Technical analysis model for stock prediction using a grammatical evolution algorithm

Aditya Kusuma Setyanegara, Imas Sukaesih Sitanggang, Mushthofa
School of Data Science, Mathematics and Informatics, IPB University, Bogor, Indonesia

Article Info

Article history:

Received Jan 11, 2025
Revised Jan 5, 2026
Accepted Jan 22, 2026

Keywords:

ARIMA
Exponential smoothing
Fibonacci retracements
Grammatical evolution
Prophet
Technical analysis

ABSTRACT

Stocks are a popular investment instrument but carry high risks, where investors may incur losses when stocks are bought at high prices and sold at lower prices. Technical analysis is used to study past stock price behavior to predict future prices. In this study, grammatical evolution (GE) is applied as an evolutionary computing technique to discover optimal functions or programs that represent historical stock price data. This study develops GE-based prediction models by utilizing objective functions and search spaces defined through grammar. The model integrates technical indicators based on complex statistical models such as autoregressive integrated moving average (ARIMA), prophet, exponential smoothing, and Fibonacci retracements. Furthermore, this study employs GE to generate ensemble weights randomly, ensuring each model contributes equitably to the final prediction formula. Experiments were conducted using multiple stock datasets, including SMAR, S&P 500, the Johannesburg Stock Exchange (JSE), the New York Stock Exchange (NYSE), and Adani Enterprises (ADANIENT), to evaluate the model's adaptability and generalization capability. The results demonstrate that the proposed GE model effectively captures complex market patterns and produces more reliable stock price predictions compared to deep learning-based approaches. Although GE requires greater computational time, the findings suggest that GE provides a flexible and effective framework for constructing hybrid stock price forecasting models in dynamic market environments.

Corresponding Author:

Imas Sukaesih Sitanggang
School of Data Science, Mathematics and Informatics, IPB University, Bogor, Indonesia
St. Meranti Wing 20 Level V, Bogor-16680, Indonesia
Email: imas.sitanggang@apps.ipb.ac.id

1. INTRODUCTION

Stocks can represent an individual's or an entity's ownership in a company or a limited liability corporation. In this context, financial analysts apply fundamental and technical analysis, statistical experts focus on time series analysis, and computer science researchers adopt machine learning approaches [1]. Stock investors typically conduct in-depth analysis before executing transactions to maximize profits and minimize losses to maximize profits and minimize losses. Technical analysis is a commonly applied method that rests on the belief that past market movements are likely to reappear, as human behavior tends to be consistent across similar situations. As a result, trends and price patterns are viewed as fractal in nature [2].

Several studies have used grammatical evolution (GE) to develop strategies for predicting stock price movements, among them [3]–[9]. Megane *et al.* [4] introduced probabilistic structured grammatical evolution (PSGE), combining structured grammatical evolution (SGE) and probabilistic grammatical evolution (PGE) to guide rule selection using probabilistic grammar updates, showing improved performance over GE and PGE but not consistently outperforming SGE. Meanwhile, Pawlak and O'Neill [7] introduced the grammatical evolution for constraint synthesis (GECS) algorithm to automate constraint synthesis in mixed-integer linear programming (MILP) models using GE and Zuse Institute Mathematical Programming Language (ZIMPL) grammar. However, it is effectiveness depends heavily on the completeness of input data and may still require iterative refinement [7]. Additionally, Ryan *et al.* [10] compared various smoothing methods, including moving average and exponential smoothing, and suggested that combining autoregressive integrated moving average (ARIMA) with exponential smoothing could enhance their model's effectiveness.

Recent studies have increasingly investigated neural forecasting and hybrid ensemble approaches for stock price prediction, including hybrid ARIMA–long short-term memory (LSTM) models and transformer-based approaches for time series forecasting, along with performance comparisons against advanced deep learning architectures like temporal fusion transformers (TFT), N-BEATS, and informer. Nti *et al.* [11] evaluated ensemble learning methods bagging, boosting, stacking, and blending using decision tree (DT), support vector machine (SVM), and neural network models on stock data from multiple exchanges, finding that stacking and blending achieved higher predictive accuracy but incurred high computational costs and exhibited dependence on dataset characteristics. Then, Aggrawal and Dhawan [12] compared deep learning models (TFT, N-BEATS, temporal convolution network (TCN)) with LSTM and gated recurrent units (GRU) for short-term prediction, revealing that LSTM and GRU were more stable and accurate, while newer architectures struggled to generalize across multivariate data. Subsequently, Li and Pan [13] proposed a blending ensemble combining LSTM and GRU with a fully connected neural network meta-learner, using quantitative and sentiment-based information from financial news to predict the S&P 500. However, their model was limited by the short six-month dataset and it is focus on a single index. Meanwhile, Abdulrahman *et al.* [14] proposed a hybrid ARIMA-LSTM model enhanced by discrete Fourier transform (DFT) decomposition to separate linear and nonlinear components, achieving improved accuracy over standalone models but lacking statistical validation and cross-market testing.

GE represents a promising branch of genetic programming that employs an evolutionary algorithm–based search mechanism together with domain-specific grammar specifications expressed in Backus-Naur form (BNF) to generate symbolic expressions [3]. This research aims to implement ARIMA, prophet, exponential smoothing, and Fibonacci retracements methods within the GE algorithm to produce more complex and accurate technical analysis indicators. This study is expected to provide significant benefits in confirming the effectiveness of GE as a tool for generating innovative stock indicators and creating a higher-accuracy stock prediction system creating a higher accuracy stock prediction system. Moreover, the findings from this research opens avenues for further in-depth research in developing more robust and accurate stock prediction systems.

