

STOCK PRICE PREDICTION IN INDONESIA USING EXTREME GRADIENT BOOSTING OPTIMIZED BY ADAPTIVE PARTICLE SWARM OPTIMIZATION

Alya Mirza Safira, Trimono, Kartika Maulida Hindrayani

Data Science Study Program, Universitas Pembangunan Nasional “Veteran” Jawa Timur,
Surabaya, Indonesia

e-mail: trimono.stat@upnjatim.ac.id

DOI: 10.14710/medstat.18.1.105-115

Article Info:

Received: 24 August 2025

Accepted: 13 October 2025

Available Online: 18 October 2025

Keywords:

*Stock Price; Fluctuation;
Prediction; XGBoost; APSO*

Abstract: High volatility is a major problem in generating accurate predictions of stock prices. It also causes unstable predictions and increases the loss risk. Therefore, an adaptive prediction model that is able to adjust to dynamic data pattern changes is needed. This study aims to address these issues by developing an Extreme Gradient Boosting (XGBoost) model optimized using Adaptive Particle Swarm Optimization (APSO). XGBoost was chosen for its ability to handle nonlinear relationships and minimize overfitting, while APSO serves to adaptively adjust parameters to obtain the optimal combination of hyperparameters. The novelty of this research lies in the application of XGBoost-APSO integration in the context of stock price prediction in the Indonesian capital market, which is characterized by high volatility. The study was conducted using daily closing price data of PT Aneka Tambang Tbk (ANTM) shares from November 2020 to May 2025 to predict prices seven days ahead. The results show that the XGBoost-APSO model provides the best performance with a MAPE value of 0.2%, superior to XGBoost-PSO (2.58%) and standard XGBoost (2.91%). This approach effectively improves prediction accuracy and supports quick and accurate investment decision making, while contributing to the development of intelligent prediction systems in the Indonesian capital market.

1. INTRODUCTION

Stocks are one of the most popular investment instruments because they are easily accessible and have the potential to generate profits in a short period of time (Fahrudin et al., 2021; Trimono et al., 2021). By April 2025, the number of stock investors in Indonesia reached 7.0 million SIDs, an increase of 9.7% from the previous year (PT Bursa Efek Indonesia, 2025). This increase indicates the high level of public enthusiasm for the stock market, but the current unstable global economic conditions have made some investors more cautious.

Amidst this situation, the public tends to choose stocks with more promising long-term profit prospects. PT Aneka Tambang Tbk (ANTM) shares have become a popular choice due to the company's strong fundamentals and positive prospects from rising global gold and nickel prices (Profil PT Aneka Tambang Tbk, 2024). However, the high volatility

of ANTM's share price remains a major challenge, caused by global commodity price movements, rupiah exchange rate volatility, mineral export-import policies, and market sentiment towards the mining sector. These fluctuations create short-term uncertainty that could potentially affect investment decisions. Therefore, technical analysis capable of producing accurate stock price estimates is needed to support more accurate and responsive investment decisions in response to market changes.

The main issue raised in this study is the difficulty of producing accurate stock price predictions amid high volatility, particularly for mining sector stocks such as ANTM. To overcome this problem, a historical data-based technical analysis method is needed that is capable of accurately predicting daily closing stock prices for the next seven days. One method that has proven effective in handling nonlinear and complex data patterns is Extreme Gradient Boosting (XGBoost). This algorithm has advantages in terms of accuracy, processing speed, and the ability to control overfitting through regularization and ensemble learning techniques (Damaliana, 2024). The urgency of using XGBoost in this study is based on its ability to learn complex relationships between variables in stock market data and produce efficient prediction models for data with high noise.

However, XGBoost's performance is highly dependent on the selection of optimal hyperparameters. The manual parameter search process is often inefficient and risks producing local solutions. To overcome this, an effective and adaptive optimization approach is used. In this case, Adaptive Particle Swarm Optimization (APSO) was chosen because it is capable of dynamically adjusting particle parameters and exploring a wider search space, thereby avoiding local optimum traps and accelerating the convergence of solutions (Chauhan et al., 2025).

