DOI: 10.24996/ijs.2024.65.12.34

ISSN: 0067-2904

Artificial Neural Network-Salp-Swarm Algorithm for Stock Price Prediction

Zuriani Mustaffa[1]*, Mohd Herwan Sulaiman[2], Azlan Abdul Aziz[3]

[1]*Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang, Malaysia*
[2]*Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang, Malaysia*
[3]*Faculty of Computer Science and Mathematics, Universiti Teknologi Mara, Arau, Perlis, Malaysia*

Received: 9/11/2022 Accepted: 7/11/2023 Published: 30/12/2024

Abstract

Predicting stock prices is a challenging task due to the numerous factors that impact them. The dataset used for analyzing stock prices often displays complex patterns and high volatility, making the generation of accurate predictions difficult. To address these challenges, this study proposes a hybrid prediction model that combines the salp-swarm algorithm and the artificial neural network (SSA-ANN). The SSA is used to optimize the weights and biases in the ANN, resulting in more reliable and accurate predictions. Before training, the dataset is normalized using the min-max normalization technique to reduce the influence of noise. The effectiveness of the SSA-ANN model is evaluated using the Yahoo stock price dataset. The results show that the SSA-ANN model outperforms other models when applied to normalized data. Additionally, the SSA-ANN model is compared with other two hybrid models: the ANN optimized by the Whale Optimization Algorithm (WOA-ANN) and Moth-Flame Optimizer (MOA-ANN), as well as a single model, namely the Autoregressive Integrated Moving Average (ARIMA). The study's findings indicate that the SSA-ANN model performs better in predicting the dataset based on the evaluation criteria used.

Keywords: Artificial Neural Network, machine learning, optimization, Salp-swarm algorithm, stock price prediction, time series prediction.

1. Introduction

Given the significance of the presence of noise, intricate dynamics, nonlinearity, non-parametric nature, and chaotic behavior observed within financial data, predicting stock prices is a challenging and highly sought-after research topic that is crucial for investors to plan strategies to achieve favorable returns. Understanding future trends and the changing characteristics of stock prices is essential for investors to make an informed investment decision, leading to increased profits. Furthermore, stock prices carry significance beyond being merely indicative of a nation's economic expansion; they also reflect the operational effectiveness of enterprises and the broader business [1]. However, accurately predicting stock prices is demanding due to their non-linear and highly volatile nature [2], and they are influenced by various unpredictable variables such as policies, market conditions, company operations, and investors' inclinations, making them complex nonlinear dynamic systems [3].

Diverse approaches have been put forward in order to attain optimal precision when predicting stock prices. These methods include statistical techniques such as autoregressive

* Email: zuriani@umpsa.edu.my

conditional heteroskedasticity (ARCH), generalized autoregressive conditional heteroskedasticity (GARCH), and autoregressive integrated moving average (ARIMA) [4]. However, owing to several limitations inherent in statistical techniques, the emergence of machine learning techniques has come to the fore, with a particular focus on artificial neural networks (ANN). Recent studies indicate that ANN outperforms statistical methods in terms of prediction accuracy, thus substantiating its effectiveness in this domain.

Since the first use of ANN in predicting the daily return of IBM ordinary stock [5], the application of ANN in time series prediction, particularly for stock prices, has gained significant attention. Drawing inspiration from the functional arrangement of biological neurons within the human brain, ANN possesses the capacity to comprehend patterns and associations between input and output data through a training process. This ability enables ANN to effectively grasp intricate and non-linear connections among variables that frequently manifest in various time series data, including stock prices. Besides, ANN is also known for its advantage in handling large datasets.

Despite its effectiveness in addressing predictive tasks, ANN relies significantly on the values assigned to its weights and biases. These parameters must be appropriately fine-tuned during training to improve the precision of predictions. The optimization of these parameters revolves around identifying the optimal values that minimize the difference between the estimated output and the actual output. This objective can be accomplished through the utilization of optimization algorithms. Within this research, the values pertinent to the parameters of concern are fine-tuned through the utilization of the Salp-Swarm Algorithm (SSA). An overview of both ANN and SSA can be found in Sections 3 and 4, respectively.

