Annals of Optimization With Applications

www.anowa.reapress.com

Ann. Optim. Appl. Vol. 2, No. 2 (2026) 108–116.

Paper Type: Original Article

Gold Price Forecasting Using Tuned Gated Recurrent Units: Comparing Random Search and Bayesian Optimization

Morteza Moradi*

Department of Industrial Engineering, Urmia University of Technology, Urmia, Iran; moradi@uut.ac.ir.

Citation:

Received: 12 December 2024 *Revised: 14 Februray 2025* *Accepted: 18 April 2025*	Moradi, M. (2026). Gold price forecasting using tuned gated recurrent units: Comparing random search and Bayesian optimization. *Annals of Optimization with Applications, 2*(2), 108-116.

Abstract

Gold price forecasting has long been a significant area of empirical and academic research, given gold's role as a safe-haven asset and a hedge against market instability for investors and central banks worldwide. This paper predicts today's gold closing price based on the previous ten-day Open, High, Low, and Closing (OHLC) prices. The Gated Recurrent Unit (GRU) was utilized for this prediction. However, GRU, like many other deep learning models, has numerous hyperparameters, the tuning of which directly impacts its performance. To address this, two common hyperparameter tuning methods, namely Random Search (RS) and Bayesian Optimization (BO), were employed. To determine the superior tuning method, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R^2, as well as tuning time, were compared using the non-parametric Mann-Whitney U test. Statistical analysis indicates that with 95% confidence, there is no statistically significant difference in any of the evaluated metrics between the two tuning methods. Only with approximately 90% confidence can it be stated that BO tunes the GRU more rapidly. In terms of performance metrics, the best parameter setting was achieved through RS, resulting in MAPE = 1.79% and R^2 = 99.07%. To the best of my knowledge, no comprehensive study to date has compared RS and BO tuning strategies in GRU-based gold price forecasting using multi-day OHLC data and a statistical method, highlighting the novelty of this research.

Keywords: Gold price, Forecasting, Gated recurrent unit, Hyperparameter tuning, Random search, Bayesian optimization.

1 | Introduction

Gold price forecasting has long been a subject of empirical and academic research interest due to gold's role as a safe-haven asset and a hedge against market volatility [1]. Accurate gold price prediction supports informed investment and risk management decisions, especially in volatile financial markets [2]. Forecasting

Corresponding Author: moradi@uut.ac.ir

https://doi.org/10.48314/anowa.v3i2.71

gold price fluctuation is essential for financial investors and central banks to make informed investment choices and effectively manage risk [3], [4].

The application of deep learning models, particularly Gated Recurrent Units (GRUs), has gained prominence for gold price forecasting because of their ability to capture long-term dependencies and reduce computational complexity compared to traditional recurrent architectures.

Sudiatmika et al. [5] utilized a GRU model to predict gold prices, demonstrating the model's effectiveness for this task. The research examined historical gold price data from Yahoo Finance and evaluated the model's performance using Mean Squared Error (MSE) and Mean Absolute Error (MAE). The findings of [6] reveal that GRU surpasses Long Short-Term Memory (LSTM) in gold price prediction. Sentiko and Zakiyyah [7] compared GRU, LSTM, and Linear Regression models for forecasting gold prices in Indonesia. The results indicated that GRU was the most accurate model, outperforming both LSTM and Linear Regression. The GRU model achieved an R^2 of 96%, exceeding LSTM's 93% and Linear Regression's 86%. A study by Yurtsever [8] utilized GRU, LSTM, and Bi-directional LSTM (Bi-LSTM) models to forecast monthly gold prices. The research investigated the influence of several macroeconomic factors, including crude oil prices, exchange rates, stock market indices, and interest rates, on gold prices from 2001 to 2021. In [9], several models were evaluated for monthly gold price prediction. It has been claimed that Simple Linear Regression proved to be the most accurate, achieving a MAPE of 2.22%.

