International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I6P113

Volume 72 Issue 6, 128-135, June 2024

Original Article

Forecasting of Philippine Stock Exchange Index Using Optimized Artificial Neural Networks with Enhanced PSO Algorithm

Dennis M. Barrios II[1], Bobby D. Gerardo[2]

[1]*Graduate Programs, Technological Institute of the Philippines, Manila, Philippines.*
[2]*College of Information and Computing Studies, Northern Iloilo State University, Estancia, Iloilo, Philippines.*

[1]*Corresponding Author : denbarrios07@gmail.com*

Received: 25 March 2024 Revised: 17 May 2024 Accepted: 24 May 2024 Published: 29 June 2024

Abstract *- This research paper explores the application of the Enhanced Particle Swarm Optimization (PSO) algorithm, called the Random Adaptive Backtracking Particle Swarm Optimization (RAB-PSO) algorithm, to optimize Artificial Neural Networks (ANNs) for forecasting the Philippine Stock Exchange Index (PSEi). The study utilizes a dataset spanning from May 11, 2018, to May 10, 2023, sourced from Yahoo Finance and standardized for analysis. The hyperparameter of an ANN model is fine-tuned using the RAB-PSO algorithm to enhance forecasting accuracy. Evaluation metrics such as Root Mean Square Error (RMSE) and R^2 are employed to assess the performance of the optimized ANN model. The results indicate that the ANN model optimized with RAB-PSO has minimal error rates, significantly outperforming the standard PSO algorithm. Generally, this research contributes to the field of PSEi forecasting and emphasizes the significance of optimizing hyperparameters through enhanced PSO for ANNs in financial prediction tasks.*

Keywords *- Artificial Neural Network, Forecasting, Machine learning, Metaheuristic optimization algorithms, Particle swarm optimization, Hyperparameter optimization.*

1. Introduction

The important aspect of investment decision-making is forecasting stock market values, but it is a complex and challenging task because of the numerous influencing factors and the inherent uncertainty in financial markets (Parker, 2014). These factors include investor sentiment, geopolitical events, company performance, economic indicators, and more, which can interact in unpredictable ways, making accurate forecasting challenging (Santi, 2023). Accurate forecasting is important for investors as it can help optimize investment portfolios and potentially maximize returns. Throughout the years, analysts have identified that price patterns can predict future market movements, including trends, reversals, and chart patterns. They can also forecast the magnitude of price changes and analyze regular price cycles to anticipate future movements. Technical analysis has recently grown in popularity as an investment decision-making tool due to its simplicity and profitability. However, very few people understand why forecasting techniques are effective (Zielonka, 2004).

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that concentrates on developing algorithms and statistical models capable of learning from data to make predictions or decisions without needing to be explicitly programmed for each specific task (Huang et al., 2012). ML techniques are applied across various domains, including disease detection (Lagunzad et al., 2022), document categorization (Abalorio et al., 2022), (Biol et al., 2023), image classification (Villaruz et al., 2022), (Gumiran et al., 2022) and time series forecasting (Joy et al., 2022). Artificial Neural Networks (ANNs) are excellent in solving difficult problems since they can learn patterns from data among the ML models. ANNs have shown outstanding accuracy in tasks like stock price forecasting (Villaruz et al., 2023), where they can learn from historical stock data to predict future price movements, leveraging their ability to capture nonlinear relationships and adjust to changing market conditions (Rouf et al., 2021).

The effectiveness of ANN in addressing particular problems is determined by the structure and the values of the hyperparameters (Göçken et al., 2016). Unfortunately, there is no standardized method for determining the ideal hyperparameters, so even experienced practitioners must rely on trial and error. Several researchers opt for default parameters, which may not be ideal for the datasets (Lujan-

Moreno et al., 2018). To solve this, researchers use optimization techniques.

Modern optimization methods for tuning model parameters (hyperparameters) now utilize metaheuristic algorithms. These algorithms make use of randomness to search efficiently through the space of possible solutions, aiming to find near-optimal hyperparameter values for building effective models with datasets. Unlike some optimization techniques, The Particle Swarm Optimization algorithm (PSO) (Kennedy & Eberhart, 1995) gained considerable interest in the literature of heuristic and swarm-based intelligent optimization techniques and performs well in various application domains due to its simplicity, low number of parameters, its ability to generate effective results, and ease of application, (Gbadega & Sun, 2022). Despite its strengths, PSO is susceptible to premature convergence, which can compromise both forecasting accuracy and reliability. To address this problem, a better version of the standard PSO was introduced, the RAB-PSO algorithm (Barrios & Gerardo, 2023). RAB-PSO aimed at reducing iteration counts for achieving global optima and enhancing convergence quality.