2. Related Work

Before the use of machine learning methods, conventional statistical techniques such as ARIMA [4] were favored for stock price prediction. However, ARIMA is unable to capture the nonlinear patterns [6] that are exhibited in stock prices. Also, autoregressive ARCH and GARCH models are better at predicting time series data that is mostly smooth [7-9]. This is different from the features that govern stock price time series datasets. The rise of machine learning methodologies, specifically artificial neural networks (ANN), has paved the way for addressing non-linear time series prediction. Numerous studies have convincingly reported the effectiveness of ANN for stock price prediction, such as in [7], where the Harmony Search Algorithm (HSA) was employed to select the number of parameters for prediction purposes. Compared to ANN optimized by Genetic Algorithm (GA) and single ANN, the HSA-ANN demonstrated superior results for the dataset of interest.

Metaheuristic methods and ANN have also been mixed in other studies to try to predict stock prices. For example, [8] used three different types of ANN: back propagation neural network (BPNN), radial basis function neural network (RBFNN), and time delay neural network (TDNN).Among the metaheuristic approaches, including particle swarm optimization (PSO), GA, and artificial bee colony (ABC), PSO-BPNN yielded the best output based on the evaluation metrics used. Alternatively, a method for predicting stock prices was proposed in [9], which combines meta-learning with the decomposition-based model, VML. The study demonstrated that this approach is more effective than other cutting-edge approaches that were examined. A study in [10] compared the performance of the BPNN and GARCH models in predicting stock prices and stock indexes. Evaluated based on MSE and R2, BP-ANN is better than GARCH in terms of index prediction or single stock prediction. Moving on, a study from [11] showed how to use hybrid normalization-based intersection feature selection along with ANN to predict daily stock movement. In comparison against Support Vector Machines (SVM) and K-Nearest Neighborhood (KNN), the study underscored the superior performance of ANN in stock price predictions. It is worth mentioning that the hybrid normalization techniques worked better when combined with the integrated intersection facture selection techniques.

In addition to finance, ANN has been applied to prediction tasks in various fields, such as thermal engineering [12]. The back-propagation algorithm was used to guess the temperature of the ground below. The results were good, as shown by measurements like mean absolute percentage error (MAPE), mean absolute deviation (MAD), mean square error (MSE), and root mean square error (RMSE). ANN has also been used for predicting security breaches in cybercrimes, as demonstrated in [13]. In this study, specific attributes, including biometric data, were fed to the ANN to increase prediction accuracy. Furthermore, [14] showed that ANN was good at predicting the value of biomass heating, and they found that an ANN trained with the Levenberg-Marquardt algorithm was more accurate than the other training algorithms that were tested. A previous study [15] presented a hybrid approach based on ANN and simulated annealing (SA) for temperature data prediction in Baghdad city. In this approach, SA was employed to optimize the weights of the ANN, and the method produced good prediction accuracy. A study in [16] used ANN to predict missing open hole logs of the Mishrif Formation, which included sonic, neutron, density, and deep resistivity. Findings of the study showed ANN provides good accuracy.

Progressing further, a study conducted in [17] showcased the effectiveness of ANN in predicting students' performance within Massive Open Online Courses (MOOCs). To construct the predictive framework, the study integrates ANN with Long Short-Term Memory (LSTM), henceforth referred to as ANN-LSTM. A comparison study of several cutting-edge models, such as the recurrent neural network (RNN) and the gated recurrent unit (GRU), showed that the ANN-LSTM was much better at making predictions. Focusing on the domain of renewable energy, the research outlined in [18] presented a prediction of hourly wind speed in Tamil Nadu. The prediction task was accomplished by proposing a hybrid model that combined an ANN with a Grey Wolf Optimizer, resulting in the designated approach of ANN-GWO. Concerning the importance of achieving optimal values of parameters in ANN, the study proposed the employment of GWO to effectively address the optimization issue. For the sake of comparison, the suggested hybrid model was compared against various hybrid ANN models, encompassing examples such as ANN optimized using Artificial Bee Colony (ANN-ABC) and ANN optimized through Particle Swarm Optimization (ANN-PSO), among others. Notably, the findings of the study suggested the superiority of ANN-GWO.

This study suggests using a combination of ANN and the Salp-Swarm Algorithm (SSA), which is based on Swarm Intelligence (SI), to make predictions about stock prices. This is because SI is good at making predictions. In this approach, the SSA is utilized as an optimization tool for optimizing the weight and bias in the ANN.