However, both GRU and LSTM models successfully identified nonlinear patterns, which is a key strength for complex, volatile datasets. The GRU model was marginally more effective than the LSTM model, with a MAPE of 2.83% compared to the LSTM's 2.96%, while the LSTM model showed slightly better performance when evaluated by Root Mean Squared Error (RMSE). Putri et al. [4] implemented and compared four neural architectures: 1) GRU, 2) Bidirectional GRU (Bi-GRU), 3) LSTM, and 4) Bi-LSTM to model global gold prices. The study investigates multiple hyperparameter configurations, including optimizers, batch sizes, and time-step settings, across the different algorithms, without delineating a specific hyperparameter tuning methodology

Hyperparameter tuning is a crucial step in optimizing the predictive performance of deep learning models, including GRU [10]. Traditional tuning strategies like Grid Search (GS) systematically explore predefined parameter spaces. However, executing this can be quite costly if the number of hyperparameters and their possible values are substantial. In contrast, Random Search (RS) samples parameters stochastically, often yielding good results with fewer evaluations [11]. Recently, Bayesian Optimization (BO) has emerged as a more sample-efficient approach that leverages probabilistic models to balance exploration and exploitation when searching for optimal hyperparameters [12], [13]. In financial time series forecasting, BO has been used frequently for competitive accuracy and reduced computational cost [14–16].

In many published studies, statistical comparisons between applied methods have not been conducted, and the selection of the best method has been based on examining only one implementation of the methods. Given the random behavior of Machine Learning (ML) algorithms at various stages of their operation, this approach cannot be deemed correct, and statistical comparison methods should be employed,

This study aims to compare the performance of RS and BO in tuning GRU-based models for gold price forecasting using the previous ten-day Open, High, Low, and Closing (OHLC) data to predict today's closing price. This work offers practical insights into the most effective tuning strategies for financial deep learning applications using a statistical method, namely the Mann-Whitney U test. The following sections present the methodology, results, and conclusions.

2 | Methodology

This study aims to assess the effectiveness of two hyperparameter tuning techniques (RS and BO) in predicting gold closing prices using GRU-based models. The overall workflow encompasses data acquisition,

preprocessing, model design, hyperparameter tuning, performance evaluation, and a statistical comparison of performance metrics.

2.1 | Data Acquisition

Daily gold prices were sourced from the Stooq financial data service[1]. Each record includes daily OHLC price information. The raw data utilized in this research encompassed all prices from 1/2/2020 to 6/30/2025. For predictive modeling, all OHLC prices were used for input features; however, only the closing price was treated as the output, as it most accurately represents market consensus at the end of a trading day.

2.2 | Data Preprocessing

To construct a time series for predictive modeling, the raw OHLC data were transformed into a ten-day rolling input window, where the OHLC values from the preceding ten days served as features to predict the next day's closing price. All numerical features were normalized to the [0,1] range using *Eq. (1)* to enhance model convergence [17]. Normalization is particularly important in recurrent networks, as large-scale differences in features can slow convergence and bias gradient updates.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (1)$$

Additionally, the data was split into 80% training and 20% testing sets, with temporal order maintained to prevent leakage of future information.

2.3 | Model Architecture

The forecasting model utilizes GRU architecture because of its efficiency in capturing sequential dependencies in time series data while lowering computational costs compared to standard Recurrent Neural Networks (RNNs) and LSTM networks. As illustrated in *Fig. 1*, GRUs employ a streamlined gating mechanism with only two gates, the update and reset gates, unlike the three gates found in LSTMs. This simplification reduces the number of parameters, allowing for quicker training and inference without significantly sacrificing performance [18].

The network consisted of one or more GRU layers, followed by dropout layers after each recurrent layer to reduce overfitting. Finally, a fully connected dense layer with a linear activation function maps the latent representation to the predicted closing price.