2. METHOD

This chapter discusses the methodology for designing a technical analysis model for stock prediction systems using the GE algorithm. The methodological framework details the research process, which includes data collection, preprocessing, model construction, training and testing phases, and the evaluation and analysis of the obtained results. Through these stages, the research methodology provides a comprehensive guide for designing, training, and testing a technical analysis model using the GE algorithm. Figure 1 presents research stages diagram, summarizing steps from data collection to performance analysis.

This research is based on stock data from PT Sinar Mas Agro Resources and Technology (PT SMART Tbk) with the stock code SMAR.JK, sourced from Yahoo Finance, covers January 2022 to January 2023. The data used is the closing price (close price) of the stock, as it is considered the most stable and widely referenced price that reflects market consensus at the end of each trading day. This research will focus on developing a comprehensive prediction model but needs to include the practical application of the model itself.

In the preprocessing stage, a min–max scaling approach is employed to normalize the data, where feature values are mapped into the [0, 1] interval. Feature normalization is needed to eliminate the effect of several quantitative features measured on different scales [15]. The dataset is partitioned into two subsets, with 80% allocated for training (from January 1, 2020 to January 14, 2023) and 20% allocated for testing (from January 15, 2023 to October 31, 2023). The training data helps the model recognize patterns, while the testing data evaluates its predictive accuracy.

The BNF grammar is designed to define the syntactic rules that shape potential programs in the population [16]. This includes basic operators, mathematical functions, and features such as ARIMA, prophet, exponential smoothing, and Fibonacci retracements. The grammatical rules help structure the functions created by GE.

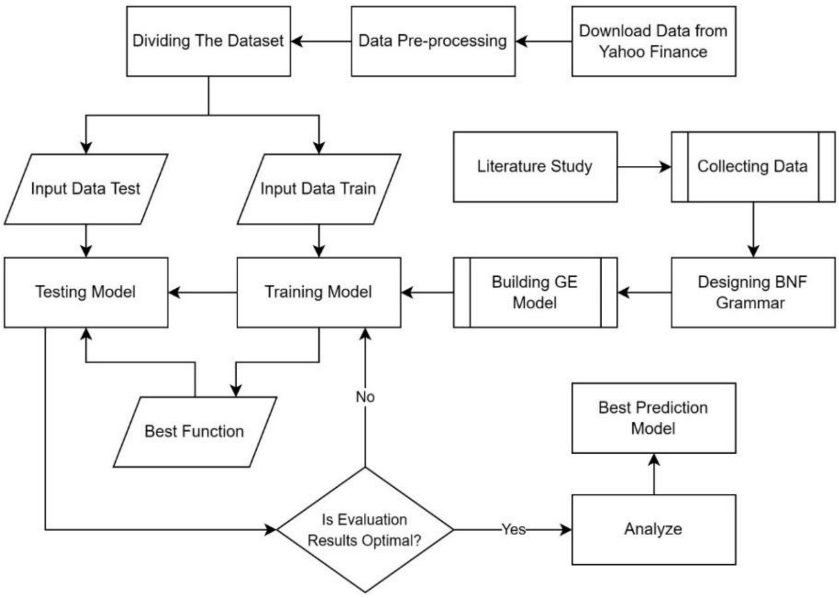

Figure 1. Research stages diagram

This study employs two BNF, the first BNF is utilized for preprocessing stock data by converting the raw data into a structured form that is appropriate for analytical purposes. Additionally, this BNF is designed to reduce noise in highly volatile data, thereby improving data quality and supporting more accurate predictions. The detailed design of the first BNF is presented in Table 1.

The second BNF is developed to construct predictive models by integrating multiple forecasting methods, including ARIMA, prophet, exponential smoothing, and Fibonacci retracements. This integration aims to produce a robust and adaptive model capable of handling complex stock data patterns. The detailed design of the second BNF is presented in Table 2. The study is conducted under two primary scenarios: using only the second BNF and combining both BNFs to optimize the analysis and prediction process.

Table 1. Backus-Naur form data transformation

<table>
<tr><th>Non-terminal</th><th></th><th>Production rules</th></tr>
<tr><td><start></td><td>::=</td><td><expression></td></tr>
<tr><td><expression></td><td>::=</td><td><expression> <operator> <expression> | <function> (<expression>) | <variable> | <constant></td></tr>
<tr><td><operator></td><td>::=</td><td>+ | - | * | /</td></tr>
<tr><td><function></td><td>::=</td><td>sin | cos | tan | sqrt | abs</td></tr>
<tr><td><variable></td><td>::=</td><td>y_{t-1} | y_{t-2} | y_{t-3} | y_{t-4} | y_{t-5} | y_{t-6} | y_{t-7} | y_{t-8} | y_{t-9}</td></tr>
<tr><td><constant></td><td>::=</td><td>0 | <digit> | <digit> <constant></td></tr>
<tr><td><digit></td><td>::=</td><td>1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9</td></tr>
</table>

Table 2. Backus-Naur form forecasting model

<table>
<tr><th>Non-terminal</th><th></th><th>Production rules</th></tr>
<tr><td><start></td><td>:=</td><td><expression></td></tr>
<tr><td><expression></td><td>:=</td><td><expression><operator><expression> | <variable> | (0.<digit>*<arima>) | (0.<digit>*<prophet>) | (0.<digit>*<expsmooth>) | <fibonacci></td></tr>
<tr><td><operator></td><td>:=</td><td>+ | -</td></tr>
<tr><td><variable></td><td>:=</td><td>$y_{t-\text{<constant>}}$</td></tr>
<tr><td><constant></td><td>:=</td><td>0 | <digit> | <digit> <constant></td></tr>
<tr><td><digit></td><td>:=</td><td>1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9</td></tr>
<tr><td><arima></td><td>:=</td><td>ARIMA([data_train], [data_test_length], [days_interval], <digit>, <digit>, <digit>)</td></tr>
<tr><td><prophet></td><td>:=</td><td>Prophet([data_train], [data_test_length])</td></tr>
<tr><td><expsmooth></td><td>:=</td><td>ExpSmoothing ([data_train], [data_test_len], 0.<digit>, 0.<digit>)</td></tr>
<tr><td><fibonacci></td><td>:=</td><td>FibRetracements([data_train], <constant>, <constant>, <constant>)</td></tr>
</table>