Research on stock price prediction has been conducted using several methods, including K-Nearest Neighbors (KNN), AdaBoost, Support Vector Machine (SVM), Random Forest Regression, and XGBoost. The results show that XGBoost is the most efficient method for predicting future stock prices, with an MSE of 0.004, MAE of 0.014, and an R^2 score of 0.995 (Sharma & Jain, 2023). Research using the XGBoost method was also conducted to predict and analyze the USD exchange rate against the rupiah. The results of the study indicate that the XGBoost method has good forecasting capabilities because it has a MAPE value below 10%, namely at 3.95% for old data and 0.116% for new data (Islam et al., 2021).

Further research by Wu et al. (2022) discusses the optimization of the XGBoost model with the PSO (Particle Swarm Optimization) model to improve accuracy in electricity price predictions. The results show that the prediction accuracy of the XGBoost-PSO model is higher than that of Autoregressive Integrated Moving Average (ARIMA), Random Walks (RW), Long Short-Term Memory (LSTM), and Support Vector Regression (SVR). XGBoost-PSO showed the smallest MRE and RMSE values at with figures of 0.1207 and 29.2056. This research has several areas that need improvement, namely the balance between the number of iterations or particles and the optimal solution. The study (Srivastava et al., 2023) discusses the prediction of cryptocurrency prices (Bitcoin, Dogecoin, and Ethereum) using the XGBoost algorithm optimized with Enhanced PSO. The performance of XGBoost-PSO is compared with DNN, RNN, FORECASTX, HOLTS, Multivariate LSTM, and the proposed system models. Performance evaluation shows that XGBoost-PSO has the lowest RMSE value compared to other models, namely 0.0176, 0.002, and 0.019 for Bitcoin, Dogecoin, and Ethereum, respectively. This study recommends further development of the model for other assets.

Various studies have proven that APSO is superior to conventional PSO. Hossen et al. (2020) show that APSO is capable of increasing convergence speed, prediction accuracy, and reliability in classification and regression models through adaptive parameter adjustment. Another study by Zheng et al. (2021) also states that the application of APSO in machine learning results in a significant increase in model accuracy and stability in nonlinear data. Thus, the integration of XGBoost - APSO is believed to produce a more accurate, adaptive, and overfitting-resistant stock price prediction model, especially for stock data with high volatility characteristics such as ANTM.

2. LITERATURE REVIEW

2.1. Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a decision tree-based machine learning algorithm developed to improve computational efficiency and prediction accuracy (Syafei & Efrilianda, 2023).

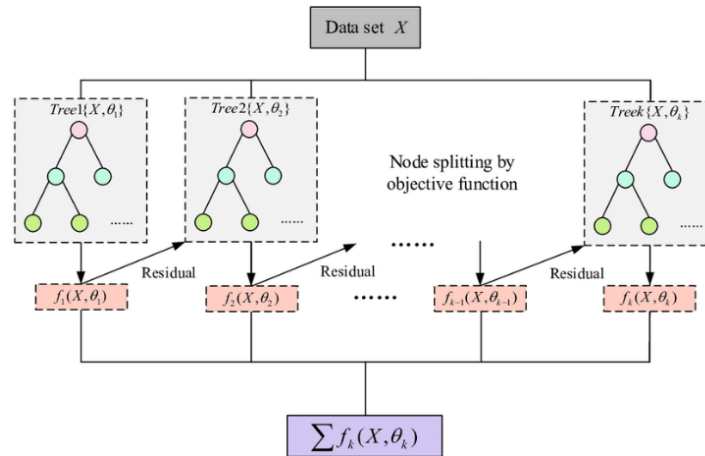


Figure 1. XGBoost Architecture

XGBoost works by building a series of decision trees sequentially, where each new tree aims to improve the prediction error (residual) of the previous tree. This process is known as boosting. The number of trees (k) in the XGBoost model is determined based on the balance between accuracy and model complexity. Too few trees can cause underfitting, while too many can cause overfitting.

