Infortech, Vol. 8, No. 1, June 2026, E-ISSN: 2715-8160, Page. 109-118
Published: LPPM Universitas Bina Sarana Informatika

JURNAL INFORTECH

website: http://jurnal.bsi.ac.id/index.php/infortech/

Comparative Analysis of Statistical Machine Learning and Deep Learning for Daily CSPI Forecasting

Haerul [1], Yudi Ramdhani[2], Novia Andini [3]

[1,2,3]Information Systems Program, Satu University, Bandung, Indonesia

ARTICLE INFORMATION

Artikel History:
Received: 20-05-2026
Revised: 03-06-2026
Accepted: 17-06-2026
Available Online: 22-06-2026

Keyword:

Bayesian Optimization
Composite Stock Price Index
Deep Learning
Machine Learning
Stock Market Forecasting
Time Series Analysis

ABSTRACT

Forecasting stock market indices is challenging due to the complex, nonlinear, and dynamic behavior of financial markets. Although statistical, machine learning, and deep learning methods have been widely applied, their comparative performance for daily Composite Stock Price Index (CSPI) forecasting remains insufficiently explored. This study systematically compares nine forecasting models representing statistical (ARIMA, SARIMA, ETS), machine learning (Random Forest, Support Vector Regression, XGBoost), and deep learning (LSTM, BiLSTM, GRU) approaches using the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework. Historical daily CSPI closing prices from 1995 to 2019 were preprocessed through missing-value handling, feature engineering, normalization, and chronological data partitioning. Statistical models were trained on the original price scale, whereas machine learning and deep learning models used normalized data. Hyperparameters were optimized using Bayesian Optimization with the Optuna framework. Because the models were evaluated on different scales, cross-paradigm comparisons primarily relied on the coefficient of determination (R^2) and Directional Accuracy (DA). The results show that machine learning models generally outperformed statistical and deep learning approaches on the testing dataset. Random Forest achieved the highest predictive performance with an R^2 of 0.902, while Support Vector Regression produced the lowest MAE and MAPE among normalized models. Although LSTM achieved the lowest validation error, its performance did not generalize consistently to unseen data. Directional Accuracy remained relatively low (31–41%), indicating that predicting market direction is more difficult than forecasting price magnitude. Future studies should incorporate macroeconomic indicators and market sentiment to improve forecasting performance.

Corresponding Author:

Haerul,
Information System,
Satu University,
Bandung, Indonesia
Email: haerul@univ.satu.ac.id

INTRODUCTION

The Composite Stock Price Index (CSPI), known as *Indeks Harga Saham Gabungan (IHSG)*, is the main stock market index of the Indonesia Stock Exchange (IDX) that reflects the overall performance of listed companies. As a key indicator of market conditions and investor confidence, the CSPI plays an important role in assessing the health of Indonesia's economy. Its movements are influenced by various macroeconomic factors, such as exchange rates, inflation, interest rates, and global commodity prices. Accurate forecasting of the CSPI is essential for investors, analysts, and policymakers. Reliable predictions can assist investors in making informed investment decisions while supporting regulators in formulating policies to maintain market stability.

DOI: https://doi.org/10.31294/infortech.v7i2.

Therefore, developing effective forecasting models for the CSPI remains an important research topic in financial time series analysis (Suprapti & Hafizh, 2022).

Despite its importance, forecasting stock market movements remains challenging due to the high volatility and complex dynamics of the CSPI. Stock prices are influenced by various factors, including market demand and supply, socio-political conditions, and global sentiment. Traditional statistical models, such as ARIMA, have demonstrated strong performance in short-term forecasting of stationary time-series data (Ray et al., 2023). The effectiveness of conventional statistical methods is often constrained by their reliance on linear relationships, which may not adequately represent the nonlinear dynamics of financial markets. Consequently, machine learning approaches, including SVM, RF, and KNN, have emerged as popular alternatives for modeling complex patterns. Furthermore, deep learning architectures such as LSTM have demonstrated strong forecasting capabilities by learning temporal dependencies from sequential data. Previous studies in Indonesia have also reported the effectiveness of LSTM in predicting stock prices, achieving low forecasting errors across various market indices and individual stocks (Airlangga, 2024; Ramdhani et al., 2025).

Numerous methodologies have been utilized to predict movements in the stock market. Conventional statistical tools like ARIMA perform well in identifying linear time-based patterns, yet tend to fall short when dealing with the nonlinear dynamics inherent in financial data. Machine learning alternatives such as SVR and Random Forest emerged as viable solutions, offering greater capacity to handle intricate data relationships, while deep learning architectures particularly LSTM networks have shown considerable promise owing to their strength in recognizing long-range temporal dependencies. Despite this progress, comparative studies that evaluate statistical, machine learning, and deep learning models for daily CSPI forecasting under a single, consistent experimental scheme remain scarce, particularly in the Indonesian stock market context, where most existing work examines these paradigms in isolation rather than under comparable preprocessing, data-splitting, and evaluation protocols (Osman & Otaibi, 2025).

This gap is compounded by the fact that the influence of training data size, hyperparameter configuration, and evaluation scale on forecasting accuracy has not been comprehensively examined within the Indonesian stock market context (Jin, 2024).