The remaining sections of this paper are structured as follows: In Section 1, the background study is presented, while Section 2 discusses the related existing works. Section 3 elaborates on the ANN prediction model, followed by the presentation of the theoretical and mathematical model of the SSA in Section 4. The methodology employed, encompassing the dataset, data normalization, the SSA-ANN prediction model, and the evaluation, is outlined in Section 5. Section 6 provides an analysis and discussion of the obtained results, and finally, Section 7 offers a conclusion based on the findings.

3. Artificial Neural Network

ANN is well-known for being a strong method in machine learning since it can mimic and understand complicated patterns in messy data. When it comes to predicting stock prices, a specific kind of ANN called Multilayer Perceptron (MLP) architecture is utilized. MLP is a type of feedforward supervised learning network that is trained using the Back Propagation (BP) algorithm. This model is structured with three layers: an input layer, multiple hidden layers, and an output layer. Referring to Figure 1, the input layer is composed of various elements, including the disparity between high and low prices (HL), closing and opening prices (CO), and moving averages spanning 7 (MA7), 14 (MA14), and 21 (MA21) days. Additionally, it encompasses the standard deviation of the stock prices over the preceding seven days (Std7), as suggested in [19]. Meanwhile, the output corresponds to the closing prices of the following day.

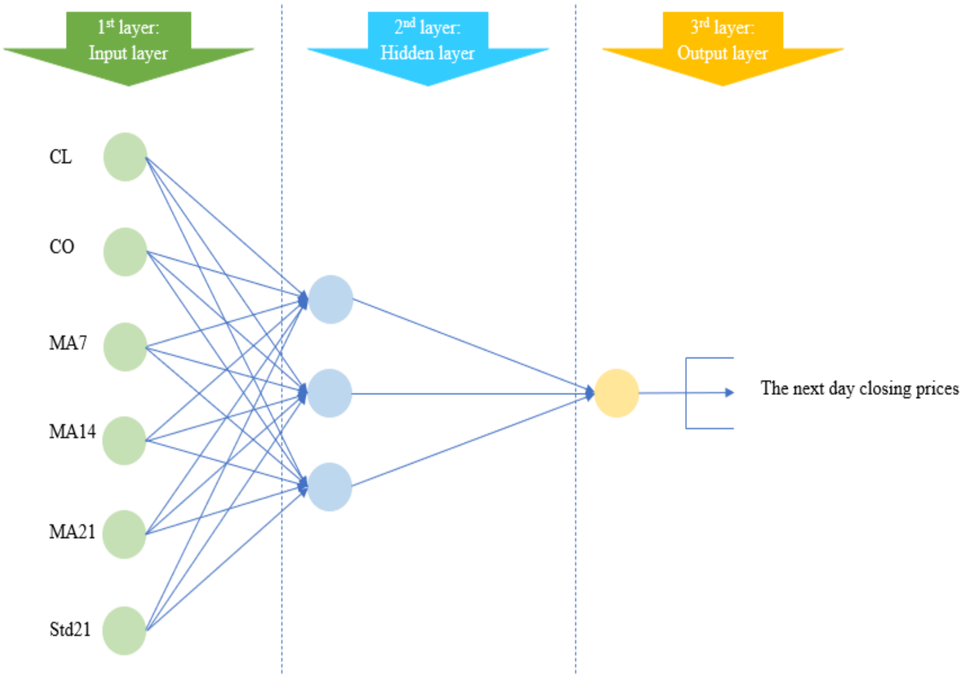

Figure 1: ANN Stock Price Prediction Model

In the figure provided, the input layer gets signals from the source and sends them to all the hidden neurons located within the hidden layer. The weight of each input load is multiplied and added before being sent to the neurons located in the hidden layer, or activation layer. Later, the total weight is calculated and propagated to the output layer, which consists of only one neuron that produces the predicted value; in this case, it is the next day's closing stock price.

4. Salp-Swarm Algorithm

The salp-swarm algorithm (SSA) was formulated by drawing inspiration from the distinct swarming behavior exhibited by salps in their natural habitat. Salps are transparent, barrel-shaped creatures belonging to the family Salpidae. They exhibit a unique behavior in which they form long chains, as shown in Figure 2. This behavior inspired the development of the SSA algorithm.