The number of trees is generally determined through cross-validation and adjusted according to the learning rate value. The smaller the learning rate, the more trees are needed to achieve optimal convergence. Additionally, early stopping can be used to halt training when accuracy improvements are no longer significant (Chen & Guestrin, 2020). Mathematically, the final prediction of XGBoost is the sum of the predictions from all trees (Fajriyah et al., 2024):

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

with $\hat{y}_i$: prediction for i -th sample; f_k : function of the k -th tree; K : number of trees; F : predicted value.

At each iteration, XGBoost minimizes the objective function, which consists of two components: the *loss* function that measures prediction error (Equation 3) and regularization

that controls model complexity to prevent *overfitting* (Equation 4). The objective function is formulated as (Ardana, 2023):

$$Obj(\theta) = \sum_{i=1}^n L(y_i - \hat{y}_i) + \sum_{k=1}^k \Omega(f_k) \quad (2)$$

with $Obj(\theta)$: objective function; $\sum_{i=1}^n L(y_i - \hat{y}_i)$: sum of loss functions for all samples; $\sum_{k=1}^k \Omega(f_k)$: number of regularization functions for all trees; $\Omega(f_k)$: regularization function for the k-th tree.

The *loss* function in XGBoost is defined as (Ardana, 2023):

$$L(y_i, \hat{y}_i) = \frac{1}{2} \sum (y_i - \hat{y}_i)^2 \quad (3)$$

with $L(y_i, \hat{y}_i)$: Loss function; y_i : actual value; $\hat{y}_i$: predicted value.

The regularization function in XGBoost is defined as (Ardana, 2023):

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2 \quad (4)$$

with $\Omega(f_k)$: regularization function; T : number of leaves on a tree; λ : ridge regression regularization parameter that controls the size of leaf weights; γ : parameter that controls leaf division; ω : predicted value on the leaf

The parameter γ controls the number of leaf splits in each decision tree. The value γ determines the minimum gain threshold for a branch split to be considered feasible. This value is obtained through a hyperparameter tuning process, such as grid search, cross validation, or metaheuristic optimization (e.g., PSO or APSO) to achieve a balance between accuracy and model complexity (Chen & Guestrin, 2020).

This algorithm optimizes the model through gradient calculations and utilizes regularization techniques to improve model generalization. Although XGBoost performs well, selecting optimal hyperparameters such as `learning_rate`, `max_depth`, `n_estimator`, `subsample`, `colsample_bytree`, `gamma`, `regularization_alpha`, and `regularization_lambda` poses a challenge.

2.2. Adaptive Particle Swarm Optimization

Adaptive Particle Swarm Optimization (APSO) is an extension of PSO that adaptively adjusts parameters during the iteration process to improve solution search efficiency and avoid premature convergence (Qin et al., 2021). The main difference between Adaptive Particle Swarm Optimization (APSO) and conventional Particle Swarm Optimization (PSO) lies in the adaptive adjustment mechanism of the inertia parameter and learning coefficient (c_1 and c_2) during the iteration process. In conventional PSO, these parameter values are fixed, which can potentially cause premature convergence or getting stuck at a local optimum.

In contrast, APSO dynamically adjusts the inertia and learning coefficient values based on the particle population conditions at each iteration. This adaptive adjustment makes APSO converge faster, have better exploration capabilities, and be more stable in finding global solutions (Djaneye-Boundjou et al., 2020). APSO works by modeling potential solutions as particles in a search space (Xiao et al., 2021). Each particle has a position (x_i) and velocity (v_i) that are iteratively updated based on two main factors: p_{best} (the best position ever achieved by that particle) and g_{best} (the best position ever achieved by the entire population). At each iteration, the velocity (Equation 5) and particle position (Equation 6) are updated using the following equations (Qin et al., 2021):

$$v_i^{(t+1)} = \omega \cdot v_i^{(t)} + c_1 \cdot r_1 \cdot (p_{best_i} - x_i^{(t)}) + c_2 \cdot r_2 \cdot (g_{best} - x_i^{(t)}) \quad (5)$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (6)$$

with v : particle velocity; ω : inertia factor; c_1, c_2 : acceleration coefficient for p_{best} and g_{best} effects; p_{best_i} : best position of i -th; g_{best} : best global position found by the swarm; r_1, r_2 : random number between $[0,1]$ to increase exploration; x : particle position.