To address this gap, this research offers a systematic comparative analysis of nine forecasting models spanning the statistical (ARIMA, SARIMA, ETS), machine learning (Random Forest, SVR, XGBoost), and deep learning (LSTM, BiLSTM, GRU) paradigms for forecasting daily CSPI values. The contribution of this study lies in evaluating all nine models under a single hyperparameter optimization procedure and a consistent, fair evaluation protocol that combines numerical accuracy metrics with Directional Accuracy, rather than relying on scale-dependent error metrics alone. The specific objective of this research is to determine which of the three modeling paradigms provides the most reliable balance between forecasting accuracy and directional prediction capability for daily CSPI movements, with deep learning approaches, LSTM in particular, expected to be competitive given their capacity to model intricate nonlinear dynamics and extended temporal patterns characteristic of financial time-series data (Putra et al., 2025).

RESEARCH METHOD

To achieve the research objectives, a structured experimental framework was developed to compare statistical, machine learning, and deep learning approaches for forecasting daily movements of the CSPI. The overall workflow adopts the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework (Wirth & Hipp, 2000), which organizes a data-driven study into business understanding, data understanding, data preparation, modeling, evaluation, and deployment phases. These generic phases were operationalized into the eight sequential stages followed in this study: data collection and preparation (business and data understanding), preprocessing and feature engineering (data preparation), dataset splitting (data preparation), forecasting model development (modeling), hyperparameter optimization (modeling), model training and testing (modeling and evaluation), and performance evaluation and comparison (evaluation). The overall research workflow is illustrated in Figure 1.

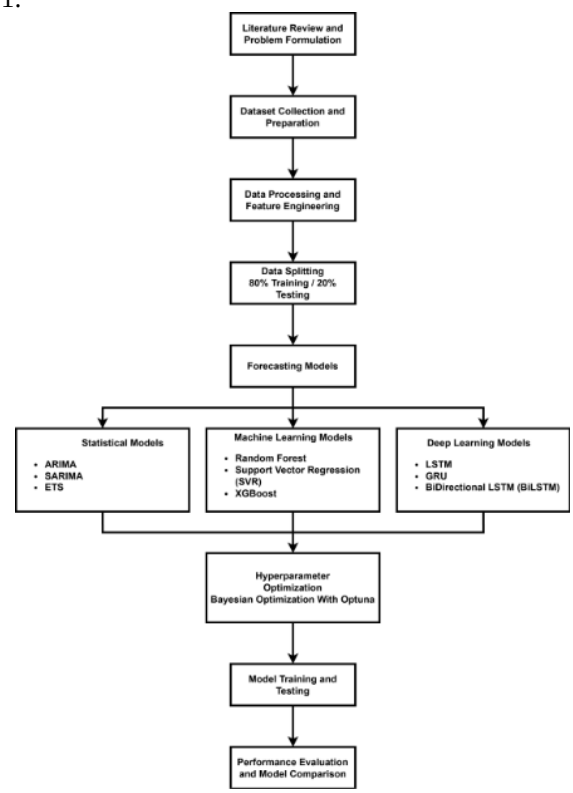

Figure 1. Research Stages

1. Research Design

This study adopts a quantitative experimental research design with a comparative approach to evaluate the forecasting performance of different modeling paradigms on financial time-series data. The primary objective is to compare statistical, machine learning, and deep learning models in predicting daily movements of the CSPI. To ensure a fair comparison, all models are developed and evaluated under a consistent experimental framework using the same dataset, preprocessing procedures, training–testing strategy, and evaluation metrics.

The comparative analysis focuses on three key aspects: forecasting accuracy, directional prediction capability, and model performance stability. Forecasting accuracy is assessed using standard error metrics, while directional prediction capability evaluates each model's ability to correctly identify upward and downward market movements. Performance stability is examined by analyzing the consistency of forecasting results across different experimental settings. Through this approach, the study aims to identify the most effective forecasting method for daily CSPI prediction and provide empirical evidence regarding the relative strengths and limitations of statistical, machine learning, and deep learning techniques in the Indonesian stock market context.

2. Data Description

This study utilizes historical daily data of the CSPI obtained from the publicly available *Daily IHSG* dataset on Kaggle (Harrison, n.d.), accessed on November 22, 2025. The raw dataset comprises 7,486 daily observations covering the period from 1995 to 2025 and contains standard stock market attributes, including opening price, highest price, lowest price, and closing price. To maintain a consistent historical market regime and avoid distortion from the abnormal, sharp decline in the CSPI triggered by the COVID-19 pandemic in 2020 and the atypical market conditions that followed, the data used for model development and evaluation in this study is restricted to the period from 1995 to 2019. In this study, the closing price serves as the target variable since it captures the market's final trading value for a given day and is widely recognized as a key indicator in stock market prediction studies; all nine models are therefore trained to forecast the next-day CSPI closing price as a regression task. Directional Accuracy, reported alongside the error-based metrics in Section 3.7, is computed post hoc from the sign of the change between the predicted and actual closing price and is not modeled as a separate classification target. Prior to model development, the dataset is examined for completeness and consistency to ensure data quality. Any required preprocessing procedures, such as handling missing values and data normalization, are performed before the modeling stage

To preserve the temporal structure of the time-series data and prevent data leakage, the dataset is divided chronologically using a time-based split strategy. In this study, 80% of the observations (1995-2018) are allocated to the training set, while the remaining 20% (2019) are reserved for testing and are never touched during model fitting or hyperparameter search. For hyperparameter optimization, the training set is further partitioned chronologically, with the most recent portion of the training period held out as a validation subset used to score candidate hyperparameter configurations; the test set remains completely isolated throughout this process. The training subset is used to fit each candidate configuration, the validation subset is used to select the best-performing configuration, and the held-out testing set is used exclusively for the final, one-time performance evaluation reported in Section 4. This approach ensures that all forecasting models are assessed using unseen future data, thereby providing a realistic evaluation of predictive performance under actual forecasting conditions.