Model construction involves initializing the population, translating chromosomes, evaluating fitness, and applying genetic operators such as crossover and mutation to produce new generations. The translated chromosomes are represented as functions and assessed using root mean square error (RMSE) to measure prediction accuracy against actual values. To ensure valid and efficient chromosome translation, a duplicate and prune mechanism is applied during the mapping process. The duplicate step handles invalid chromosomes by regenerating valid ones, while the prune step marks the last used element to indicate the end of a valid expression, effectively ignoring unused codons to improve translation efficiency and reduce computational overhead. These functions evolve across generations until an optimal solution is achieved. The highest fitness value is considered the optimal solution; a; a lower RMSE value results in a higher fitness score, indicating a model with better predictive capability [17]. Figure 2 illustrates the GE diagram, providing a detailed visualization of the evolutionary process and its components, including initialization, translation, fitness evaluation, and the application of genetic operators. In the training process, functions are progressively refined based on historical stock data to identify the best-performing model, which is subsequently tested on unseen data using mean absolute percentage error (MAPE) as the evaluation metric. Finally, parameter optimization and computational efficiency are analyzed.

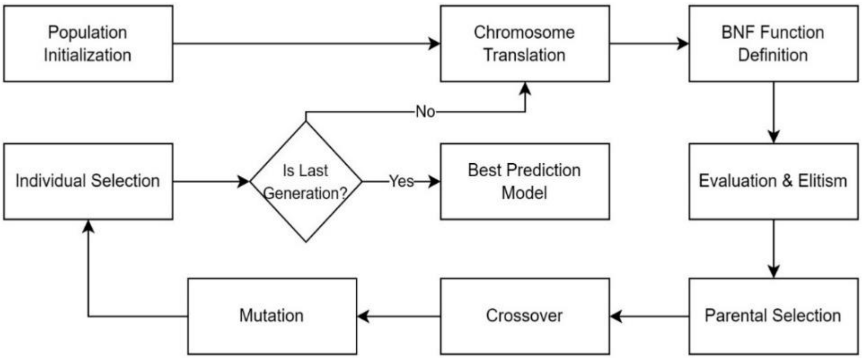

Figure 2. Grammatical evolution diagram

3. RESULTS AND DISCUSSION

This section presents the experimental results and evaluation metrics used in the proposed prediction framework. All experiments in this study were conducted using Python 3.11.0, along with the scikit-learn, ARIMA, prophet, and exponential smoothing libraries. The primary dataset was divided into a training set containing 745 samples (80% of the total data) and a testing set with 187 samples (20% of the total data) to train the model and assess its performance. Model performance was measured using RMSE, R^2 score, and MAPE. The prediction results are summarized in the Table 3. In this study, a comparative analysis was conducted on the performance of the GE model alongside three other prediction models: ARIMA [18]–[20], prophet [21]–[24], exponential smoothing [25], [26], and Fibonacci retracements [27]. Each model was evaluated using the same stock data, with identical training and testing data splits, and consistent preprocessing methods.

3.1. Evaluation of the best individual

This study evaluates individuals in the GE model using RMSE as the primary fitness metric to measure prediction error. At the end of each generation, model performance is further assessed using MAPE and the R^2 score, where MAPE measures the average percentage error and R^2 indicates the degree of variance in the dataset that can be attributed to the model. The equations used in this evaluation are shown in (1) to (6) [28]–[32].

$$Fitness = \frac{1}{RMSE - \varepsilon} \quad (1)$$

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

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| x\ 100\% \quad (3)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (4)$$

$$SS_{res} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (5)$$

$$SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \quad (6)$$

The terms used in the equations are defined as follows: ε is a small constant added to avoid division by zero, n denotes the total number of data points, y_i represents the actual values, and $\hat{y}_i$ indicates the predicted values generated by the regression model. For the R^2 metric, $\bar{y}_i$ is the mean of the observed values, while SS_{res} and SS_{tot} denote the residual and total sums of squares, respectively. These definitions collectively support the evaluation of model performance.

Table 3. Summary of prediction results

Model	Running time (s)	R^2 score	RMSE (Rp)	MAPE	Parameter
Simple exponential smoothing	0.5	-0.15	396.85	0.07	α =0.913
Double exponential smoothing	0.6	0.37	293.84	0.04	α =0.6; β =0.2
Holt's winter Exponential smoothing	0.8	0.36	294.91	0.05	Additive+Damped α =0.6; β =0.15; γ =0.15
Facebook prophet	1.5	-0.528	456.55	0.077	seasonality='multiplicative'; daily_seasonality = True; weekly_seasonality = False; yearly_seasonality = True
ARIMA	1	-0.302	421.4	0.078	diff =30 days; (p, d, q) =(1, 0, 0)
Grammatical evolution	6129	0.784	171.71	0.0281	Population =50; Generation =20; Crossover =0.9; Mutation =0.5; Selection Method = Roulette Wheel;

3.2. Optimal chromosome and translation process

In this study, the search for the optimal chromosome begins with an initial population of 100 chromosomes. Each chromosome is then translated into a mathematical formula, and the resulting formula is tested using test data. The chromosome with the highest fitness value, indicating the best performance based on the test data, is designated as the optimal chromosome.