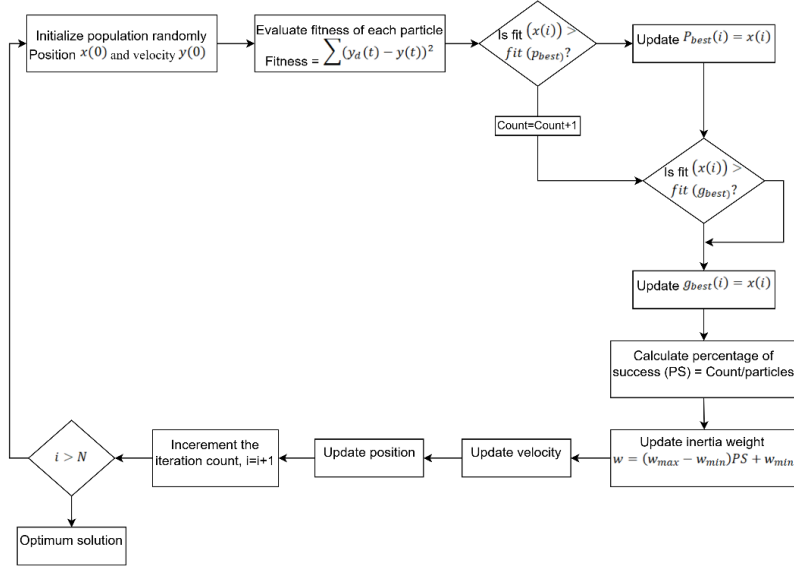


Figure 2. APSO Architecture

The optimization process is carried out by initializing the particle population representing the hyperparameter values, evaluating performance using the MAPE metric, updating the particle position based on the best local and global performance, and iterating until convergence is achieved.

2.3. Extreme Gradient Boosting Optimized with Adaptive Particle Swarm Optimization

The XGBoost-APSO algorithm combines the predictive capabilities of XGBoost and the optimization advantages of APSO. XGBoost builds prediction models incrementally through decision trees, where each new tree improves on the error of the previous tree (Equation 2), and the final result is the sum of all trees (Equation 1). For optimal results, XGBoost requires eight customized hyperparameters. The APSO optimization process begins with the initialization of particles representing hyperparameter combinations. The velocity and position of the particles are updated iteratively (Equations 5 and 6), evaluated using MAPE, and adjusted based on p_{best} and g_{best} . APSO adaptively adjusts the parameters to balance exploration and exploitation until the best hyperparameters for model training are obtained.

3. MATERIAL AND METHOD

3.1. Variables and Data

This study uses historical daily closing price data for PT Aneka Tambang Tbk (ANTM) shares downloaded from Investing.com for the period November 2, 2020 – May 28, 2025, with a total of $\pm 1,100$ observations. The main variable used is the daily closing price of the stock. The input variables are prepared as lag values of the closing price (lag-1, lag-2, ...) as candidate features; the final lag selection is determined based on PACF analysis.

The target variable is the closing price on the following day or at a seven-day prediction horizon.

3.2. Analysis Steps

The analysis steps are as follows: (1) Collection of stock data; (2) Data preprocessing, includes data type conversion, variable selection, missing data handling, and the creation of lag features to capture the temporal dependence in the stock data; (3) Dividing data into training data and testing data with a ratio of 20%:80%; (4) Built the model includes namely conventional XGBoost, XGBoost-PSO, and XGBoost-APSO; (5) The APSO initialization process includes determining the hyperparameter range, defining the objective function, evaluating the particle fitness value, and iteratively updating the personal best (p_{best}) and global best (g_{best}) values until the best hyperparameter combination is obtained; (6) After obtaining the optimal hyperparameters, the XGBoost model was improved using the best configuration and used to predict stock prices; (7) Forecast stock prices for the next seven days based on the accumulated results of the optimized XGBoost model.

4. RESULTS AND DISCUSSION

Descriptive analysis was performed to determine the basic characteristics of the data to be analyzed. Descriptive statistics are presented in Table 1.