Figure 2: An example of a salp chain [20]

In the initialization phase of SSA, the population of salps is divided into two parts: the first part consists of leaders, and the second part consists of followers. The leader takes a position on the front line of the chain and guides the swarm, while the followers follow each other. In SSA, there is only one control parameter, which is the maximum number of iterations.

Mathematically, the arrangement of salps is established within an *n*-dimensional space, where *n* represents the count of variables relevant to the current problem.

$$x_j^i = \begin{cases} F_j + c_1\left[\left(ub_j - lb_j\right)c_2 + lb\right] & c_3 \geq 0 \\ F_j - c_1\left[\left(ub_j - lb_j\right)c_2 + lb\right] & c_3 < 0 \end{cases} \tag{1}$$

where;

x_j^i = the position of the first salp i.e. the leader

F_j = food source position in the *j*th dimension

ub_j and lb_j = upper and lower bound of *j*th dimension, respectively

c_1, c_2 and c_3 = random numbers

As outlined in (1), x_j^i solely adjusts its position relative to the food source. Meanwhile, c_1 serves a vital role in harmonizing both exploitation and exploration processes. Its definition is provided in (2):

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \tag{2}$$

where *l* indicates the current iteration and *L* signifies the maximum number of iterations. Both c_2 and c_3 are random numbers uniformly generated within the range of [0,1]. Conversely, for followers, their position are updated according to (3):

$$x_j^i = \frac{1}{2}at^2 + v_0 t \tag{3}$$

where;
$i \geq 2$
x_j^i = position of *i*th follower salp in *j*th dimension
t = time
v_0 =initial speed
$a=\frac{v_{final}}{v_0}$, where $v=\frac{x-x_0}{t}$

In optimization, time is iteration. Due to that, the discrepancy between iterations is equal to 1, and considering v_0 =0, this is defined as in (4):

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \tag{4}$$

$i \geq 2$
x_j^i = position of *i*th follower salp in *j*th dimension

The simulation on salp chain can be simulated using (1) to (4). The algorithm of SSA is shown in Algorithm 1.

Algorithm 1: SSA Algorithm
1. Start
2. Initialize the salp population x_i (i=1, 2, ..., n), considering *ub* and *lb*
3. **While** (end condition is not satisfied)
4. Calculate fitness function of each agent (salp)
5. F=the best search agent
6. Update c_i using (2)
7. **For** x_i (each salp)
8. **If**(i==1)
9. Update the position of the leading salp by (1)
10. **Else**
11. Update teh position of the follower salp by (4)
12. **End**
13. **End**
14. Update the salps based on *ub* and *lb*
15. Return F

5. Methodology

In this study, a hybrid prediction model for stock price is constructed by hybridizing the notable machine learning technique ANN with an optimization algorithm called SSA. The methodology for developing the model involved data collection, data normalization, the development of the SSA-ANN prediction model, and finally, the evaluation of the model's performance.

5.1 Details of Dataset

For this research, a historical dataset containing Yahoo stock prices was utilized, which is freely available from [21]. The dataset contains daily records and covers the period from

November 2015 until November 2020, with no missing values. It includes fundamental information about the stock prices, namely high, low, open, close, volume, and adjacent close. The dataset is free from outliers, meaning that there are no data points that deviate significantly from the overall pattern, and it also does not contain any missing values, indicating that all data points are complete and available for analysis. Table 1 shows a sample of the dataset.

Table 1: Sample of the dataset

Date	High	Low	Open	Close	Volume	Adjacent Close
12/12/2015	2047.2700	2008.8000	2047.2700	2012.3700	4301060000.0000	2012.3700
12/13/2015	2047.2700	2008.8000	2047.2700	2012.3700	4301060000.0000	2012.3700
12/14/2015	2022.9200	1993.2600	2013.3700	2021.9399	4612440000.0000	2021.9399
12/15/2015	2053.8701	2025.5500	2025.5500	2043.4100	4353540000.0000	2043.4100
12/16/2015	2076.7200	2042.4301	2046.5000	2073.0701	4635450000.0000	2073.0701
12/17/2015	2076.3701	2041.6600	2073.7600	2041.8900	4327390000.0000	2041.8900
12/18/2015	2040.8101	2005.3300	2040.8101	2005.5500	6683070000.0000	2005.5500
12/19/2015	2040.8101	2005.3300	2040.8101	2005.5500	6683070000.0000	2005.5500
12/20/2015	2040.8101	2005.3300	2040.8101	2005.5500	6683070000.0000	2005.5500
12/21/2015	2022.9000	2005.9301	2010.2700	2021.1500	3760280000.0000	2021.1500
12/22/2015	2042.7400	2020.4900	2023.1500	2038.9700	3520860000.0000	2038.9700