If the translation process results in an invalid formula, the chromosome will be duplicated up to a maximum limit of 600 chromosomes. If the duplication exceeds this limit without producing a valid formula, a new chromosome is generated to replace the invalid one. This approach aims to maintain the quality of the chromosome population and ensure that the translation process produces reliable solutions.

The translation process aims to produce the most accurate prediction formula. Once a valid formula is generated, pruning is applied to remove irrelevant or redundant chromosomes, simplifying the model and improving efficiency by retaining only components that significantly contribute to prediction accuracy. As shown in Table 4, translation is performed by modulating each chromosome value according to the BNF grammar rules, where each modulation result determines the applied expression rule. This process continues until all chromosome values are used or a valid formula is obtained.

Table 4. The best model prediction

Best chromosome	Best model prediction
[22, 49, 84, 21, 48, 95, 99, 38, 78, 6, 49, 72, 25, 39, 29, 48, 49, 41, 47, 10, 11, 42, 41, 35, 24, 84, 66, 78, 37, 58, 83, 2, 41, 35, 86, 51, 31, 2, 62, 84, 94, 80, 41, 73, 41, 29, 12, 62, 77, 2, 27, 86, 91, 31, 80, 41, 75, 7, 75, 50, 80, 65, 56, 64, 44, 19, 71, 7, 52, 22, 98, 21, 46, 57, 11, 21, 34, 83, 44, 41, 83, 29, 62, 24, 30, 80, 12, 59, 96, 94, 57, 45, 21, 86, 96, 1, 18, 89, 30, 77, 1, 87, 57, 15, 62, 95, 37, 8, 55, 57, 22, 26, 11, 12, 63, 36, 24, 62, 42, 3, 41, 44, 83, 34, 53, 43, 57, 46, 79, 88, 40, 35, 88, 32, 19, 7, 24, 53, 65, 14, 99, 31, 47, 37, 41, 80, 22, 91, 18, 3, 1, 47, 8, 98, 20, 77, 73, 56, 90, 17, 84, 62, 12, 88, 89, 33, 69, 88, 58, 61, 83, 37, 25, 66, 87, 58, 89, 58, 44, 15, 47, 11, 36, 49, 31, 53, 37, 39, 100, 62, 76, 1, 72, 82, 81, 94, 60, 60, 65, 5]	(((0.7*ma. Model arima (data_saham, 187, 1, 4, 3, 4))+((0.5*mes.holt_expsmooth (data_saham, 187, 0.7, 0.6))+(0.1*ma.model_arima(data_saham, 187, 1, 6, 3, 2))))/1.3

In this formula, the weights 0.7, 0.5, and 0.1 represent the contributions of each model, specifically ARIMA and exponential smoothing. The total weight is 1.3, obtained by summing the individual weights. The final forecast is generated through a weighted integration of predictions from individual models, followed by normalization using the total weight. This approach aligns with the ensemble concept, combining multiple predictive models combining multiple predictive models to enhance accuracy.

The ensemble method in this context utilizes a weighted average of multiple model predictions, a technique that has been employed since early research on ensemble models [33], [34]. Each output from the base models is assigned a weight based on its performance criteria, with the total weight of all models summing to one [35]. The performance evaluation of the resulting formula yielded results: R^2 score: 0.7838; RMSE: 171.7123; MAPE: 0.0281.

3.3. Results analysis and parameter tuning

After identifying the best chromosome, parameter tuning was conducted using the grid search method to optimize model performance. Various parameter combinations were evaluated by varying population size (10, 50, 100), number of generations (10, 30, 50), crossover probability (0.2, 0.5, 0.8), mutation probability (0.2, 0.5, 0.8), and parent selection methods (roulette wheel, rank-based, and tournament selection) [36]–[38]. Experimental results demonstrated that each parameter combination significantly influenced model performance and computational efficiency. In the first trial, GE achieved an R^2 of 0.7634, RMSE of 179.6632, and MAPE of 0.0237, with a computation time of 4707 seconds. The second trial showed slight improvement with an R^2 of 0.7638, RMSE of 179.5129, and a lower MAPE of 0.0234, with a computation time of 2844 seconds. The third trial delivered the best performance with an R^2 of 0.7838, RMSE of 171.7123, and MAPE of 0.0281 in a computation time of 6129 seconds.

The fourth trial, using a population size of 100 and 50 generations, required 26,845 seconds but showed no significant accuracy improvement (R^2 =0.7671, RMSE =178.2318, MAPE 2=0.025). This suggests that increasing population size and generations can raise computational costs without proportional performance gains. The best configuration used a population size of 50, 20 generations, a crossover probability of 0.9, a mutation probability of 0.5, and roulette wheel selection, achieving the highest R^2 with relatively low computation time. Figure 3 compares the predicted and actual stock prices, demonstrating the model's performance.

Figure 3. The best prediction model

However, the roulette wheel method has limitations in small populations or when crossover and mutation probabilities are low. Under these conditions, high-fitness individuals tend to be repeatedly selected, resulting in identical chromosomes and reduced population diversity, which can lead to suboptimal solutions. Therefore, careful parameter selection and appropriate parent selection methods are required to maintain diversity. Alternative approaches such as tournament or rank-based selection, along with diversity-preserving techniques, can help sustain exploration and prevent premature convergence.