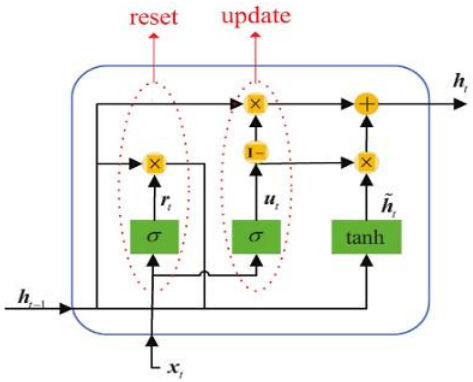

Fig. 1. GRU architecture.

[1] https://stooq.com/q/d/?s=xauusd

2.4 | Hyperparameter Tuning

We systematically compared two widely adopted hyperparameter optimization strategies: RS and BO. Both methods were implemented under controlled conditions to enable a fair and statistically sound comparison.

RS served as a strong baseline due to its simplicity and proven effectiveness in high-dimensional search spaces [11]. Hyperparameter values were sampled independently from predefined uniform (or log-uniform, where appropriate) distributions without adaptive guidance. This stochastic sampling provides good coverage of the search space but does not exploit information from previous evaluations [12].

BO employs a Gaussian Process (GP) surrogate model to approximate the unknown objective function. This surrogate captures uncertainty in unexplored areas of the hyperparameter space, facilitating a more efficient search strategy. The optimization begins with a set of randomly selected configurations, after which the GP is iteratively updated using the results of evaluations. In each subsequent iteration, the Expected Improvement (EI) acquisition function is maximized to identify the most promising candidate configuration. BO thus integrates exploration of uncertain areas with exploitation of regions predicted to deliver high performance, thereby reducing the number of evaluations required to achieve optimal performance [13].

The following hyperparameter space was searched to tune GRU:

I. Number of GRU layers: {1, 2}.

II. Units per GRU layer: {50, 75, 100, 125, 150}

III. Dropout rate: {0.0, 0.05, 0.10, 0.15, 0.20}

IV. Learning rate: Ten points of $\mathcal{LU} \sim (10^{-3}, 10^{-2})$

V. Optimizer: {Adam, SGD, RMSprop}.

VI. Batch size: {32, 64, 128}

This search space was designed to balance computational efficiency and model expressiveness, following best practices for neural architecture hyperparameter optimization.

2.5 | Cross-Validation Strategy

A time split cross-validation approach was used [19] to respect the temporal order of the data. This strategy ensured that at no point did the model train on data from the future relative to the validation set. This process mimics real-world forecasting tasks where future values must be predicted based only on historical data. *Fig. 2* depicts a 3-fold time split cross-validation.

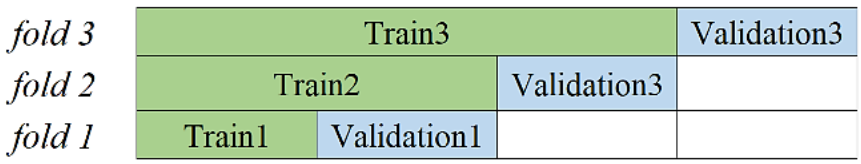

Fig. 2. 3-fold time split cross-validation.

2.6 | Final Training and Evaluation

Following the tuning process, the optimal hyperparameters (for each method) were retrained on the complete training set using the early-stopping strategy. The independent test set was set aside solely for the final evaluation. The performance evaluation metrics included:

I. RMSE.

II. MAE.

III. MAPE.

IV. Coefficient of determination (R^2).

V. Tuning time.

2.7 | Statistical Comparison

To rigorously evaluate whether BO outperforms RS, the Mann–Whitney U test (also called the Wilcoxon Rank-Sum test) is employed across the 11 independent repetitions. The Mann–Whitney U test is a non-parametric statistical method used to determine if two independent samples originate from the same distribution. Unlike the t-test, it does not assume normality, which is crucial in optimization benchmarking where metric distributions may be skewed or multi-modal [20]. Given two independent samples, i.e., $X = \{x_1, x_2, \dots, x_{n_X}\}$ from RS results, and $Y = \{y_1, y_2, \dots, y_{n_Y}\}$ from the BO results, the test statistic is calculated as

$$R_X = \sum_{i=1}^{n_X} \text{rank}(x_i), \quad (2)$$

$$U_X = n_X n_Y + \frac{n_X(n_X + 1)}{2} - R_X, \quad (3)$$

$$U_Y = n_X n_Y - U_X. \quad (4)$$

R_X is the sum of ranks assigned to sample X when both samples are combined and ranked.

In this study, one-sided statistical tests are employed to evaluate whether BO outperforms RS. Specifically, the alternative hypothesis (H1) states that the coefficient of determination (R^2) of BO is greater than that of RS, while the remaining performance metrics for BO are lower than those of RS. In all cases, the null hypothesis (H0) assumes no difference between the two methods, i.e., the metrics of BO and RS are equal. The significance level was set at 5%, meaning that the conclusions can be made with 95% confidence. For each metric, the U statistic and corresponding p-value are reported.

2.8 | Experimental Setting

In an experimental setting to ensure methodological rigor and comparability between the two tuning methods, a specific experimental protocol was implemented.

Each candidate configuration was assessed using a time split cross-validation scheme with a 3-fold using the time series split function, which maintained the temporal ordering of observations and eliminated look-ahead bias.

To account for the inherent randomness in both search methods and their initialization, each optimizer was executed over 11 independent runs. This repetition produced empirical distributions of performance, allowing for robust statistical comparisons. To ensure an equivalent computational budget between the two methods, both RS and BO were limited to 30 evaluations per run. Specifically, RS was restricted to 30 evaluations, while BO initially conducted five random evaluations to initialize its surrogate model, followed by 25 guided iterations using the bayes_opt package.

The MSE was utilized as the loss function, consistent with its common application in regression tasks. Training continued for up to 200 epochs, but an early stopping criterion (patience = 25) was applied to mitigate overfitting by halting training when the validation loss failed to improve.

3 | Results

To predict the closing price of gold for the next day using today's and the previous nine days' OHLC daily prices, the described methodology was implemented in Python/Keras within Jupyter Lab IDE. For this purpose, several libraries were utilized, including Pandas, NumPy, and bayes_opt. Hyperparameter tuning was performed using two search strategies, BO and RS, each executed independently 11 times. Model

performance was evaluated using RMSE, MAE, MAPE, R^2, and tuning time. The results of these runs are presented in *Table 1*.

It seems that BO demonstrated marginally superior average predictive accuracy, with lower RMSE (0.01781 vs. 0.01807), MAE (0.01314 vs. 0.01335), and MAPE (1.93% vs. 1.95%) compared to RS. BO also achieved a slightly higher mean R^2 value (98.89% vs. 98.84%), indicating a marginally better fit to the data. Computational efficiency favored BO, which required an average tuning time of 4,737 seconds, 632 seconds less than RS (5,369 seconds). Consistency metrics further highlighted BO's stability: lower standard deviations for RMSE (0.00151 vs. 0.00272), MAE (0.00092 vs. 0.00189), and MAPE (0.11% vs. 0.22%) underscored reduced variability across runs. However, the observed differences in performance metrics were small in magnitude, prompting further statistical validation, which is often overlooked in most ML research.

In this study, a one-sided Mann-Whitney U test was employed to compare the distributions of performance metrics between BO and RS, and *Table 2* presents the results. For all metrics (Tuning time, RMSE, MAE, MAPE, and R^2), the p-values (ranging from 0.106 to 0.744) exceeded the 0.050 significance threshold. Consequently, the null hypothesis (no difference between methods) could not be rejected for any metric, indicating that the observed differences in *Table 1* lacked statistical significance. However, only at approximately a 90% confidence level, BO was significantly faster at tuning the GRU model's parameters compared to RS.