This research focuses on forecasting stock prices in emerging countries like the Philippines in terms of the Philippine Stock Exchange Composite Index (PSEi). The novelty of this study is that it explores a technique called hyperparameter tuning to make better forecasting about the PSEi. The study has three primary contributions: (1) by implementing the improved PSO called the RAB-PSO algorithm to improve the accuracy of predicting the PSEi; (2) by adding more literature to the limited research available on forecasting the PSEi; and (3) by showing why optimizing artificial neural networks' parameters through metaheuristic techniques is important.

2. Background and Related Works

2.1. Forecasting the Philippine Stock Exchange Index

The PSEi tracks the performance of the 30 largest publicly traded companies in the Philippines and is used to monitor the stock market. Financial experts and traders rely on composite indexes to forecast market trends and assess the performance of their investments. The composite index is a summary of different stocks, bonds, or indicators, giving an overview assessment of the market's overall performance. Despite the advancements in understanding financial stability (Dakila, 2020) and the involvement of market participants (Dumlao-Abadilla, 2021), there has been no sufficient research done on PSEi forecasting.

Prior research on PSEi forecasting has used a variety of strategies, including the ARIMA model (Gayo et al., 2015), social media analysis (Caliñgo et al., 2016), and machine learning techniques such as the Long Short-Term Memory (LSTM) network (Chua et al., 2021). Furthermore, factors influencing PSEi price movement have been investigated. However, more research needs to be done to make PSEi (Philippine Stock Exchange Index) forecasting models more accurate and dependable. Specifically, the importance of exploring hyperparameter tuning, which is a technique used to optimize the performance of these models.

2.2. Metaheuristic Optimization Algorithms

Metaheuristic optimization algorithms are often used to make machine learning models work better. Metaheuristics are applied to ANNs to enhance their performance and optimize their hyperparameters. These algorithms, such as Firefly Algorithm (FA), Ant Colony Optimization (ACO), Bayesian Optimization, Genetic Algorithms (GA), and Particle Swarm Optimization (PSO), are applied for hyperparameter optimization, weight initialization, feature selection, and architecture search. They efficiently explore the vast search space of hyperparameters, enabling ANNs to achieve optimal configurations for improved performance. Additionally, metaheuristics assist in training optimization by fine-tuning weights and biases, as well as in ensemble learning by selecting and combining multiple models. Furthermore, they facilitate transfer learning and fine-tuning of pre-trained models, allowing ANNs to adapt quickly to new tasks and datasets. The application of metaheuristic optimization algorithms to ANNs empowers them to be more adaptable, accurate, and effective across various domains and tasks.

2.3. Artificial Neural Networks

The concept of ANNs originated in 1943 with McCulloch and Pitts (McCulloch & Pitts, 1943), gaining popularity in 1986 when Rumelhart et al. introduced the backpropagation algorithm (Rumelhart et al., 1986). ANNs, a prominent technique in Artificial Intelligence (AI), emulate human brain functions with computational nodes and transfer functions. They consist of input, hidden, and output layers, each composed of interconnected neurons. These neurons process data through transfer functions, forming a layered structure where outputs from one layer serve as inputs to the next (Medina et al., 2003). The input data, along with bias terms, are multiplied by initial weights, transformed through functions, and propagated through the layers to produce the final output (Dixit & Londhe, 2016). ANN has been commonly used in time series forecasting, such as in traffic control (Olayode et al., 2021), medical systems (Sadek et al., 2019), economic data (Ashin & V, 2020), image processing (Saquib et al., 2020), classification tasks (Villaruz et al., 2022) and stock forecasting (Rouf et al., 2021).

2.4. Hyperparameter Tuning Techniques for ANN

Hyperparameter tuning in ANNs often relies on a trial-and-error method, which not only consumes a lot of computational resources but also frequently falls short of finding an almost perfect solution (Sarkar et al., 2022). The various hyperparameters of ANNs encompass parameters like the learning rate, the number of epochs, the training batch, and the number of neurons in the hidden layer. However, identifying the best values for each of these hyperparameters remains an engaging and difficult research domain (Bischl et al., 2023).