Overall, the results show that appropriate parameter tuning can substantially improve GE performance despite its high computational cost. The findings highlight the importance of balancing exploration and exploitation to achieve robust predictive accuracy and provide guidance for developing more efficient and adaptive methods. Figures 4 to 7 present alternative models generated using different parameter configurations, illustrating the impact of parameter variations on model performance.

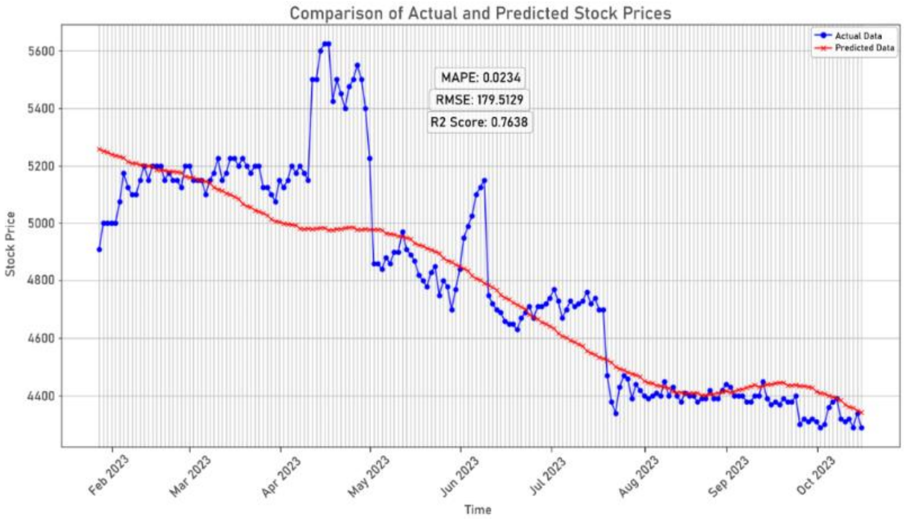

Figure 4. Alternative models of various parameter configurations

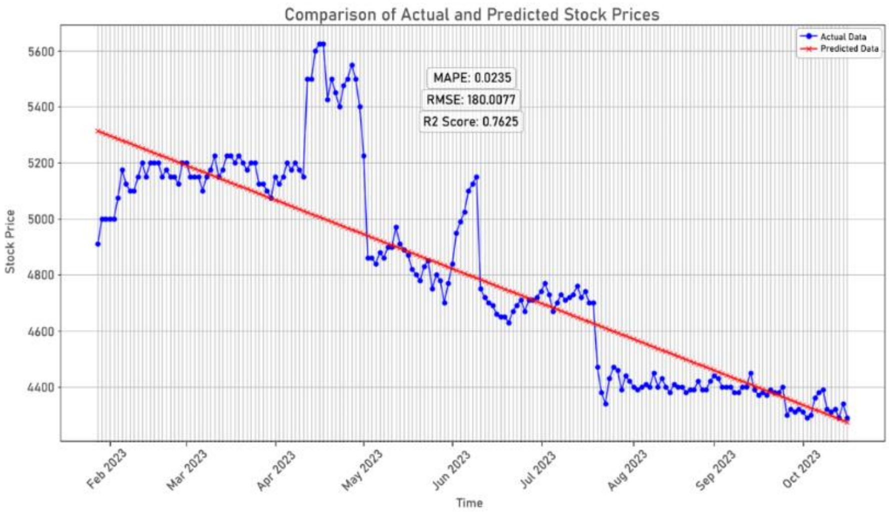

Figure 5. Alternative models of various parameter configurations

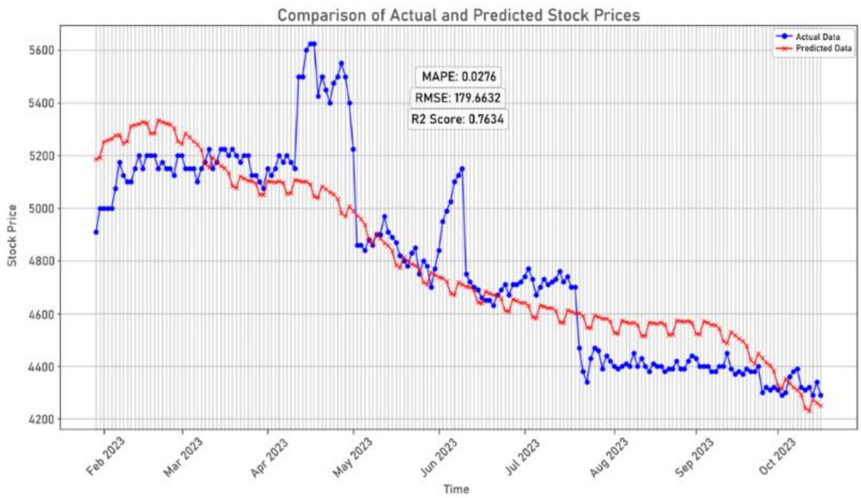

Figure 6. Alternative models of various parameter configurations

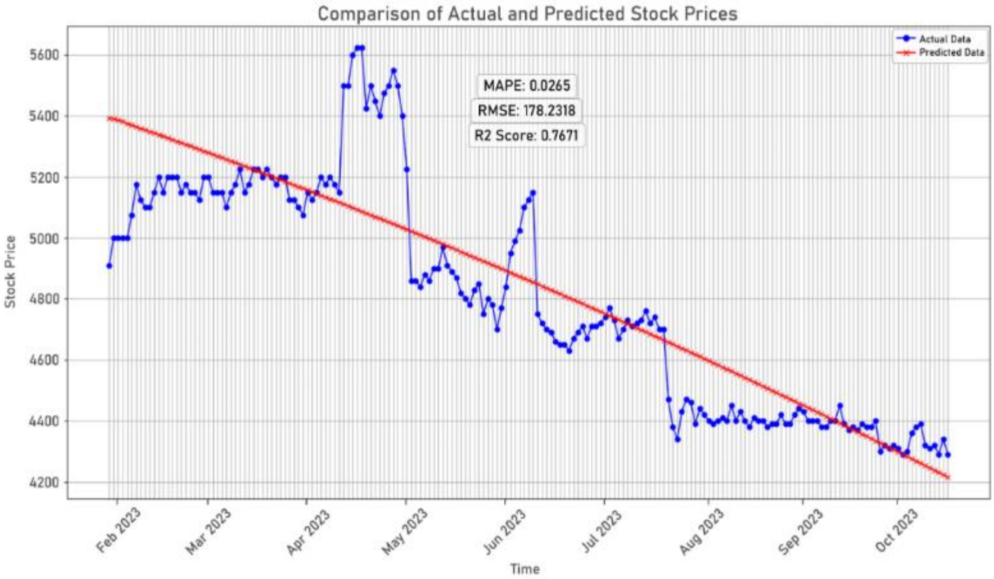

Figure 7. Alternative models of various parameter configurations

After evaluating the proposed GE model on the SMAR stock dataset, further experiments were conducted on several additional stock datasets used in previous studies to assess the model's generalization ability and enable fair comparison. Table 5 compares the prediction performance of the proposed GE model with that of Li and Pan [13] on the S&P 500 Index dataset using R^2, RMSE, and MAPE. The proposed GE model was applied to stock price prediction on the Johannesburg Stock Exchange (JSE) and the New York Stock Exchange (NYSE), with performance evaluated using k-fold cross-validation to ensure reliable results. The evaluation metrics for both datasets are summarized in Table 6. Table 7 compares the proposed GE model with the results reported by Aggrawal and Dhawan [12] for the Adani Enterprises (ADANIENT) stock using R^2, RMSE, and MAPE.

Table 5. Comparison of prediction results between GE and the study by Li and Pan [13]

Study	Model	R^2 Score	RMSE	MAPE
Li and Pan [13]	LSTM	Not reported	20.96	Not reported
	DP-LSTM	Not reported	18.18	Not reported
	GRU	Not reported	15.79	Not reported
	Averaging ensemble	Not reported	15.21	Not reported
	Weighted avg. ensemble	Not reported	15.15	Not reported
	Blending ensemble	Not reported	13.65	Not reported
This study	GE	0.1842 (avg)	29.55 (avg)	1.36 (avg)

Table 6. Performance of GE model on JSE and NYSE datasets using k-fold cross-validation

Stock exchange		R^2 score	RMSE	MAPE
JSE	Min	-0.0044	1070.8136	0.0567
	Max	0.4267	2171.5377	0.1263
	Avg	0.1540	1802.93	0.0957
NYSE	Min	-0.0873	1.4681	0.0377
	Max	0.4494	2.0631	0.0546
	Avg	0.2604	1.6970	0.0444

Table 7. Comparison of GE results with Aggrawal and Dhawan [12] for ADANIENT stock

Study	Model	R^2 Score	RMSE	MAPE (%)
This study	GE	0.5947	230.87	5.62