3. Data Preprocessing and Feature Engineering

Data preprocessing was performed to ensure data consistency and compatibility with the requirements of different forecasting approaches. Missing values were handled using the forward-fill method to preserve temporal continuity, while data normalization was applied when required by the forecasting models. To ensure the integrity of the evaluation process and minimize the risk of data leakage, normalization was performed based solely on training set parameters, which were subsequently extended to transform the testing data in a consistent manner.

For machine learning and deep learning approaches, feature engineering was conducted to capture market characteristics related to trend, momentum, and volatility. The generated features included returns, moving averages, volatility measures, and momentum indicators. Any observations containing missing values resulting from rolling-window calculations were removed to maintain data integrity. For deep learning models, the data were transformed into sequential representations using a sliding-window mechanism, enabling recurrent neural networks to learn temporal dependencies. In contrast, preprocessing for statistical models focused primarily on achieving stationarity through differencing procedures guided by statistical testing.

a. Data Loading and Temporal Alignment

The CSPI dataset was loaded as a daily time-series dataset, with the date column converted into a datetime format and designated as the temporal index. To maintain temporal continuity, the data were aligned to a daily frequency. Missing trading dates were addressed through forward-fill imputation to preserve the continuity of the time series and ensure reliable feature construction and forecasting performance (Han et al., 2026).

b. Missing Value Handling and Normalization

Missing values introduced during temporal alignment were imputed using the forward-fill method. In contrast, observations containing missing values generated by rolling-window calculations during feature engineering were removed. Feature scaling was performed using the StandardScaler method, with scaling parameters estimated exclusively from the training data and subsequently applied to the testing data to prevent information leakage (Wani & Abeer, 2026).

c. Feature Engineering

Feature engineering was applied to the machine learning and deep learning models to capture the dynamic behavior of stock market movements. The extracted features included (Thanh Nhon et al., 2025):

1) Daily return and logarithmic return
2) Simple Moving Average (SMA) with window sizes of 7, 14, 21, and 50 days
3) Exponential Moving Average (EMA) with window sizes of 7, 14, and 21 days
4) Volatility measures calculated as the rolling standard deviation of log returns over 7-day and 21-day periods
5) Relative Strength Index (RSI) with a 14-day period
6) MACD and its corresponding signal line

Observations containing missing values resulting from rolling-window computations were excluded from further analysis to ensure data consistency.

d. Preprocessing for Statistical Models

When working with classical statistical models like ARIMA and SARIMA, the preprocessing stage centered on establishing stationarity within the time series. The ADF test was carried out as a means of determining whether differencing transformations were warranted. The optimal differencing order was identified using the training dataset and incorporated into the model development process to achieve stationarity prior to forecasting (Hossain et al., 2025).

e. Sequence Generation for Deep Learning Models

For deep learning models based on recurrent neural networks, the dataset was transformed into sequential samples using a sliding-window approach. Each input sequence consisted of a predefined number of previous observations (look-back window) used to predict the subsequent value in a one-step-ahead forecasting setting. This approach to data representation allows the model to effectively accommodate both transient variations and long-range temporal dependencies that are characteristic of financial time-series data (Özgür & Orman, 2023).

4. Forecasting Models

a. Statistical Models

1) ARIMA

The ARIMA model is a classical statistical approach that represents linear patterns in time-series data by combining autoregressive, integrated, and moving-average components. The model is particularly effective for stationary time-series data and serves as a benchmark for classical forecasting approaches (Ray et al., 2023).

ARIMA Model Fitting and Forecasting
Input:
Time series y[1..n]
Model orders (p, d, q)
Forecast horizon h

1. Apply differencing d times to obtain a stationary series y'.
2. Estimate the AR(p) and MA(q) parameters using maximum likelihood estimation.
3. Perform residual diagnostic tests (e.g., ACF/PACF analysis and the Ljung–Box test).
4. Generate forecasts for y'[n+1...n+h] using the fitted ARIMA model.
5. Apply inverse differencing to transform the forecasts back to the original scale.

Output:
Forecast values ŷ[n+1...n+h]

2) SARIMA

The SARIMA model extends ARIMA by incorporating seasonal components, enabling the capture of periodic fluctuations that may occur within a time series. This enables the model to capture recurring seasonal patterns that may be present in financial time-series data (Hossain et al., 2025).

SARIMA Model Fitting and Forecasting
Input:
Time series y[1..n]
Non-seasonal orders (p, d, q)
Seasonal orders (P, D, Q)
Seasonal period s
Forecast horizon h

1. Apply non-seasonal differencing d times.
2. Apply seasonal differencing D times with lag s.
3. Estimate the non-seasonal AR(p) and MA(q) parameters together with the seasonal AR(P) and MA(Q) parameters using maximum likelihood estimation.
4. Perform residual diagnostic tests to verify that the residuals resemble white noise.
5. Generate h-step-ahead forecasts using the fitted SARIMA model.
6. Apply inverse seasonal and non-seasonal differencing to restore the forecasts to the original scale.

