654–659. doi: 10.5220/0013703600004670.

[40] H. Lee, J. H. Kim, and H. S. Jung, "Deep-learning-based stock market prediction incorporating ESG sentiment and technical indicators," *Sci. Rep.*, vol. 14, no. 1, p. 10262, May 2024, doi: 10.1038/s41598-024-61106-2.