The combined results reveal a nuanced relationship between performance and statistical validity. While BO exhibited marginally superior average metrics and faster tuning times (*Table 1*), the absence of statistically significant differences (*Table 2*) suggests that these improvements may stem from random variability rather than inherent superiority. This discrepancy highlights the importance of complementing descriptive performance comparisons with inferential statistical analysis to avoid overinterpreting minor numerical advantages.

Additionally, the hyperparameter tuning process yielded valuable insights into the model's optimized architecture. The search concluded with a slightly balanced distribution of network depth, selecting a two-layer stacked GRU in 12 instances and a single-layer architecture in the remaining 10. Notably, the RS method was primarily responsible for identifying the two-layer configurations. Across all 22 tuning trials, the Adam optimizer was exclusively chosen, showcasing its robust and consistent performance. The selection of activation functions demonstrated a strong preference for tanh, which was selected in 14 of the 22 runs, while Scaled Exponential Linear Unit (SELU) was chosen 8 times, and Rectified Linear Unit (ReLU) was never selected. Dropout regularization was most frequently set at a rate of 0.05, occurring in 6 iterations, with two additional instances opting for a value close to 0.05. The number of neurons per layer varied considerably, although 100 units emerged as the most common choice, selected five times. Similarly, batch size values were diverse; however, 32 and 128 were the most frequent selections, chosen seven and four times, respectively.

Table 1. Summary of performance results and tuning times of tuned GRUs using BO and RS.

Method	Run	RMSE	MAE	MAPE (%)	R^2 (%)	Tuning Time(s)
BO	1	0.01703	0.01259	1.87	98.99	4,265
	2	0.01671	0.01259	1.87	99.03	4,757
	3	0.01680	0.01256	1.86	99.02	4,320
	4	0.01968	0.01435	2.08	98.65	5,734
	5	0.01670	0.01246	1.85	99.03	6,177
	6	0.02017	0.01468	2.10	98.58	3,499
	7	0.01657	0.01227	1.82	99.04	5,289
	8	0.01840	0.01364	2.01	98.82	3,484
	9	0.02023	0.01438	2.06	98.58	4,180
	10	0.01659	0.01243	1.85	99.04	4,251
	11	0.01708	0.01263	1.87	98.98	6,145
	Average	0.01781	0.01314	1.93	98.89	4,737
	SD	0.00151	0.00092	0.11	0.19	970

Table 1. Continued.

Method	Run	RMSE	MAE	MAPE (%)	R^2 (%)	Tuning Time(s)
RS	1	0.01649	0.01232	1.83	99.05	5032
	2	0.02081	0.01540	2.21	98.49	5229
	3	0.01717	0.01274	1.88	98.97	5413
	4	0.02519	0.01819	2.51	97.79	7036
	5	0.01636	0.01197	1.79	99.07	7815
	6	0.01756	0.01303	1.93	98.93	4603
	7	0.01663	0.01243	1.85	99.04	6492
	8	0.01643	0.01206	1.79	99.06	4295
	9	0.01884	0.01384	2.02	98.76	4471
	10	0.01649	0.01234	1.84	99.05	4498
	11	0.01680	0.01248	1.85	99.02	4174
	Average	0.01807	0.01335	1.95	98.84	5,369
	SD	0.00272	0.00189	0.22	0.39	1,220

Table 2. Summary of the Mann–Whitney U test.