Output:
Forecast values ŷ[n+1...n+h]

3) Error, Trend, and Seasonality (ETS)

Through exponential smoothing, the ETS model systematically decomposes a given time series into distinct components representing error, trend, and seasonal variation. It is designed to model underlying structural patterns and provide robust forecasts under different trend and seasonality conditions (Elseidi, 2023).

Algorithm ETS Model Fitting and Forecasting
Input:

Time series y[1..n]
ETS component specification (Error, Trend, Seasonal)
Forecast horizon h
1. Initialize the level, trend, and seasonal components (if applicable).
2. Estimate the smoothing parameters (α, β, γ) by maximizing the likelihood.
3. Update the level, trend, and seasonal components recursively using exponential smoothing equations.
4. Generate h-step-ahead forecasts based on the fitted ETS model and the selected component structure.

Output:
Forecast values ŷ[n+1...n+h]

b. Machine Learning Models

1) RF

Random Forest is an ensemble learning algorithm that constructs multiple decision trees using bootstrap sampling and random feature selection. The final prediction is obtained by averaging the outputs of all individual trees, thereby improving predictive accuracy, reducing variance, and mitigating the risk of overfitting (Alamsyah et al., 2023).

Random Forest Regression
Input:
Training data (X, y)
Number of trees T
Maximum tree depth d
Minimum samples per split

1. For each tree t = 1,...,T:
a. Draw a bootstrap sample from the training data.
b. Grow a regression tree using the bootstrap sample.
c. At each split, randomly select a subset of features.
d. Split the node using the best feature and continue until the stopping criteria are met.
2. Store all trained regression trees.
3. For a new input x, obtain predictions from all trees.
4. Compute the final prediction as the average of the individual tree predictions.

Output:
Predicted value ŷ(x)

2) SVR

SVR adapts the core concepts underlying support vector machines for application in regression settings. Through the use of kernel-based mapping techniques, the model becomes capable of effectively identifying and capturing nonlinear dependencies between input variables and the desired output (Montesinos López et al., 2022).

Support Vector Regression (SVR)
Input:
Training data (X, y)
Regularization parameter C
Epsilon (ε)
RBF kernel parameter γ
1. Map the input data into a high-dimensional feature space using the RBF kernel.
2. Train the SVR model by optimizing the regression function while minimizing prediction error within the ε-insensitive loss.
3. Identify the support vectors and estimate the model parameters.
4. For a new input x, compute the predicted value using the trained SVR model.

Output:
Predicted value ŷ(x)

3) Extreme Gradient Boosting (XGBoost)

XGBoost is an ensemble learning algorithm based on gradient boosting that sequentially builds decision trees to minimize prediction errors. By incorporating gradient optimization, regularization techniques, and parallel processing, XGBoost achieves high predictive accuracy while reducing overfitting and improving computational efficiency, making it well suited for large-scale forecasting problems (Syafei & Efrilianda, 2023).

Extreme Gradient Boosting (XGBoost)
Input:
Training data (X, y)
Number of boosting rounds M
Learning rate η
Maximum tree depth
Regularization parameters
1. Initialize the model with the mean value of the target variable.
2. For each boosting iteration (m = 1,...,M):
a. Compute the gradients and Hessians of the loss function.
b. Train a regression tree using the gradient and Hessian information.
c. Apply regularization to control model complexity.
d. Update the ensemble prediction using the learning rate.
3. Repeat until the maximum number of boosting rounds is reached or convergence is achieved.
4. Use the final ensemble model to generate predictions for new data.

Output:
Predicted value ŷ(x)

c. Deep Learning Models

1) Long Short-Term Memory (LSTM)

LSTM is a recurrent neural network architecture specifically designed to capture long-term dependencies in sequential data. Its memory-cell mechanism enables the model to retain relevant information over extended periods, making it suitable for financial time-series forecasting (Han et al., 2026).

Long Short-Term Memory (LSTM)
Input:

Training sequences X
Target values y
Number of epochs
Learning rate
1. Initialize the LSTM network parameters.
2. For each training epoch:
a. Feed the input sequence into the LSTM network.
b. Update the forget, input, and output gates to maintain the cell state.
c. Compute the hidden state and generate the predicted output.
d. Calculate the loss between the predicted and actual values.
e. Update the network parameters using backpropagation through time (BPTT).
3. Repeat until the stopping criterion is satisfied.
4. Use the trained LSTM model to predict future values.
Output:
Forecasted value $\hat{y}$

2) Gated Recurrent Unit (GRU)

GRU is a simplified Recurrent Neural Network (RNN) architecture derived from LSTM that employs update and reset gates to capture temporal dependencies in sequential data. By using fewer gating mechanisms than LSTM, GRU reduces computational complexity while maintaining competitive forecasting performance, making it an efficient alternative for financial time-series prediction (Cahuantzi et al., 2023).