Choosing the right number of neurons in an ANN's hidden layer is crucial (Adil et al., 2022). Too few can make it miss important details that will underfit the model, while too many can make it overly focused on the training data and bad at handling new information that might cause overfitting.

An "epoch" in neural network training is a full run-through of the training data. During each epoch, the model's parameters adjust based on observed data. Each epoch is split into smaller batches because processing all data at once can be impractical. One epoch may lead to "underfitting," so multiple epochs for better generalization I needed. When epochs are increased, the model fits better until it reaches an optimal level. However, going beyond this point may cause "overfitting" (Afaq & Rao, 2020).

The "training batch", or the alternative term for batch size, specifies how many data samples are processed in each training iteration. The dataset is usually split into smaller batches, with parameters updated after processing each batch. The optimal training batch varies based on dataset size, hardware, and the problem at hand. Researchers often experiment with different batch sizes to balance computational efficiency and model performance because of their substantial impact on both prediction accuracy and training time (Smith et al., 2017).

The learning rate is a crucial scalar value that dictates the size of steps the neural network takes along the loss gradient during training (Goodfellow et al., 2016). Discovering the ideal learning rate poses a challenge, as opting for a value that is too low will make the model's learning process very slow. Conversely, if the value is too high, the learning process will fail to reach an optimal solution. Achieving a balance learning rate is important for effective training and attaining optimal model performance.

3. Materials and Methods

The experimental simulation configuration employs a personal computer equipped with a processor of an Intel i7 and boasts a substantial 40.0 GB of RAM. This computer system operates on a 64-bit operating system with an operating system of Windows 11 Home Edition. The simulation framework was created using Python, which incorporates dedicated libraries such as Numpy 1.24.2, Keras 2.11.0, and Matplotlib 3.7.0. Furthermore, the setup and customization of numerous parameters are handled within the Keras framework.

3.1. Data Collection

This study utilized a dataset comprised of daily PSEi's closing price, spanning from May 11, 2018, to May 10, 2023. This dataset was meticulously collected from the Yahoo Finance website (PSEi INDEX (PSEI.PS) Historical Data - Yahoo Finance, 2023) to ensure its credibility and availability. To ensure the dataset's integrity and uniformity, any missing or null values associated with days lacking trading activity were deleted from the dataset. A total of 66% of the dataset is allocated to the training set, while the remaining 34% for the testing set. Notably, the PSEi dataset manifests a noisy, non-linear, and dynamic nature. In preparation for the modeling phase, the dataset was subjected to standardization via min-max normalization, thereby scaling the values within a range of 0 to 1.

3.2. The Enhanced Particle Swarm Optimization Algorithm

The standard Particle Swarm Optimization (PSO) algorithm gained considerable interest in the literature of heuristic and swarm-based intelligent optimization techniques and performs well in various application domains due to its simplicity, low number of parameters, ease of application, and ability to produce effective results (Gbadega & Sun, 2022). PSO operates by leveraging interactions among particles to find the best solution within a given search space. Although efficient, the standard PSO algorithm tends to exhibit slow convergence rates (He et al., 2022) and premature convergence issues, particularly when confronted with intricate, multi-dimensional, multimodal optimization problems (Zhang & Lin, 2022). These limitations can be traced back to the limited number of parameters, which represents a significant barrier to enhancing the algorithm (Sahin & Akay, 2016). To solve this problem, the RAB-PSO algorithm was introduced (Barrios & Gerardo, 2023) as an enhanced version of the standard PSO.

In the study of Barrios and Gerardo in 2023, RAB-PSO encompasses three interrelated modifications aimed at reducing iteration counts for achieving global optima and enhancing convergence quality. The first modification relies on the dynamics of golf ball motion and incorporates the delta parameter to gradually decrease particle velocity. In this approach, the initial iteration ensures maximum velocity, progressively slowing down in subsequent iterations. This strategy aims to improve convergence accuracy while reducing the total number of iterations. The second modification advocates for the utilization of an inertia weight-adjustment method based on an S-shaped pattern to achieve balance in the search process. This method is applied by allowing particle velocities to be high during the early solution phases. Subsequently, in the middle phase, the velocity diminishes to expedite convergence toward the global optimum. Finally, in the later phase, the velocity is stabilized at a consistent level to facilitate the final convergence. The third modification implements an innovative update approach, allowing particles to reverse course if they become trapped in their prior locations. Using Eq. 1, particles may backtrack to a previously successful position, which is personal best (*pBest*), and takes a randomized step during this backtracking process to explore new possibilities.