5.2 Input Derivation

New input variables were derived from the original dataset to produce the following new variables, as tabulated in Table 2:

Table 2: Input Derivation

Variable	Description
HL	Difference between high and low prices
CO	Difference between close and open price
7_dMA	7 days moving average
14_dMA	14 days moving average
21_dMA	21 days moving average
7_dStdDev	Standard deviation for the past 7 days

The derived input variables were then split into a 0.7:0.3 ratio for training and testing, respectively, as tabulated in Table 3. The total number of instances for this study is 1824, with 1277 instances allocated for training (70%) and the remaining 547 instances employed for testing (30%). A sample of the input and output data is shown in Table 4.

Table 3: Data Division for Training and Testing

Input	Training Data	Testing Data	Output
HL CO 7_dMA 14_dMA 21_dMA 7_dStdDev	1277 instances (0.7)	547 instances (0.3	Next day stock prices

Table 4: Sample of Input and Output: Raw Data

10	HL	CO	7_dMA	14_dMA	21_dMA	7_dStdDev	Output
12/12/2015	38.4700	-34.9000	2030.0857	2033.4350	2041.7224	24.0159	2012.3700
12/13/2015	38.4700	-34.9000	2029.1115	2036.9079	2043.2257	24.9736	2021.9399
12/14/2015	29.6600	8.5699	2028.1372	2040.3807	2044.7291	25.8531	2043.4100
12/15/2015	28.3201	17.8600	2028.0243	2042.8493	2044.2872	25.8864	2073.0701
12/16/2015	34.2899	26.5701	2027.3900	2045.3457	2043.0157	25.4980	2041.8900
12/17/2015	34.7101	-31.8700	2026.1357	2044.6522	2039.0724	22.9668	2005.5500
12/18/2015	35.4801	-35.2600	2028.8643	2044.7986	2034.3676	26.0747	2005.5500
12/19/2015	35.4801	-35.2600	2036.7843	2047.5407	2030.3905	26.2322	2005.5500
12/20/2015	35.4801	-35.2600	2044.7043	2050.2829	2026.4133	23.4521	2021.1500
12/21/2015	16.9700	10.8800	2052.6243	2053.0250	2022.4362	16.2947	2038.9700
12/22/2015	22.2500	15.8199	2057.6743	2052.4186	2017.7943	8.5535	2064.2900

5.3 Data Normalization

To mitigate the impact of noise in the dataset, min-max normalization was applied, as defined below:

$$v'_i = \frac{v_i - min_a}{max_a - min_a} \tag{5}$$

where min_a and max_a indicate the minimum and maximum values of an attribute, respectively. The equation maps a value v_i to v'_i. By using min-max normalization, it preserves the relationships among the original data values.

After normalization, the sample of input and output tabulated in Table 4 is transformed into the sample shown in Table 5.

Table 5: Sample of Input and Output: Normalized Data

Date	HL	CO	7dMA	14dMA	21dMA	7dStdDev	Output
12/12/2015	0.1612	0.3959	0.3928	0.4985	0.5377	0.1255	0.1020
12/13/2015	0.1612	0.3959	0.3924	0.4997	0.5381	0.1308	0.1073
12/14/2015	0.1202	0.5452	0.3920	0.5008	0.5386	0.1357	0.1192
12/15/2015	0.1140	0.5771	0.3920	0.5016	0.5385	0.1358	0.1357
12/16/2015	0.1417	0.6070	0.3917	0.5024	0.5381	0.1337	0.1184
12/17/2015	0.1437	0.4064	0.3913	0.5022	0.5368	0.1197	0.0982
12/18/2015	0.1473	0.3947	0.3923	0.5022	0.5353	0.1369	0.0982
12/19/2015	0.1473	0.3947	0.3954	0.5031	0.5341	0.1377	0.0982
12/20/2015	0.1473	0.3947	0.3985	0.5040	0.5328	0.1224	0.1068
12/21/2015	0.0613	0.5531	0.4016	0.5049	0.5316	0.0830	0.1167
12/22/2015	0.0858	0.5701	0.4035	0.5047	0.5301	0.0403	0.1308