Table 1 . Descriptive Statistics

Statistics	Count	Mean	Std. Dev.	Min	Q1	Median	Q3	Max
Closing Price	1100	1960.81	424.488	1105	1620	1965	2295	3190

The closing price of ANTM shares had an average of Rp1960.81 with a standard deviation of Rp424.49, indicating considerable price fluctuations during the observation period. The minimum value of Rp1105 and the maximum of Rp3190 signified a wide range of price movements, illustrating significant stock market volatility. Most of the closing prices were around the median of Rp1965, which is relatively close to the average, so the data distribution can be said to be fairly symmetrical.

The preprocessing stage was carried out by selecting the variables "date" and "last". The name of the variable "last" was changed to "Closing Price" to make it easier to understand. The data type "date" was also changed to datetime to support time series analysis. Next, lag features were formed as determined by the PACF test.

A PACF test was conducted from lag 1 to 30. Lag 1 showed a very high PACF value of 0.9887, far above the significance threshold (0.0626). The RMSE value of the lag test also validated the determination of this lag value (the best RMSE was lag 1 at 0.155). Therefore, lag_1 was selected as the main input feature because it had a strong direct influence on the price on the following day.

The shifting process in creating the lag caused empty values (NaN) to appear in the first row, so that row was deleted. After the entire preprocessing process was complete, the amount of data ready for use in model training was 1,099 rows. The data is then divided into training data (80%) and testing data (20%) to measure the model's generalization ability. The training data range is November 3, 2020-June 26, 2024 (879 rows), while the testing data covers June 27, 2024-May 28, 2025 (220 rows). This separation ensures that the model learns from historical patterns and is tested on representative recent data.

Table 2. Data Preprocessing Results

Date	Lag 1	Closing Price
3/11/20	1100	1110
4/11/20	1110	1105
5/11/20	1105	1125
...	...	...
26/5/25	3130	3110
27/5/25	3110	3030
28/5/25	3030	3110

After splitting the data, the next step is to determine the best hyperparameters to optimize model performance. The hyperparameter values of the three models were obtained through different methods. The conventional XGBoost model uses the default parameters from the XGBoost library without any optimization process, thus serving as a baseline comparison. The XGBoost-PSO model obtains hyperparameter values through a search process using the Particle Swarm Optimization (PSO) algorithm. Meanwhile, the APSO XGBoost model uses an adaptive version of PSO, where the speed and direction parameters of the particles are automatically adjusted during the search process. This approach makes APSO more effective in finding the best parameter combination that produces the highest prediction accuracy. The optimization results show differences in the hyperparameter configuration of each model, as shown in Table 3.

Table 3. Best Hyperparameter Results

XGBoost - APSO		XGBoost - PSO		XGBoost	
Hyperparameter	Value	Hyperparameters	Value	Hyperparameter	Value
N_estimators	100	N_estimators	149	Number of estimators	100
max_depth	3	max_depth	3	max_depth	6
learning_rate	0.067	learning_rate	0.065	learning_rate	0.3
subsample	0.5	subsample	0.6	subsample	1.0
colsample_bytree	1.0	colsample_bytree	0.6	colsample_bytree	1.0
min_split_loss	1.0	min_split_loss	3.3	min_split_loss	0.0
reg_alpha	0.054	reg_alpha	9.334	reg_alpha	0.1
reg_lambda	9.837	reg_lambda	9.808	reg_lambda	1.0

The results in the table show that the XGBoost-APSO model has a more balanced combination of parameters compared to the other two models. The learning rate value of 0.067 indicates that the model learns gradually and carefully, not too quickly like the conventional XGBoost which has a larger learning rate of 0.3. In this way, the model can understand data patterns more stably and avoid errors due to overly aggressive learning. Meanwhile, the smaller max depth value of 3 indicates that the depth of the decision tree built is not too complex. This helps the model avoid "memorizing" the training data, which could potentially cause overfitting, and still be able to provide good results on new data.