$$x_{ij}^{t+1} = pBest_{i,j}^{t} + rand(x_{imax} - x_{imin})\frac{T-t}{T} \qquad (1)$$

The rand represents a random number between -1 and 1, adding randomness to the particle's movement direction. The T signifies the maximum iteration, t denotes the current iteration, and x_{imin} and x_{imax} represents the minimum and maximum values for the search space dimensions, respectively. This strategy enhances the algorithm's ability to avoid convergence to local optima and prevents premature convergence.

Algorithm 1 outlines the standard PSO algorithm, with the incorporation of the novel RAB technique depicted in specific lines marked in red, namely lines 8, 9, 13, 14, 15, 18, 19, and 20.

Algorithm 1. RAB-PSO Pseudocode

1 **Initialize Xi, Vi, iteration, maxIteration, pBest, gBest, fitness_criterion**
2 t = 1 // iteration counter
3 T = **maxIteration**
4 Initialize random swarm *P(t)*
5 Evaluate *P(t)*
6 **while** t <= T **do**
7 **for** each particle *i* **do**
8 Calculate Delta, $Delta = \frac{-iterations}{e^{maxiteration}}$
9 **if** *Delta* < *rand* **then** calculate new value of ω using:
$\omega = \omega_{min} + (\omega_{max} - \omega_{min})/(1 + exp(2 * a * t/T - a)) + \omega_{min}$
10 Update $pBest_{i,j}^{t}$ of each particle *i* and find $gBest$
11 Update the particle's velocity v_i^{t+1} using Eq. 1
12 Update the particle's position x_i^{t+1} using Eq. 2
13 **if** current position $\boldsymbol{x_i^t} == \boldsymbol{x_i^{t-1}}$ (previous position) **then**
14 Backtrack particle *I* using: $x_{ij}^{t+1} = pBest_{i,j}^{t} + rand(x_{imax} - x_{imin})\frac{T-t}{T}$
15 **end if**
16 Evaluate x_i^{t+1} and include it in *P*(*t*+1)
17 **end for**
18 **if** each particle's fitness value < fitness_criterion **then**
19 break
20 **end if**
21 **end while**

3.3. Optimizing ANN Based on RAB-PSO Algorithm

This study leverages the effectiveness of the Feed-Forward Neural Network (FFNN) architecture, a well-established ANN configuration, for constructing the forecasting model. FFNNs are renowned for their ability to facilitate the unidirectional flow of information, devoid of loops or feedback connections, and have demonstrated success across a wide domain of applications (Cuk et al., 2021).

The training and testing of the ANN prediction model commence with the random initialization of hyperparameters, and the model's initial performance is assessed based on these settings. The hyperparameter tuning method starts by generating a set of candidate solutions equivalent to the population size of the algorithm, often referred to as a "swarm." In the initialization phase, this initial population is randomly dispersed throughout the search space, with each individual representing an ANN model characterized by a unique combination of hyperparameters. In each iteration, the population undergoes updates involving adjustments to the velocity and position of each particle. Both pBest and gBest influence these updates. Consequently, particles adjust their positions in pursuit of convergence to these new positions, with each particle's position signifying a potential solution, its quality measured by its proximity to the best solution identified thus far in the optimization process.

3.4. Evaluation Metrics

3.4.1. Evaluation of Hyperparameter Tuning

The performance of RAB-PSO in the hyperparameters tuning of ANN is measured in terms of the accuracy of the forecasting model based on RMSE and R^2 comparing the specific iteration *i* of the forecasted price $\hat{p}_i$ and in the same iteration *i* of the actual price p_i, where $\bar{p}$ represents the average of the actual prices, and the forecasting period spans *N* iterations.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - p_i)^2} \qquad (2)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(p_i - \hat{p}_i)^2}{\sum_{i=1}^{N}(p_i - \bar{p})^2} \qquad (3)$$

The RMSE is a commonly used metric in forecasting models, particularly in regression analysis and time series forecasting. It serves as a measure of the accuracy of forecasts. It assesses how closely the model's predictions align with the actual outcomes. A lower RMSE value indicates that the forecasting model is better at making accurate predictions, as it signifies that the errors between predicted and actual values are smaller on average. On the other hand, a lower forecasting accuracy is indicated by a higher RMSE, suggesting that the model's predictions deviate further from the actual values.