5.4 SSA-ANN Prediction Model

The primary goal of this study is to optimize the weight and bias values of the ANN model in order to minimize the objective function, which is the mean square error. To achieve this objective, this study uses the SSA algorithm as an optimization tool for these parameters. The hybridization process involves integrating the SSA function into the ANN, where the SSA will run until maximum iteration is reached. This process will optimize the parameters of interest, and it will be repeated until the output obtained is satisfactory. The flowchart of SSA-ANN is visualized in Figure 3.

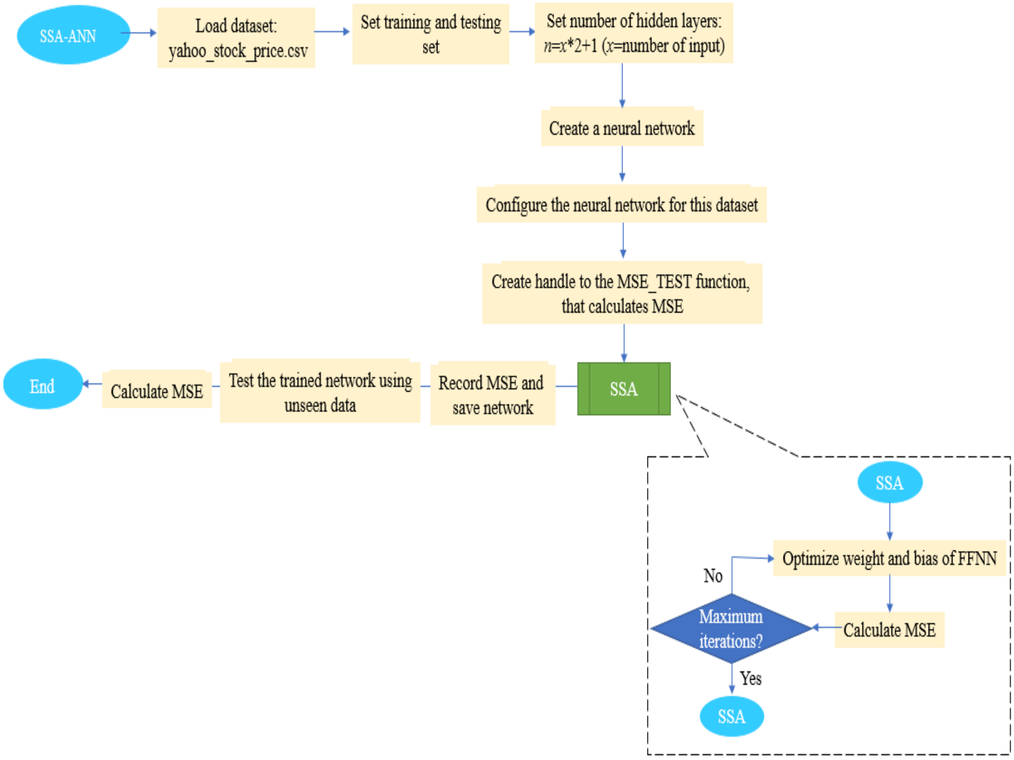

Figure 3: Flowchart of SSA-ANN for Stock Price Prediction

5.5 Performance Evaluation Criteria

In order to assess performance, two quantitative measures were employed, viz., mean absolute percentage error (MAPE) and mean square error (MSE). Their definitions are as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - Y_i)^2 \tag{6}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - Y_i|}{y_i}.100 \tag{7}$$

Here, y_i denotes the target value for the ith data point, Y_i signifies the predicted value at period i, and n denotes the number of prediction periods. The MSE interprets the average mean square error between the predicted value and the target value. Meanwhile, MAPE computes the average of the absolute errors across dataset entries, expressed as a percentage. Consequently, lower values of these metrics correspond to superior prediction performance.