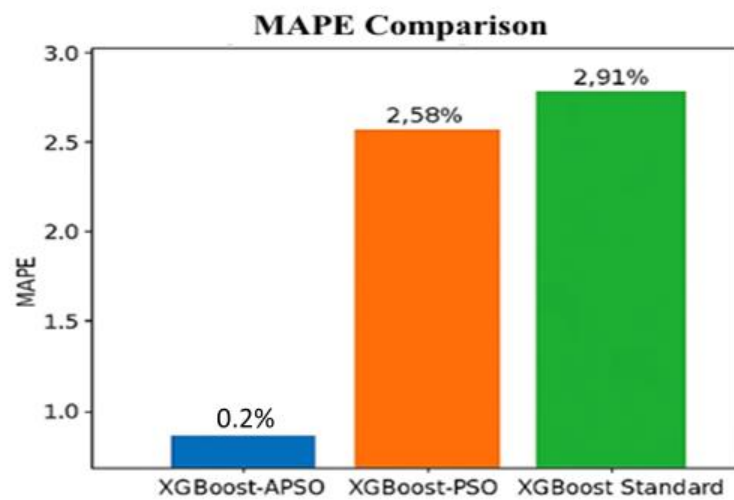
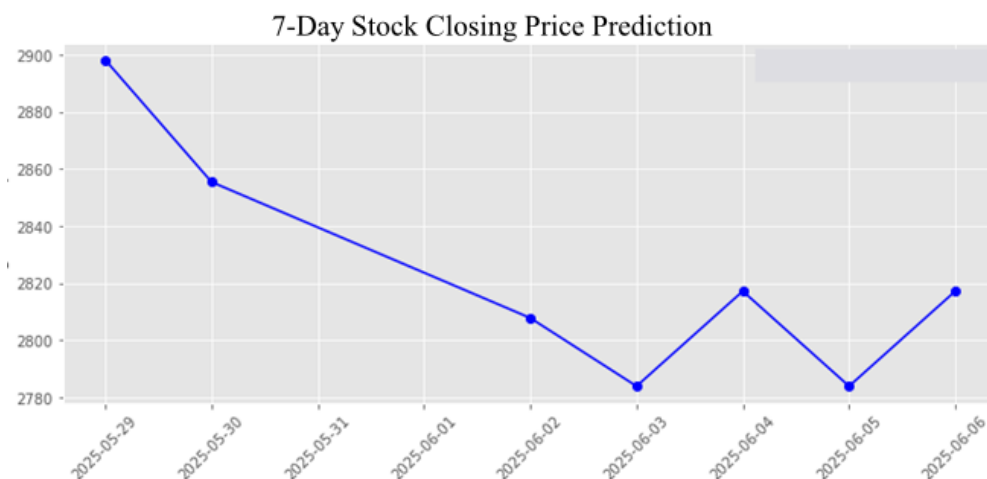
The XGBoost-PSO model has similar results, but the regularization parameter values (reg_alpha and reg_lambda) are higher. This means that PSO tends to emphasize controlling the complexity of the model so that it is not too complicated, but in some cases it can reduce the flexibility of the model. Meanwhile, conventional XGBoost uses default parameters without an optimization process, so it tends to be less efficient in adjusting to the characteristics of volatile stock data. Next is comparing the model's predicted values and actual values.

Table 4. Comparison of Predicted Values and Actual Values

Date	Actual	Predicted	Difference
27/06/24	1230	1221.884888	8.115112
28/06/24	1250	1243.133911	6.866089
01/07/24	1305	1299.555786	5.444214
...	...	...	...
26/5/25	3110	3101.820557	8.179443
27/5/25	3030	3028.072754	1.927246
28/5/25	3110	3102.329590	7.670410

The results in Table 4 show that the predicted values generated by the model are very close to the actual values. The average difference between the actual price and the predicted result is in the range of 1-8 points, which is very small compared to the stock price scale above one thousand rupiah.

After comparing the predicted values with the actual values, the next step is to compare the MAPE values to evaluate the model's performance.