Item	U Statistic	p-value	Reject H_0
Tuning time	41.0	0.106082	No
RMSE	70	0.744297	No
MAE	70	0.744297	No
MAPE	69	0.722735	No
R^2	51	0.744297	No

4 | Conclusion

This study forecasts today's gold closing price using daily OHLC data from the past 10 days. The dataset covers January 2, 2020, to June 30, 2025, from the Stooq financial data service. A GRU model was used for this task. To tune the model, RS and BO were compared. The effectiveness of these methods was evaluated over 11 repetitions using metrics like RMSE, MAE, MAPE, and R^2, along with the total tuning time. This study examined the effects of hyperparameter tuning on a GRU model, concentrating on the number of layers, units per layer, learning rate, dropout rate, optimizer, and batch size.

While superficial evaluations indicated the superiority of BO over RS, the statistical analysis at a 95% confidence level showed no significant difference between the two methods across all performance metrics. A significant difference was noted at approximately the 90% confidence level, where BO was observed to tune the GRU parameters more swiftly. In terms of performance, the most effective parameter tuning was achieved using RS, which resulted in a MAPE of 1.79% and an R^2 value of 99.07%. Conversely, the least favorable results also came from RS, with a MAPE of 2.51% and an R^2 of 97.79%. Regarding computational efficiency, BO achieved the quickest tuning times, at 3484 seconds, while RS recorded the slowest tuning times, at 7815 seconds.

An analysis of 22 tuned GRU models revealed that a two-layer stacked GRU was chosen in 12 of the cases, with the remaining models using a single-layer architecture. The Adam optimizer was consistently used across all models. For activation functions, tanh was the preferred choice, utilized in 14 models, followed by SELU in 8, while ReLU (which is the primary choice for the activation function in most research) was not selected at all.

Future work could investigate whether larger sample sizes or alternative evaluation frameworks might reveal more definitive distinctions between BO and RS. Also, investigating alternative deep learning architectures such as BiLSTM, BiGRU, 1D-CNN, or Transformers for forecasting gold prices is recommended. A statistical comparison of these models with the GRU-based approach presented in this study would be advantageous. The scope of future studies could also be broadened to assess the effects of using different time series lengths, as this research employed a 10-day time window. Additionally, sentiment analysis could be utilized to analyze news and events, evaluating their impact on gold price fluctuations.

Authors' Contributions

All aspects of the research and manuscript preparation were carried out by the author. The author has read and approved the final version of the manuscript.

Data Availability

All data supporting the reported findings in this research paper are provided within the manuscript.

Funding

Not applicable.

Conflict of Interest

The author declares that they do not have any conflict of interest.

Consent for Publication

The author confirms consent for the publication of this work

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants performed by the author.

References

[1] Reboredo, J. C. (2013). Is gold a safe haven or a hedge for the US dollar? Implications for risk management. *Journal of banking & finance*, *37*(8), 2665–2676. https://doi.org/10.1016/j.jbankfin.2013.03.020

[2] Pangestu, P., Rochman, A., & Zuhdi, A. (2023). The comparison of gold price prediction techniques using long short term memory (LSTM) and fuzzy time series (FTS) method. *Intelmatics*, *3*(2), 57–62. https://doi.org/10.25105/itm.v3i2.17325

[3] Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN-LSTM model for gold price time-series forecasting. *Neural computing and applications*, *32*(23), 17351–17360. https://doi.org/10.1007/s00521-020-04867-x

[4] Putri, A. I., Syarif, Y., Aisyi, N. R., & Waeyusoh, N. (2025). Implementation of gated recurrent unit, long short-term memory and derivatives for gold price prediction. *Public research journal of engineering, data technology and computer science*, *2*(2), 68–80. https://doi.org/10.57152/predatecs.v2i2.1609

[5] Sudiatmika, I. P. G. A., Putra, I. M. A. W., & Artana, W. W. (2024). The implementation of gated recurrent unit (GRU) for gold price prediction using Yahoo finance data: A case study and analysis. *Brilliance: Research of artificial intelligence*, *4*(1), 176–184. https://doi.org/10.47709/brilliance.v4i1.3865