Aggrawal and Dhawan [12]	LSTM	0.93	15.38	0.54
	GRU	0.95	13.59	0.51
	TFT	0.18	40.67	1.66
	TCN	-4.06	58.52	2.37
	N-BEATS	-122.3	652.39	42.42

4. CONCLUSION

This study demonstrates that the GE method effectively generates accurate and reliable predictive models by integrating complex statistical techniques and technical indicators. The best performance was achieved using a GE configuration with a population size of 50, 20 generations, a crossover probability of 0.9, and a mutation probability of 0.5, yielding an R^2 of 0.7838, RMSE of 171.7123, and MAPE of 0.0281 indicating superior predictive accuracy compared to traditional methods. These findings highlight GE's potential as an alternative approach in stock price prediction, offering adaptive and efficient formulas in response to market dynamics. Furthermore, additional experiments conducted on various stock datasets including those from the S&P 500 Index, JSE, NYSE, and ADANIENT further validate the versatility of the proposed GE model. Although the model performance varies across different markets, the results consistently demonstrate GE's capability to adapt to diverse financial data characteristics while maintaining competitive predictive accuracy compared to advanced deep learning and ensemble-based approaches reported in previous studies. Future work should focus on automated parameter optimization using techniques like random search, Bayesian optimization, or genetic algorithms, and explore deeper tuning of parameters in ARIMA, prophet, and exponential smoothing models to enhance model responsiveness. Expanding GE testing across various industrial sectors or global markets is also crucial to validate its generalizability and robustness. Ultimately, this research underscores the promise of evolutionary computation in financial market analysis and paves the way for the development of more adaptive and reliable predictive algorithms.

FUNDING INFORMATION

The publication cost was partially supported by School of Data Science, Mathematics and Informatics IPB University and partially funded by the author's personal contribution.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Aditya Kusuma Setyanegara	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Imas Sukaesih Sitanggang		✓				✓			✓	✓	✓	✓	✓	✓
Mushthofa	✓			✓	✓			✓		✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The author states that no financial or personal relationships exist that could be perceived as influencing the findings of this work.

DATA AVAILABILITY

The dataset used in this study was obtained from Yahoo Finance. The data were directly downloaded from https://finance.yahoo.com/ and are available upon reasonable request to the corresponding author.