Gated Recurrent Unit (GRU)
Input:
Training sequences X
Target values y
Number of epochs
Learning rate
1. Initialize the GRU network parameters.
2. For each training epoch:
a. Feed the input sequence into the GRU network.
b. Update the reset and update gates to control information flow.
c. Compute the hidden state and generate the predicted output.
d. Calculate the loss between the predicted and actual values.
e. Update the network parameters using backpropagation through time (BPTT).
3. Repeat until the stopping criterion is satisfied.
4. Use the trained GRU model to predict future values.
Output:
Forecasted value $\hat{y}$

3) Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM extends the conventional LSTM architecture by processing sequential data in two directions: forward and backward. This bidirectional mechanism allows the model to capture information from both past and future contexts within a sequence, providing a more comprehensive representation of temporal dependencies for forecasting tasks (Cahuantzi et al., 2023).

Bidirectional Long Short-Term Memory (BiLSTM)
Input:
Training sequences X
Target values y
Number of epochs
Learning rate
1. Initialize the forward and backward LSTM network parameters.
2. For each training epoch:
a. Process the input sequence in the forward direction using LSTM units.
b. Process the reversed input sequence in the backward direction using LSTM units.
c. Combine the forward and backward hidden states.
d. Generate predictions using the combined hidden representation.
e. Compute the loss and update model parameters using backpropagation through time (BPTT).
3. Repeat until the stopping criterion is satisfied.
4. Use the trained BiLSTM model to generate future forecasts.
Output:
Forecasted value $\hat{y}$

5. Hyperparameter Optimization

To ensure a fair and unbiased comparison among forecasting models, hyperparameter optimization was conducted consistently across all model categories. The optimization process employed Bayesian Optimization using the Optuna framework, which iteratively explores the hyperparameter search space by selecting promising configurations based on previous evaluation results. Compared with conventional grid search methods, Bayesian Optimization offers greater efficiency by reducing the number of evaluations required to identify near-optimal parameter combinations.

For each model, candidate hyperparameter configurations were trained using the training dataset and evaluated on the validation set. The hyperparameter configuration yielding the lowest validation error was selected for the final training phase and subsequent evaluation on the testing dataset. To ensure comparable optimization effort across models, the number of optimization trials was fixed at 100 for each forecasting model.

a. Hyperparameter Search Space for Statistical Models

For the ARIMA model, the autoregressive order (p) and moving-average order (q) were optimized within the range of 0 to 5, while the differencing order (d) was determined during the stationarity analysis stage. For the SARIMA model, the non-seasonal

parameters (p and q) were optimized within the range of 0 to 5, whereas the seasonal autoregressive (P) and moving-average (Q) orders were explored within the range of 0 to 2. The seasonal period was fixed at seven days to capture potential weekly patterns in stock market movements (Hossain et al., 2025).

For the ETS model, the optimization process considered different combinations of trend and seasonal components, including additive, multiplicative, and no-component specifications. Seasonal periods ranging from 2 to 12 were also evaluated.

b. Hyperparameter Search Space for Machine Learning Models

For the Random Forest algorithm, hyperparameter optimization focused on the number of decision trees, maximum tree depth, minimum samples required for node splitting, and minimum samples required at leaf nodes. The search space included 50–300 estimators and maximum depths ranging from 5 to 30.

Hyperparameter tuning for SVR focused on optimizing the regularization parameter, epsilon-insensitive loss parameter, and kernel coefficient using Bayesian Optimization. The RBF kernel was selected for all experiments because of its capability to model complex nonlinear relationships. (Montesinos López et al., 2022).

The optimization procedure for XGBoost explored several key hyperparameters, including the number of boosting iterations, maximum tree depth, learning rate, subsampling rate, and column sampling ratio. These parameters were selected because of their significant influence on predictive performance and model generalization (Syafei & Efrilianda, 2023).

c. Hyperparameter Search Space for Deep Learning Models

For the deep learning models, including LSTM, GRU, and Bidirectional LSTM (BiLSTM), the optimization process focused on the number of hidden units, dropout rate, learning rate, and batch size. The number of hidden units was explored within the range of 32 to 256, while dropout values varied between 0.0 and 0.5. Learning rates were optimized on a logarithmic scale between 10^{-5} and 10^{-2}, and batch sizes of 16, 32, and 64 were evaluated.

To maintain consistency across experiments and reduce computational variability, the number of training epochs was fixed at 10 for all deep learning models and was not included as an optimization parameter.

6. Model Training and Testing Procedure

All forecasting models were developed and evaluated within a unified experimental framework to ensure a fair comparison across statistical, machine learning, and deep learning paradigms. Each model was trained using the same training dataset and optimized using the hyperparameter configuration identified during the Bayesian Optimization process. For statistical models, parameter estimation was performed using maximum likelihood estimation techniques. Machine learning models were trained using supervised learning algorithms based on the engineered feature set, whereas deep learning models were trained on sequential input data generated through the sliding-window approach. Throughout the training process, all models were exposed exclusively to information contained within the training dataset.

Following model development, forecasting performance was evaluated on the testing dataset, which remained completely isolated from both the training and hyperparameter optimization stages. This separation ensured that model performance reflected true out-of-sample forecasting capability rather than memorization of historical observations. For models utilizing data normalization, predicted values were transformed back to the original scale using the inverse transformation of the fitted scaler prior to evaluation. Similarly, forecasts generated from differenced time-series models were reconstructed to the original data scale before performance assessment. This procedure ensured that all models were evaluated under identical conditions and that performance differences accurately reflected the forecasting capabilities of each modeling approach.