The R^2 values usually fall within the range of 0 to 1, indicating how much of the variability in the dependent variable is accounted for by the independent variables included in the model. A higher R^2 signifies better predictive performance, while a lower R^2 suggests reduced explanatory power. Both metrics are essential for assessing forecasting model accuracy and explanatory capability.

Table 1. Parameters of the PSO used in the study

Parameter Name	Value
Population	30
Maximum Generations	100
Number of Independent Runs	100
c1	1
c2	2
ω_{min}	0.20
ω_{max}	0.90
Fitness criterion	10^-5

A one-sample t-test assesses the statistical significance of the improvements made by the RAB-PSO algorithm, with significance indicated if the derived p-value from the t-test is below the 0.05 threshold. (Tao et al., 2022).

4. Experimental Findings and Results

4.1. Convergence Performance

The convergence performance of the RAB-PSO is compared to the standard PSO and is evaluated by comparing its average convergence out of the 100 simulation runs in five well-established benchmark functions that exhibit unimodal and multimodal characteristics. Table 1 presents the parameters used by both algorithms.

4.2. Performance on Hyperparameter Tuning

The hyperparameter settings of the ANN model were fine-tuned using the RAB-PSO algorithm. Table 2 outlines the ranges of hyperparameters that govern the neural network used to forecast the PSEi closing price. This study adopts a single hidden layer architecture for the ANN—a configuration well-established for its efficacy across various practical problems (Nguyen & Bui, 2019).

The Rectified Linear Unit (ReLU) activation function is selected within this layer due to its operation within the interval of 0 to infinity. Furthermore, the optimization process involves employing the Adaptive Moment Estimation (Adam) algorithm, distinguished by its stochastic gradient approximation approach, wherein gradients are calculated based on randomly selected data subsets, as opposed to using the entire dataset—a strategy designed to enhance efficiency and computational speed.

The control parameters that will be used for all PSO implementations to fine-tune the ANN hyperparameter are provided in Table 3.

Table 2. ANN hyperparameters and the search ranges

Hyperparameters	Search Ranges
Hidden Layer Size	(2, 256)
Epochs	(20, 250)
Training Steps	(1, 10)
Training Batch	[None, 2, 4, 8, 16, 32, 64, 128, 256]
Learning Rate	[0.1, 0.01, 0.001]

Table 3. PSO control parameters to Fine-Tune the ANN hyperparameter

Parameter Name	Value
Population	10
Maximum Generations	30
Number of Independent Runs	10
c1	2
c2	2
ω_{min}	0.5
ω_{max}	1.0

Table 4. Comparative analysis of accuracy results using Standard-PSO and RAB-PSO in Fine-Tuning the ANN

Trial	Standard-PSO		RAB-PSO	
	RMSE	R^2	RMSE	R^2
1	79.46402	0.95904	79.51172	0.95900
2	79.33568	0.95918	79.16331	0.95935
3	79.78375	0.95871	79.08642	0.95943
4	79.24349	0.95927	79.39230	0.95912
5	79.29092	0.95922	79.30783	0.95920
6	80.03456	0.95845	79.79310	0.95870
7	80.03877	0.95845	79.60268	0.95890
8	79.24552	0.95927	79.42765	0.95908
9	79.85121	0.95864	79.41324	0.95910
10	79.77638	0.95872	79.32638	0.95919
Average	79.60643	0.95890	**79.40246**	**0.95911**

Table 5. Statistical significance test results for stock price prediction

Error Metric	p-value	Interpretation
RMSE	0.00005	Significant
R^2	0.00005	Significant

The result shows compelling proof of the enhanced effectiveness attained by using the ANN fine-tuned with the RAB-PSO algorithm compared to the standard PSO algorithm. This distinction is noticeable when evaluating the metrics RMSE and R^2. Optimization with the RAB-PSO algorithm consistently yields lower error values for RMSE and higher R^2 values, which signify a more favourable fit of the regression model to the data. Table 4 demonstrates the algorithm's effectiveness in enhancing predictive accuracy and model quality.

The visual representation in Figure 1 complements the statistical findings presented and emphasizes the superiority of fine-tuning the ANN hyperparameters using the RAB-PSO algorithm in terms of predictive accuracy and error reduction.