[6] Dewi, N. P. J. R., Ginantra, N. L. W. S. R., Darma, I. W. A. S., & Indrawan, I. G. A. (2023). Application of long short-term memory (LSTM) and gated recurrent unit (GRU) algorithm in gold price prediction. *2023 11th international conference on Cyber and IT service management (CITSM)* (pp. 1-6). IEEE. https://doi.org/10.1109/CITSM60085.2023.10455749

[7] Sentiko, S. L., & Zakiyyah, A. Y. (2024). Gold price prediction using machine learning and deep learning. *2024 6th international conference on cybernetics and intelligent system (ICORIS)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICORIS63540.2024.10903557

[8] Yurtsever, M. (2021). Gold price forecasting using LSTM, Bi-LSTM and GRU. *Avrupa bilim ve teknoloji dergisi*, (31), 341–347. https://doi.org/10.31590/ejosat.959405

[9] Aljohani, H., Alshamrani, S., Aljojo, N., Tashkandi, A., & Alsahfi, T. (2024). Predicting monthly gold prices in indian rupees using ARIMA, LSTM, GRU, and simple linear regression models. *Romanian journal of information technology and automatic control*, *34*(2), 35-48. https://doi.org/10.33436/v34i2y202403

[10] Gkonis, V., & Tsakalos, I. (2025). A hybrid optimized deep learning model via the Golden Jackal optimizer for accurate stock price forecasting. *Expert systems with applications*, *278*, 127287. https://doi.org/10.1016/j.eswa.2025.127287

[11] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2), 282–305. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

[12] Shekhar, S., Bansode, A., & Salim, A. (2021). A comparative study of hyper-parameter optimization tools. *2021 IEEE Asia-Pacific conference on computer science and data engineering (CSDE)* (pp. 1-6). IEEE. https://doi.org/10.1109/CSDE53843.2021.9718485

[13] Victoria, A. H., & Maragatham, G. (2021). Automatic tuning of hyperparameters using Bayesian optimization. *Evolving systems*, *12*(1), 217–223. https://doi.org/10.1007/s12530-020-09345-2

[14] Zulfiqar, M., Gamage, K. A. A., Kamran, M., & Rasheed, M. B. (2022). Hyperparameter optimization of Bayesian neural network using Bayesian optimization and intelligent feature engineering for load forecasting. *Sensors*, *22*(12), 4446. https://doi.org/10.3390/s22124446

[15] Dezhkam, A., Manzuri, M. T., Aghapour, A., Karimi, A., Rabiee, A., & Shalmani, S. M. (2023). A Bayesian-based classification framework for financial time series trend prediction. *The journal of supercomputing*, *79*(4), 4622–4659. https://doi.org/10.1007/s11227-022-04834-4

[16] Olaniyan, J., Olaniyan, D., Obagbuwa, I. C., Esiefarienrhe, B. M., Adebiyi, A. A., & Bernard, O. P. (2024). Intelligent financial forecasting with granger causality and correlation analysis using Bayesian optimization and long short-term memory. *Electronics*, *13*(22), 4408. https://doi.org/10.3390/electronics13224408

[17] Switrayana, I. N., Hammad, R., Irfan, P., Sujaka, T. T., & Nasri, M. H. (2025). Comparative analysis of stock price prediction using deep learning with data scaling method. *Jurnal teknologi informasi dan multimedia*, *7*(1), 78–90. https://doi.org/10.35746/jtim.v7i1.650

[18] Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, *15*(9), 517. https://doi.org/10.3390/info15090517

[19] Hoque, K. E., & Aljamaan, H. (2021). Impact of hyperparameter tuning on machine learning models in stock price forecasting. *IEEE access*, *9*, 163815–163830. https://doi.org/10.1109/ACCESS.2021.3134138

[20] Sheskin, D. J. (2003). *Handbook of parametric and nonparametric statistical procedures*. Chapman and Hall/CRC. https://doi.org/10.1201/9781420036268