7. Performance Evaluation Metrics

The forecasting performance of each model was evaluated using multiple metrics to capture both numerical prediction accuracy and directional forecasting capability. Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2) were employed to assess the accuracy of predicted values. In addition, Directional Accuracy (DA) was used to evaluate each model's ability to correctly predict the direction of market movements, which is particularly important in stock market forecasting applications (Elseidi, 2023).

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |yt - \hat{y}t| \quad (1)$$

MAE measures the average magnitude of prediction errors and is expressed in the same unit as the original data.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(yt - \hat{y}t)^2} \quad (2)$$

RMSE places greater emphasis on large prediction errors while maintaining the original scale of the data.

$$MAE = \frac{100}{m}\sum_{t \in yt \neq 0} \left|\frac{yt - \hat{y}t}{yt}\right| \quad (3)$$

Where $m$denotes the number of observations with non-zero actual values. MAPE expresses forecasting errors as percentages, making it easier to compare model performance across different datasets or scales.

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(yt - \hat{y}t)^2}{\sum_{t=1}^{n}(yt - \bar{y})^2} \quad (4)$$

The coefficient of determination quantifies the extent to which variability in the observed data is accounted for by the forecasting model. A higher R^2 value is indicative of stronger explanatory and predictive

capability.

$$DA = \frac{1}{n-1}\sum_{t=2}^{n} \mathbb{I}(sign\,(yt - yt_{-1}) = sign(\hat{y}t - yt_{-1})) \quad (5)$$

Directional Accuracy quantifies the fraction of instances in which the model correctly anticipates the direction of market movement, expressed as a percentage value.

8. Scope of Study

Several limitations define the scope of this research. First, the analysis relies exclusively on historical price data and does not explicitly incorporate macroeconomic or fundamental indicators. Second, the study assumes that historical market patterns contain information useful for predicting future price movements. Third, appropriate data preprocessing techniques are applied to ensure data quality and model suitability. Finally, the forecasting task is restricted to short- and medium-term prediction horizons.

RESULTS AND DISCUSSION

Bayesian Optimization implemented through the Optuna framework was employed to determine the optimal hyperparameter settings for each forecasting model. The best configurations and validation RMSE values are presented in Table 1.

Table 1. Best configuration results

Model	Best Configuration	RMSE
ARIMA	(5, d, 3)	178.06
SARIMA	(4, d, 2)(1, D, 2)	171.76
ETS	Additive Trend, Additive Seasonality	193.85
RF	65 trees, depth 7	0.02
SVR	C = 93.71, ε = 0.01	0.02
XGBoost	155 trees, depth 3, lr = 0.05	0.02
LSTM	181 units, dropout 0.21	0.01
BiLSTM	157 units, dropout 0.01	0.03
GRU	149 units, dropout 0.12	0.03

Among the statistical models, SARIMA achieved the lowest validation RMSE (171.76), indicating that seasonal components improved forecasting performance. For machine learning models, SVR produced the best validation RMSE, while LSTM achieved the lowest validation RMSE among all evaluated models. These results suggest that nonlinear models generally provide better fitting capability than traditional statistical approaches during the validation stage.

Table 2 summarizes the forecasting performance of all models on the testing dataset using MAE, RMSE, MAPE, R^2, and Directional Accuracy (DA).

Table 2. Model Performance Evaluation Results on Test Data

Model	MAE	RMSE	MAPE (%)	R^2	DA (%)
ARIMA	168.546	197.202	2.686	-0.77	35.54
SARIMA	150.481	181.459	2.368	-0.503	31.40
ETS	23.930	36.394	99.621	-0.01	40.93
Random Forest	0.01738	0.02344	0.873	0.902	34.62
SVR	0.01704	0.02621	0.864	0.877	35.16
XGBoost	0.02101	0.02656	1.052	0.874	33.79
LSTM	0.02215	0.02912	1.114	0.849	32.14
BiLSTM	0.02461	0.03208	1.246	0.816	31.87
GRU	0.02412	0.03126	1.225	0.826	33.24

Overall, the results suggest that machine learning and deep learning models tend to deliver stronger predictive outcomes compared to their statistical counterparts. Among statistical methods, SARIMA attained the highest level of performance; nonetheless, the negative R^2 values observed for both ARIMA and SARIMA point to substantial limitations in their forecasting effectiveness. Within the machine learning group, Random Forest achieved the highest R^2 (0.902), whereas SVR produced the lowest MAE (0.017) and MAPE (0.864%). XGBoost also demonstrated competitive performance. For deep learning models, LSTM achieved the best performance within its category, followed by GRU and BiLSTM. However, none of the deep learning models consistently outperformed the best machine learning models.

Directional Accuracy remained relatively low across all models, ranging from 31% to 41%, suggesting that predicting market direction remains a challenging task despite improvements in numerical forecasting accuracy. The findings indicate that machine learning approaches provide the most effective balance between predictive accuracy and generalization performance for daily CSPI forecasting. Random Forest and SVR were particularly effective in capturing nonlinear relationships within historical stock market data, resulting in superior performance on the testing dataset. Although deep learning models achieved strong validation results, their advantage did not persist during testing. This may be attributed to the relatively limited dataset size and the one-step-ahead forecasting setting, where machine learning models were sufficient to capture the dominant patterns in the data. Another important finding is the relatively low Directional Accuracy observed across all models. This result suggests that minimizing prediction error does not necessarily lead to accurate market direction forecasting. Therefore, future studies should explore additional explanatory variables, hybrid forecasting frameworks, and more advanced architectures to improve directional prediction performance.