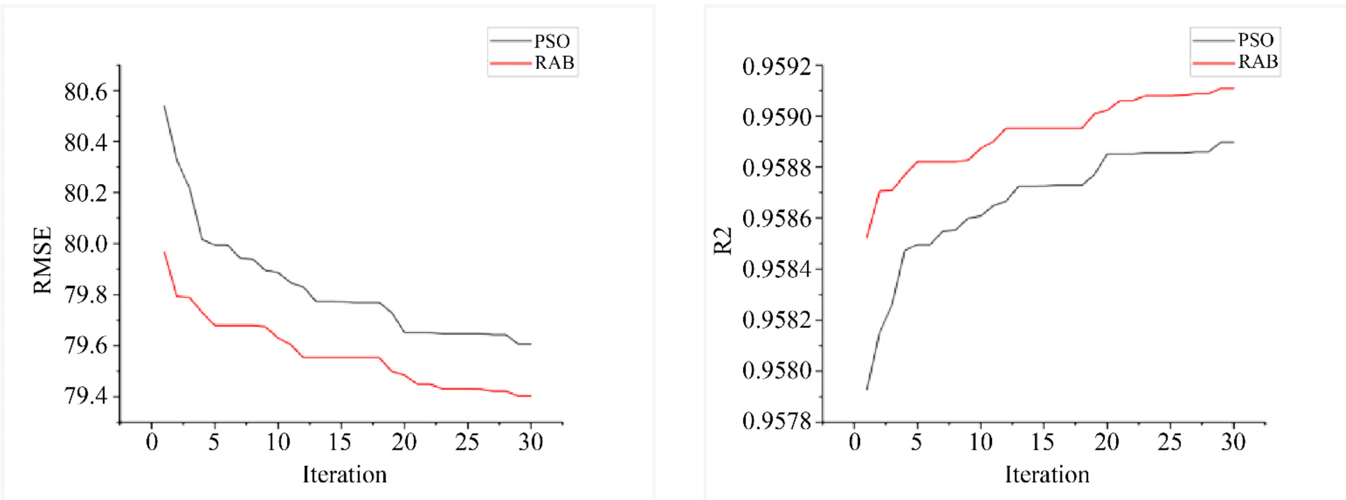
To validate the dominance of the RAB-PSO algorithm over the standard PSO algorithm, the outcomes of the t-test, along with their corresponding interpretations, are summarized in Table 5.

Upon comparing the p-values of RMSE and R^2 to the 0.05 significance level, it becomes evident that the discrepancy between the mean ranks is sufficiently substantial to establish statistical significance. This implies that there are indeed statistically significant improvements in forecasting accuracy and error reduction when employing the RAB-PSO algorithm in contrast to the standard PSO algorithm.

Table 6 presents the ideal hyperparameter values for the ANN model employed in predicting the PSEi closing price.

Table 6. PSEi Dataset Used for Fine-Tuned ANN hyperparameters

Hyperparameter	Optimal Hyperparameters Values
Hidden Layer Size	124
Epochs	191
Training Steps	9
Training Batch	64
Learning Rate	0.001



(a) RMSE (b) R^2

Fig. 1 Convergence plots of the evaluation metrics in PSEi Forecasting

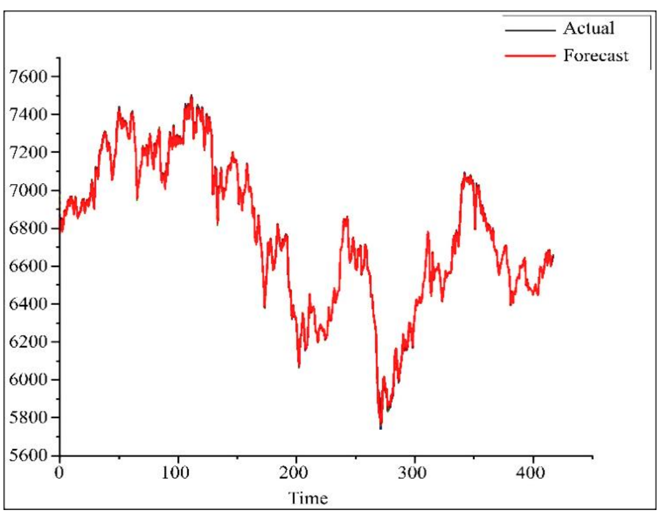


Fig. 2 Actual Vs Forecasted PSEi dataset

Figure 2 displays the forecasted PSEi closing price using the top-performing RAB-PSO algorithm model alongside the actual observed values. The graph underscores the alignment between forecasted and actual values, serving as compelling evidence of the efficacy of the RAB-PSO methodology in delivering reasonably accurate predictions of the PSEi.