REFERENCES

[1] A. S. Saud and S. Shakya, "Technical indicator empowered intelligent strategies to predict stock trading signals," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 4, Dec. 2024, doi: 10.1016/j.joitmc.2024.100398.

[2] J. Ayala, M. G.-Torres, J. L. V. Noguera, F. G.-Vela, and F. Divina, "Technical analysis strategy optimization using a machine learning approach in stock market indices," *Knowledge-Based Systems*, vol. 225, Aug. 2021, doi: 10.1016/j.knosys.2021.107119.

[3] P. Carvalho, J. Mégane, N. Lourenço, and P. Machado, "Context matters: adaptive mutation for grammars," *Genetic Programming (EuroGP 2023)*, Cham, Switzerland: Springer, Mar. 2023, pp. 117-132, doi: 10.1007/978-3-031-29573-7_8.

[4] J. Megane, N. Lourenco, and P. Machado, “Probabilistic structured grammatical evolution,” in *2022 IEEE Congress on Evolutionary Computation (CEC)*, Padua, Italy, 2022, pp. 1-9, doi: 10.1109/CEC55065.2022.9870397.

[5] I. G. Tsoulos, C. Stylios, and V. Charalampous, “COVID 19 predictive models based on grammatical evolution,” *SN Computer Science*, vol. 4, no. 2, 2023, doi: 10.1007/s42979-022-01632-w.

[6] J. Jeschke, D. Sun, A. Jamshidnejad, and B. D. Schutter, “Grammatical-evolution-based parameterized model predictive control for urban traffic networks,” *Control Engineering Practice*, vol. 132, Mar. 2023, doi: 10.1016/j.conengprac.2022.105431.

[7] T. P. Pawlak and M. O’Neill, “Grammatical evolution for constraint synthesis for mixed-integer linear programming,” *Swarm and Evolutionary Computation*, vol. 64, Jul. 2021, doi: 10.1016/j.swevo.2021.100896.

[8] A. V. B. Rodríguez, B. I. A.-Cerezo, and C. A. C. Coello, “Improving multi-objective evolutionary algorithms using grammatical evolution,” *Swarm and Evolutionary Computation*, vol. 84, Feb. 2024, doi: 10.1016/j.swevo.2023.101434.

[9] J. Mégane, N. Lourenço, and P. Machado, “Co-evolutionary probabilistic structured grammatical evolution,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, New York, NY, USA: ACM, Jul. 2022, pp. 991–999, doi: 10.1145/3512290.3528833.

[10] C. Ryan, M. Kshirsagar, P. Chaudhari, and R. Jachak, “GETS: grammatical evolution based optimization of smoothing parameters in univariate time series forecasting,” in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*, SCITEPRESS-Science and Technology Publications, 2020, pp. 595–602, doi: 10.5220/0008963305950602.

[11] I. K. Nti, A. F. Adekoya, and B. A. Weyori, “A comprehensive evaluation of ensemble learning for stock-market prediction,” *Journal of Big Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00299-5.

[12] T. Aggrawal and M. Dhawan, “State-of-the-art vs prominent models: an empirical analysis of various neural networks on stock market prediction,” *Engineering archive*, 2022, doi: 10.31224/2745.

[13] Y. Li and Y. Pan, “A novel ensemble deep learning model for stock prediction based on stock prices and news,” *International Journal of Data Science and Analytics*, vol. 13, no. 2, pp. 139–149, Mar. 2022, doi: 10.1007/s41060-021-00279-9.

[14] U. F. I. Abdulrahman, N. Ussiph, and B. H. -Acquah, “A hybrid ARIMA-LSTM model for stock price prediction,” *International Journal of Computer Engineering and Information Technology*, vol. 12, no. 8, pp. 48–51, 2020.

[15] A. Subasi, “Data preprocessing,” in *Practical Machine Learning for Data Analysis Using Python*, Elsevier, 2020, pp. 27–89, doi: 10.1016/B978-0-12-821379-7.00002-3.

[16] R. B.-M.-Tonis, R. Bucea, and M. Tonis, “Automating scientific paper screening with backus-naur form (BNF) grammars,” *Didactica Danubiensis*, vol. 4, no. 1, 2024, pp. 46-57.

[17] D. K. Sharma, M. Chatterjee, G. Kaur, and S. Vavilala, “Deep learning applications for disease diagnosis,” in *Deep Learning for Medical Applications with Unique Data*, Elsevier, 2022, pp. 31–51, doi: 10.1016/B978-0-12-824145-5.00005-8.

[18] T.-T.-H. Phan and X. H. Nguyen, “Combining statistical machine learning models with ARIMA for water level forecasting: the case of the Red River,” *Advances in Water Resources*, vol. 142, Aug. 2020, doi: 10.1016/j.advwatres.2020.103656.

[19] E. Dave, A. Leonardo, M. Jeanice, and N. Hanafiah, “Forecasting Indonesia exports using a hybrid model ARIMA-LSTM,” *Procedia Computer Science*, vol. 179, pp. 480–487, 2021, doi: 10.1016/j.procs.2021.01.031.

[20] D. Kobiela, D. Krefta, W. Król, and P. Weichbroth, “ARIMA vs LSTM on NASDAQ stock exchange data,” *Procedia Computer Science*, vol. 207, pp. 3836–3845, 2022, doi: 10.1016/j.procs.2022.09.445.

[21] B. K. Jha and S. Pande, “Time series forecasting model for supermarket sales using FB-prophet,” in *Proceedings-5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, 2021, pp. 547–554, doi: 10.1109/ICCMC51019.2021.9418033.

[22] J. M. Sangeetha and K. J. Alfia, “Financial stock market forecast using evaluated linear regression based machine learning technique,” *Measurement: Sensors*, vol. 31, Feb. 2024, doi: 10.1016/j.measen.2023.100950.