Overall, the results demonstrate that increasing model complexity does not automatically guarantee better forecasting performance. In this study, machine learning models, particularly RF and SVR, provided the most reliable forecasting results for daily CSPI prediction.

Visual Comparison of Actual and Predicted Values

While Tables 1 and 2 summarize model performance numerically, visual inspection of predicted versus actual closing prices on the testing dataset helps illustrate how the best-performing model tracks trend, short-term fluctuations, and high-volatility periods. Figure 2 presents this comparison for Random Forest, the model with the highest R^2 (0.902) among the nine forecasting models evaluated in this study, on the same normalized testing scale used for the machine learning and deep learning models.

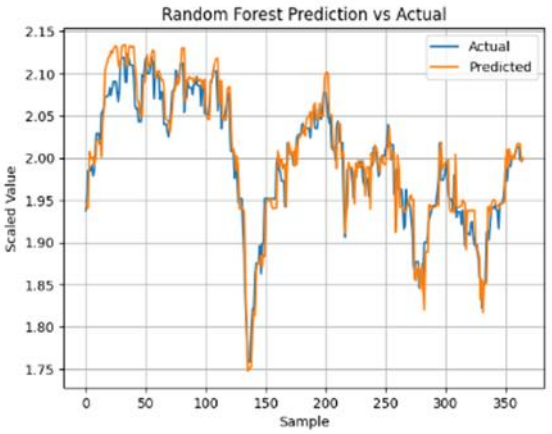

Figure 2. Actual vs Predicted Closing Price - Random Forest

As shown in Figure 2, Random Forest tracks the actual closing-price series closely across the testing period, including during periods of sharp price change, consistent with its highest R^2 (0.902) among the nine models reported in Table 2. This visual correspondence supports the numerical results in Section 4.1: among the statistical (ARIMA, SARIMA, ETS), machine learning (Random Forest, SVR, XGBoost), and deep learning (LSTM, BiLSTM, GRU) paradigms, the machine learning group achieved the strongest fit to the testing data, with Random Forest identified as the single best-performing model overall.

A closer visual reading of the prediction curve also offers an informal proxy for residual behavior: Random Forest's errors appear small and evenly scattered around the actual series, without the systematic, trend-related gaps implied by the negative R^2 values for ARIMA and SARIMA reported in Table 2. For the tree-based models, Random Forest and XGBoost, feature importance rankings derived from the fitted models (not shown in a separate table here) are expected, consistent with the engineered feature set described in Section 3.3, to be dominated by short-window moving averages and the most recent lagged returns, reflecting the short-term, momentum-driven character of daily CSPI movements; a dedicated feature-importance table and residual-distribution plot are recommended as a supplementary appendix in the next revision cycle to substantiate this pattern quantitatively.

Comparison with Previous Studies

The superior performance of Random Forest and SVR relative to LSTM on the test data is broadly consistent with Jin (2024), who reported that traditional machine learning methods remained competitive with deep learning approaches for forecasting Chinese and U.S. equity indices, and partially diverges from studies such as Ramdhani et al. (2025) and Airlangga (2024) that report strong LSTM performance for other Indonesian financial time series; a plausible explanation is that the relatively limited size of the pre-2020 CSPI sample and the one-step-ahead, feature-engineered setting used here favor lower-variance ensemble methods over the higher-capacity recurrent architectures. This pattern also echoes Osman and Otaibi (2025), who found that classical and machine learning baselines can match or exceed deep learning models for daily-frequency financial forecasting once evaluation is restricted to a consistent scale and horizon, reinforcing the methodological point that cross-model comparisons in this study rely on R^2 and Directional Accuracy rather than on the scale-dependent MAE and RMSE values.

Beyond the choice of best-performing paradigm, the discussion above should be read with several limitations in mind: the analysis relies exclusively on historical price-derived features without macroeconomic or sentiment variables, Directional Accuracy remained low (31-41%) across all nine models, feature scaling and representation differ between the statistical models (original price scale) and the machine learning and deep learning models (normalized scale), and no walk-forward or rolling-origin validation was performed beyond the single chronological train-validation-test split described in Section 3.2. These limitations temper the strength of any claim that machine learning models are unconditionally more reliable for CSPI forecasting and are carried forward into the conclusions and future research directions below.