5. Conclusion and Future Works

In conclusion, the RAB-PSO algorithm significantly outperformed the standard PSO algorithm in predicting stock prices. For future works, it is recommended to explore the applicability of the RAB-PSO algorithm in other financial forecasting domains like currency exchange rates, commodity prices, or cryptocurrency markets. Future research should also include a quantitative comparison of the RAB-PSO algorithm against other optimization techniques to evaluate both computational efficiency and forecasting accuracy.

References

[1] Cristopher C. Abalorio et al., "Extended Max-Occurrence with Normalized Non-Occurrence as MONO Term Weighting Modification to Improve Text Classification," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 91-97, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Mohammad Adil et al., "Effect of Number of Neurons and Layers in an Artificial Neural Network for Generalized Concrete Mix Design," *Neural Computing and Applications*, vol. 34, no. 11, pp. 8355-8363, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Saahil Afaq, and Smitha Rao, "Significance of Epochs on Training a Neural Network," *International Journal of Scientific & Technology Research*, vol. 9, no. 6, pp. 485-488, 2020. [Google Scholar] [Publisher Link]

[4] M.K. Ashin Nishan, and V. Muhammed Ashiq, "Role of Energy Use in the Prediction of CO2 Emissions and Economic Growth in India: Evidence from Artificial Neural Networks (ANN)," *Environmental Science and Pollution Research*, vol. 27, pp. 23631-23642, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Xinming Zhang, and Qiuying Lin, "Three-Learning Strategy Particle Swarm Algorithm for Global Optimization Problems," *Information Sciences*, vol. 593, pp. 289-313, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Ivic Jan A. Biol et al., "Automated Categorization of Research Papers with MONO Supervised Term Weighting in RECApp," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, pp. 332-339, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Bernd Bischl et al., "Hyperparameter Optimization: Foundations, Algorithms, Best Practices, and Open Challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 2, pp. 1-43, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Anthony R. Calingo, Ariel M. Sison, and Bartolome T. Tanguilig, "Prediction Model of the Stock Market Index Using Twitter Sentiment Analysis," *International Journal of Information Technology and Computer Science*, vol. 8, no. 10, pp. 11-21, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Alvin Ken Steven C. Chua et al., "AI to Predict Price Movements in the Stock Market," *IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management*, Manila, Philippines, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Aleksa Cuk et al., *Feedforward Multi-Layer Perceptron Training by Hybridized Method between Genetic Algorithm and Artificial Bee Colony*, *Data Science and Data Analytics*, 1st ed., Chapman and Hall/CRC, pp.1-14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Francisco G. Dakila Jr, "The Development of Financial Markets in the Philippines and Its Interaction with Monetary Policy and Financial Stability," *BIS Papers Chapters*, vol. 113, pp. 219-242, 2020. [Google Scholar] [Publisher Link]

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, pp. 1-800, 2016. [Google Scholar] [Publisher Link]

[13] Pradnya Dixit, and Shreenivas Londhe, "Prediction of Extreme Wave Heights Using Neuro Wavelet Technique," *Applied Ocean Research*, vol. 58, pp. 241–252, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[14] Doris Dumlao-Abadilla, Young, Lower-Income Filipinos Flock to Stocks Searching for Better Profits, Inquirer.Net, 2021. [Online]. Available: https://business.inquirer.net/324022/young-lower-income-filipinos-flock-to-stocks-searching-for-better-profits