[23] J. Cheng, S. Tiwari, D. Khaled, M. Mahendru, and U. Shahzad, “Forecasting bitcoin prices using artificial intelligence: Combination of ML, SARIMA, and Facebook prophet models,” *Technological Forecasting and Social Change*, vol. 198, 2024, doi: 10.1016/j.techfore.2023.122938.

[24] S. R. Riady, “Stock price prediction using prophet Facebook algorithm for BBCA and TLKM,” *International Journal of Advances in Data and Information Systems*, vol. 4, no. 2, Apr. 2023, doi: 10.25008/ijadis.v4i2.1258.

[25] I. K. Suwintana, I. G. A. O. Sudiadnyani, and N. G. A. P. H. Saptarini, “Developing web-based application of sales forecasting system using triple exponential smoothing method for small and medium garment enterprises,” in *International Conference on Science and Technology (ICST 2018)*, 2018, doi: 10.2991/icst-18.2018.215.

[26] L. Kumar, S. Khedlekar, and U. K. Khedlekar, “A comparative assessment of holt winter exponential smoothing and autoregressive integrated moving average for inventory optimization in supply chains,” *Supply Chain Analytics*, vol. 8, Dec. 2024, doi: 10.1016/j.sca.2024.100084.

[27] L. Lusindah and E. Sumirat, “Implementation of fibonacci retracements and exponential moving average (EMA) trading strategy in Indonesia stock exchange,” *European Journal of Business and Management Research*, vol. 6, no. 4, pp. 402–408, Aug. 2021, doi: 10.24018/ejbmr.2021.6.4.1033.

[28] S. M. Ayene and A. M. Yibre, “Wind power prediction based on deep learning models: the case of Adama wind farm,” *Heliyon*, vol. 10, no. 21, Nov. 2024, doi: 10.1016/j.heliyon.2024.e39579.

[29] T. Iida, “Identifying causes of errors between two wave-related data using performance metrics,” *Applied Ocean Research*, vol. 148, Jul. 2024, doi: 10.1016/j.apor.2024.104024.

[30] A. M. Khan and M. Osińska, “Comparing forecasting accuracy of selected grey and time series models based on energy consumption in Brazil and India,” *Expert Systems with Applications*, vol. 212, Feb. 2023, doi: 10.1016/j.eswa.2022.118840.

[31] Ü. Ağbulut, A. E. Gürel, and Y. Biçen, “Prediction of daily global solar radiation using different machine learning algorithms: evaluation and comparison,” *Renewable and Sustainable Energy Reviews*, vol. 135, Jan. 2021, doi: 10.1016/j.rser.2020.110114.

[32] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Computer Science*, vol. 7, Jul. 2021, doi: 10.7717/peerj-cs.623.

[33] J. Yuan, J. Li, and J. Hao, “A reliable ensemble forecasting modeling approach for complex time series with distributionally robust optimization,” *Computers and Operations Research*, vol. 173, Jan. 2025, doi: 10.1016/j.cor.2024.106831.

[34] L. Li, J. Long, and M. Yuan, “Novel wind speed ensemble forecasting system based on the critic weighing principle of fuzzy information granulation and reverse mixed-frequency modeling,” *Energy*, vol. 330, Sep. 2025, doi: 10.1016/j.energy.2025.136419.

[35] H. Wu and D. Levinson, “The ensemble approach to forecasting: a review and synthesis,” *Transportation Research Part C: Emerging Technologies*, vol. 132, Nov. 2021, doi: 10.1016/j.trc.2021.103357.

[36] H. Ma, M. Li, S. Lv, L. Wang, and S. Deng, “Roulette wheel-based level learning evolutionary algorithm for feature selection of high-dimensional data,” *Applied Soft Computing*, vol. 163, Sep. 2024, doi: 10.1016/j.asoc.2024.111948.

[37] G. D. C. Cavalcanti and R. J. O. Soares, "Ranking-based instance selection for pattern classification," *Expert Systems with Applications*, vol. 150, Jul. 2020, doi: 10.1016/j.eswa.2020.113269.
[38] T. Huang, X. Tang, S. Zhao, Q. Zhang, and W. Pedrycz, "Linguistic information-based granular computing based on a tournament selection operator-guided PSO for supporting multi-attribute group decision-making with distributed linguistic preference relations," *Information Sciences*, vol. 610, pp. 488–507, Sep. 2022, doi: 10.1016/j.ins.2022.07.050.

BIOGRAPHIES OF AUTHORS

Aditya Kusuma Setyanegara is a magister student of Computer Science in IPB University, Indonesia. Specializing in mathematical computing, his research interests primarily revolve around machine learning and artificial intelligence. He can be contacted at email: setyanegara.aditya@apps.ipb.ac.id.

Imas Sukaesih Sitanggang received a Ph.D. degree in Computer Science from the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, in 2013. She is a lecturer in Department of Computer Science at IPB University, Indonesia. Her main research interests include spatial data mining. She can be contacted at email: imas.sitanggang@apps.ipb.ac.id.

Mushthofa received his Ph.D. from Ghent University, Belgium, in 2019. He is currently a lecturer at the School of Data Science, Mathematics and Informatics, IPB University, Indonesia. His research interests focus on artificial intelligence and machine learning, particularly their applications in agriculture and bioinformatics. He has been actively involved in various research projects and publications related to data-driven solutions for sustainable agriculture and biological data analysis. He has published several papers in international journals and conferences, contributing to the advancement of computational techniques in agrotechnology and life sciences. He can be contacted at email: mush@apps.ipb.ac.id.