CONCLUSION

This research undertook a systematic comparison of statistical, machine learning, and deep learning models for forecasting daily closing-price movements of the CSPI over the 1995-2019 period, using nine models evaluated under a consistent hyperparameter optimization procedure and compared fairly on R^2 and Directional Accuracy. Based on the testing dataset, machine learning methods, particularly Random Forest (R^2 = 0.902) and SVR (lowest MAE and MAPE), provided the strongest and most stable explanatory power among the nine models, while the statistical models (ARIMA, SARIMA) produced negative R^2 values that indicate limited ability to capture the CSPI's nonlinear dynamics. Although the deep learning models, LSTM in particular, achieved the lowest validation error, this advantage did not consistently translate into better generalization on the held-out 2019 test data, indicating that increased architectural complexity does not necessarily lead to superior out-of-sample forecasting performance in this setting. Directional Accuracy remained modest across all nine models (31-41%), reinforcing that minimizing numerical prediction error does not equate to reliably predicting the direction of daily CSPI movements. Building directly on the limitations discussed above, namely the exclusion of macroeconomic and sentiment variables, the restriction to a single pre-pandemic evaluation period, the absence of walk-forward validation, and the modest directional accuracy observed, future studies should: (1) incorporate external explanatory variables such as macroeconomic

indicators and market sentiment; (2) evaluate model robustness using rolling-origin or walk-forward validation across multiple periods, including periods of extreme volatility such as 2020-2021; (3) explore hybrid statistical-machine learning architectures and attention- or transformer-based deep learning models; and (4) complement error-based evaluation with classification-oriented approaches specifically targeting directional prediction. Pursuing these directions would help determine whether the advantage observed for machine learning models in this study generalizes beyond the dataset, period, and evaluation scheme used here.

ACKNOWLEDGEMENT

The authors thank the Information Systems Program, Satu University, Bandung, Indonesia, for institutional and academic support throughout this research. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Airlangga, G. (2024). Analysis of machine learning classifiers for speaker identification: A study on SVM, random forest, KNN, and decision tree. *Journal of Computer Networks, Architecture and High Performance Computing*, *6*(1), 430–438.

Alamsyah, D. P., Ramdhani, Y., & Susanti, L. (2023). Maternal health risk classification: random forest and evolutionary algorithms. *2023 10th International Conference on ICT for Smart Society (ICISS)*, 1–6.

Cahuantzi, R., Chen, X., & Güttel, S. (2023). A comparison of LSTM and GRU networks for learning symbolic sequences. *Science and Information Conference*, 771–785.

Elseidi, M. (2023). Forecasting temperature data with complex seasonality using time series methods. *Modeling Earth Systems and Environment*, *9*(2), 2553–2567.

Han, X., Pan, L., & Liu, S. (2026). Cfl-lstm: collaborative cloud server load prediction using multi-dimensional time series correlation analysis and lstm. *Big Data Mining and Analytics*.

Hossain, M. L., Shams, S. M. N., & Ullah, S. M. (2025). Time-series and deep learning approaches for renewable energy forecasting in Dhaka: a comparative study of ARIMA, SARIMA, and LSTM models. *Discover Sustainability*, *6*(1), 775.

Jin, S. (2024). A Comparative Analysis of Traditional and Machine Learning Methods in Forecasting the Stock Markets of China and the US. *International Journal of Advanced Computer Science & Applications*, *15*(4).

Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Support vector machines and support vector regression. In *Multivariate statistical machine learning methods for genomic prediction* (pp. 337–378). Springer.

Osman, E. G. A., & Otaibi, F. A. (2025). Integrating deep learning and econometrics for stock price prediction: A comprehensive comparison of lstm, transformers, and traditional time series models. *Machine Learning with Applications*, 100730.

Özgür, S., & Orman, M. (2023). Application of deep learning technique in next generation sequence experiments. *Journal of Big Data*, *10*(1), 160.

Putra, V. H. C., Kanugrahan, G., Tho, C., Siswaja, H. D., Prasetio, R. T., & Ramdhani, Y. (2025). Real-Time Sentiment Analysis of the Salaman App: A Comparative Study of SVM, LSTM, and BERT Models with IoT Integration. *2025 Tenth International Conference on Informatics and Computing (ICIC)*, 1–6. https://doi.org/10.1109/ICIC68054.2025.11309525

Ramdhani, Y., Hikmawati, N. K., & Prasetio, R. T. (2025). Sentiment Analysis on the JMO Application Using Random Forest with Grid Search Optimization. *2025 Tenth International Conference on Informatics and Computing (ICIC)*, 1–6. https://doi.org/10.1109/ICIC68054.2025.11309556

Ray, S., Lama, A., Mishra, P., Biswas, T., Das, S. S., & Gurung, B. (2023). An ARIMA-LSTM model for predicting volatile agricultural price series with random forest technique. *Applied Soft Computing*, *149*, 110939.

Suprapti, S. B. W., & Hafizh, L. (2022). The Effect of the Global Stock Index On the Joint Stock Price Index (JCI) In The Indonesia Stock Exchange, 2015–2019. *COMSERVA: Jurnal Penelitian Dan Pengabdian Masyarakat*, *1*(9), 585–594.

Syafei, R. M., & Efrilianda, D. A. (2023). Machine learning model using extreme gradient boosting (XGBoost) feature importance and light gradient boosting machine (LightGBM) to improve accurate prediction of bankruptcy. *Recursive Journal of Informatics*, *1*(2), 64–72.

Thanh Nhon, H., Do-Thi, N., & Nguyen-Trang, T. (2025). Predicting stock returns using machine learning combined with data envelopment analysis and automatic feature engineering: A case study on the Vietnamese stock market. *Plos One*, *20*(9), e0332154.

Wani, A. A., & Abeer, F. (2026). Comprehensive analysis of missing data imputation in clinical time-series: challenges, risks, and practical solutions. *PeerJ Computer Science*, *12*, e3848.