[15] W.S. Gayo et al., "Modeling Philippine Stock Exchange Composite Index Using Time Series Analysis," *Journal of Physics: Conference Series*, vol. 622, no. 1, pp. 1-11, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[16] Peter Anuoluwapo Gbadega, and Yanxia Sun, "A Hybrid Constrained Particle Swarm Optimization-Model Predictive Control (CPSO-MPC) Algorithm for Storage Energy Management Optimization Problem in Micro-Grid," *Energy Reports*, vol. 8, pp. 692-708, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mustafa Göçken et al., "Integrating Metaheuristics and Artificial Neural Networks for Improved Stock Price Prediction," *Expert Systems with Applications*, vol. 44, pp. 320-331, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Samuel L. Smith et al., "Don't Decay the Learning Rate, Increase the Batch Size," *arXiv*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Cherry R. Gumiran et al., "Aedes Aegypti Egg Morphological Property and Attribute Determination Based on Computer Vision," *7th International Conference on Signal and Image Processing*, Suzhou, China, pp. 581-585, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Yiming He et al., "Semi-Airborne Electromagnetic 2.5D Inversion Based on A PSO–LCI Strategy," *Journal of Applied Geophysics*, vol. 197, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Guang-Bin Huang et al., "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513-529, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[22] Meiyao Tao et al., "Knowledge Graph and Deep Learning Combined with a Stock Price Prediction Network Focusing on Related Stocks and Mutation Points," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4322-4334, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, vol. 4, pp. 1942-1948, 1995. [CrossRef] [Google Scholar] [Publisher Link]

[24] Herminiño C. Lagunzad et al., "Predicting the Early Sign of Diabetes Using ID3 as a Data Model," *14th International Conference on Computer and Automation Engineering*, Brisbane, Australia, pp. 135-139, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Gustavo A. Lujan-Moreno et al., "Design of Experiments and Response Surface Methodology to Tune Machine Learning Hyperparameters, with a Random Forest Case-Study," *Expert Systems with Applications*, vol. 109, pp. 195-205, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[26] Warren S. McCulloch, and Walter Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *The Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943. [CrossRef] [Google Scholar] [Publisher Link]

[27] Josep R. Medina et al., "Armor Damage Analysis Using Neural Networks," *Coastal Structures 2003 - Proceedings of the Conference*, pp. 236-248, 2003. [CrossRef] [Publisher Link]

[28] Hoang Nguyen, and Xuan-Nam Bui, "Predicting Blast-Induced Air Overpressure: A Robust Artificial Intelligence System Based on Artificial Neural Networks and Random Forest," *Natural Resources Research*, vol. 28, no. 3, pp. 893-907, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[29] Isaac Oyeyemi Olayode et al., "Comparative Traffic Flow Prediction of a Heuristic ANN Model and a Hybrid ANN-PSO Model in the Traffic Flow Modelling of Vehicles at a Four-Way Signalized Road Intersection," *Sustainability*, vol. 13, no. 19, pp. 1-28, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] David Parker, "Property Investment Decision Making by Australian REITs," *Journal of Property Investment and Finance*, vol. 32, no. 5, pp. 456-473, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[31] PSEi INDEX (PSEI.PS), Historical Data - Yahoo Finance, 2023. [Online]. Available: https://finance.yahoo.com/quote/PSEI.PS/history

[32] Nusrat Rouf et al., "Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions," *Electronics*, vol. 10, no. 21, pp. 1-25, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[33] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning Representations by Back-Propagating Errors," *Nature*, vol. 323, pp. 533-536, 1986. [CrossRef] [Google Scholar] [Publisher Link]

[34] Ramzi M. Sadek et al., "Parkinson's Disease Prediction Using Artificial Neural Network," *International Journal of Academic Health and Medical Research*, vol. 3, no. 1, pp. 1-8, 2019. [Google Scholar] [Publisher Link]

[35] Omur Sahin, and Bahriye Akay, "Comparisons of Metaheuristic Algorithms and Fitness Functions on Software Test Data Generation," *Applied Soft Computing*, vol. 49, pp. 1202-1214, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[36] Caterina Santi, "Investor Climate Sentiment and Financial Markets," *International Review of Financial Analysis*, vol. 86, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[37] Dania Saquib, Mohammed Nabeel Nasser, and Swaroop Ramaswamy, "Image Processing Based Dust Detection and Prediction of Power Using ANN in PV Systems," *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology*, Tirunelveli, India, pp. 1286-1292, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[38] Debanjali Sarkar, Taimoor Khan, and Fazal Ahmed Talukdar, "Hyperparameters Optimization of Neural Network Using Improved Particle Swarm Optimization for Modeling of Electromagnetic Inverse Problems," *International Journal of Microwave and Wireless Technologies*, vol. 14, no. 10, pp. 1326-1337, 2021. [CrossRef] [Google Scholar] [Publisher Link]