**Figure 3 .** Comparison of MAPE Values for the Three Models**Figure 4 .** Seven-Day Stock Price Prediction

The XGBoost-APSO model has the lowest MAPE value of 0.2%, indicating a very small prediction error rate. The XGBoost-PSO model recorded a MAPE of 2.58%, while the standard XGBoost had the highest MAPE of 2.91%. These results show that the use of

adaptive hyperparameter optimization methods such as APSO can significantly improve model performance. APSO allows for more dynamic parameter adjustments than conventional optimization methods, enabling the model to be more responsive to fluctuating data patterns such as stock price movements. Thus, the XGBoost-APSO model proved to be more effective in producing accurate predictions than the other two models.

After comparing the MAPE values, the next step is to look at the stock price predictions for the next seven days using the best model. The model predicts that ANTM's stock price will experience moderate fluctuations in the next seven days, starting at 2,898 on May 29, 2025, and declining to around 2,784 on June 3, then rising slightly to 2,817 on June 6. This pattern shows relatively stable price movements with a downward trend at the beginning, but accompanied by a potential rebound at the end of the period. These results can be used by investors as a reference for short-term strategies.

In this study, the XGBoost model that has been optimised using APSO was found to demonstrate the highest levels of prediction accuracy in comparison to both the basic XGBoost model and the XGBoost model that has been optimised with PSO. Several previous studies have examined the implementation of the XGboost algorithm for stock price prediction in national and international stock markets. Faqih & Sugihartono, (2025) implemented XGboost to predict Tesla stock prices with an MAE accuracy value of 13.71. Furthermore, Jange (2022) predicted BBKA.JK stock prices using a standard XGBoost model with a MAPE prediction value of 4.01%. Based on this research, it is known that the implementation of XGBoost is still limited to standard models without using optimization methods to determine the best parameters. Therefore, this study implements XGBoost optimized with APSO to fill the gap from previous studies and provide empirical evidence that APSO optimization provides better accuracy than standard models.

5. CONCLUSION

This study shows that the combination of the Extreme Gradient Boosting (XGBoost) algorithm with Adaptive Particle Swarm Optimization (APSO) optimization is capable of providing more accurate and stable stock price prediction results compared to previous studies. The XGBoost-APSO model produced a MAPE value of 0.2% on ANTM stock training and testing data, which is much better than the standard XGBoost (2.91%) and XGBoost-PSO (2.58%) in previous studies. This improved performance occurs because APSO can automatically adjust its search for optimal values so that the model training process does not stop too early before finding the best results. In addition, the selection of the most influential lag based on PACF analysis results makes the model more focused on price change patterns that are truly relevant to past data. These results indicate that combining boosting and adaptive optimization methods is effective for predicting highly volatile stock prices and can help investors and decision makers understand market movements more accurately.

REFERENCES

- Ardana, A. (2023). Performance Analysis of XGBoost Algorithm to Determine the Most Optimal Parameters and Features in Predicting Stock Price Movement. *Telematika*, 20(1), 91. <https://doi.org/10.31315/telematika.v20i1.9329>

- Chauhan, D., Shivani, & Suganthan, P. N. (2025). *Learning Strategies in Particle Swarm Optimizer: A Critical Review and Performance Analysis*. 1–53. <https://arxiv.org/pdf/2504.11812.pdf>
- Chen, T., & Guestrin, C. (2020). XGBoost: A Scalable Tree Boosting System. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Damaliana, A. T. (2024). *Forecasting The Occupancy Rate of Star Hotels In Bali*. 12(1), 24–33. <https://doi.org/10.14710/JSUNIMUS>.
- Djaneye-Boundjou, O., Ordonez, R., & Gazi, V. (2020). Stable Adaptive Particle Swarm Optimization. *International Conference on Control, Automation and Systems*, 39(October), 440–445. <https://doi.org/10.1109/ICCAS.2013.6703971>
- Fahrudin, T. M., Riyantoko, P. A., Hindrayani, K. M., & Diyasa, I. G. S. M. (2021). Daily Forecasting for Antam's Certified Gold Bullion Prices in 2018-2020 using Polynomial Regression and Double Exponential Smoothing. *Journal of International Conference Proceedings*, 3(4), 45–53. <https://doi.org/10.32535/jicp.v3i4.1009>
- Fajriyah, R., Isnandar, H. A., & Arifuddin, A. (2024). Gene Markers Identification of Acute Myocardial Infarction Disease Based on Genomic Profiling Through Extreme Gradient Boosting (XGBoost). *Media Statistika*, 17(1), 69–80. <https://doi.org/10.14710/medstat.17.1.69-80>
- Faqih, A., & Sugihartono, T. (2025). Perbandingan Algoritma XGBoost dan LSTM dalam Prediksi Harga Saham Tesla Menggunakan Data Tahun 2025. *Jurnal Pendidikan Dan Teknologi Indonesia*, 5(6), 1563–1573. <https://doi.org/10.52436/1.jpti.836>
- Hossen, M. S., Rabbi, F., & Rahman, M. M. (2020). Adaptive Particle Swarm Optimization (APSO) for Multimodal Function Optimization. *International Journal of Engineering and Technology*, 1(3), 98–103.
- Islam, S. F. N., Sholahuddin, A., & Abdullah, A. S. (2021). Extreme Gradient Boosting (XGBoost) Method in Making Forecasting Application and Analysis of USD Exchange Rates Against Rupiah. *Journal of Physics: Conference Series*, 1722(1). <https://doi.org/10.1088/1742-6596/1722/1/012016>
- Jange, B. (2022). Prediksi Harga Saham Bank BCA Menggunakan XGBoost. *Arbitrase: Journal of Economics and Accounting*, 3(2), 231–237. <https://doi.org/10.47065/arbitrase.v3i2.495>
- Profil PT Aneka Tambang Tbk. (2024). *Profil PT Antam Tbk*. <https://www.antam.com/id/about-us/profile>

- PT Bursa Efek Indonesia. (2025). *IDX Bell Laporan Tahunan Rapat Umum pemegang Saham Tahunan 2025 dan 17 Juta Investor Pasar Modal Indonesia*. <https://www.idx.co.id/id/tentang-bei/idxbell-newsletter/>
- Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P., & Liu, P. (2021). XGBoost Optimized by Adaptive Particle Swarm Optimization for Credit Scoring. *Mathematical Problems in Engineering*, 2021(5), 1-18. <https://doi.org/10.1155/2021/6655510>
- Sharma, P., & Jain, M. K. (2023). Stock Market Trends Analysis using Extreme Gradient Boosting (XGBoost). *Proceedings - 4th IEEE 2023 International Conference on Computing, Communication, and Intelligent Systems, ICCIS 2023, February*, 317–322. <https://doi.org/10.1109/ICCIS60361.2023.10425722>
- Srivastava, V., Dwivedi, V. K., & Singh, A. K. (2023). Cryptocurrency Price Prediction Using Enhanced PSO with Extreme Gradient Boosting Algorithm. *Cybernetics and Information Technologies*, 23(2), 170–187. <https://doi.org/10.2478/cait-2023-0020>
- Syafei, R. M., & Efrianda, 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. <https://doi.org/10.15294/rji.v1i2.71229>
- Trimono, T., Diyasa, I. G. S. M., Fauzi, A., & Idhom, M. (2021). Value at Risk with Bootstrap Historical Simulation Approach for Prediction of Cryptocurrency Investment Risk. *Proceedings - 2021 IEEE 7th Information Technology International Seminar, ITIS 2021*. <https://doi.org/10.1109/ITIS53497.2021.9791657>
- Wu, K., Chai, Y., Zhang, X., & Zhao, X. (2022). Research on Power Price Forecasting Based on PSO-XGBoost. *Electronics (Switzerland)*, 11(22). <https://doi.org/10.3390/electronics11223763>
- Xiao, Y., Yan, W., Wang, R., Jiang, Z., & Liu, Y. (2021). Research on Blank Optimization Design Based on Low-Carbon and Low-Cost Blank Process Route Optimization Model. *Sustainability (Switzerland)*, 13(4), 1–21. <https://doi.org/10.3390/su13041929>
- Zheng, D., Qin, C., & Liu, P. (2021). Adaptive Particle Swarm Optimization Algorithm Ensemble Model Applied to Classification of Unbalanced Data. *Scientific Programming*, 2021, 1–13. <https://doi.org/10.1155/2021/7589756>