6. Results and Discussion

The simulation phase of this study was conducted in two stages. In the first stage, the SSA-ANN experiment was carried out using two types of datasets: raw and normalized datasets of stock prices. The normalized dataset was obtained by applying the min-max normalization techniques, as described in Section 5.3. The results obtained from the simulation showed that the SSA-ANN model was able to produce a lower prediction error when fed with the normalized dataset. The MSE recorded when using normalized data was 0.0344, which is 0.6264 lower than the MSE produced using a raw dataset, as shown in Table 6. Moreover, consistent results were obtained in terms of MAPE, where the SSA-ANN model was able to achieve a lower error rate of 3.7016% when using a normalized dataset compared to the raw dataset, which had a higher error rate of 10.3940% MAPE.

Table 6: SSA-ANN using a Raw Dataset

Methods	MSE	MAPE (%)
Raw data	0.6608	10.3940
Normalized data	0.0344	3.7016

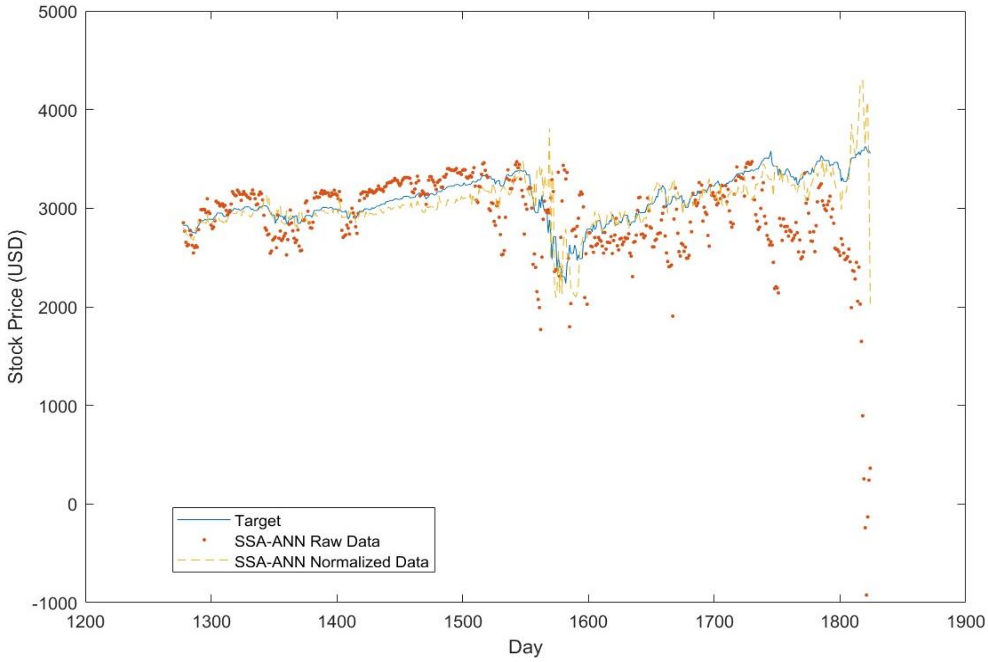

Figure 4: Graph comparison for SSA-ANN: normalized vs. raw dataset

Figure 4 shows what SSA-ANN can do. When the hybrid algorithm is applied to normalized data (shown by the dashed line symbol), the results are closer to the target value. Conversely, when raw data is fed to the model, the produced prediction values are much more scattered (indicated by the dot symbol). By performing data normalization, the features are transformed to be on a similar scale, which promotes better learning in the prediction model. Consequently, good generalizations in prediction can be achieved.

In the next phase of the experiment, the normalized dataset was used to compare the performance of SSA-ANN with two other hybrid prediction models, namely ANN optimized by the Whale Optimization Algorithm [22] and Moth Flame Optimizer [23], and a single model, namely ARIMA. The results, as shown in Table 7, indicate that the SSA-ANN was able to produce lower error rates in terms of MSE and MAPE compared to the other three models. Specifically, the error rates produced by ARIMA were much larger than the other two models, with a MSE of 6.0214 and a MAPE of 78.9184.

Table 7: SSA-ANN vs. compared approaches using Normalized Dataset

Methods	MSE	MAPE (%)
SSA-ANN	0.0344	3.7016
WOA-ANN	0.7124	7.1315
MFO-ANN	1.5505	24.6898
ARIMA	6.0214	78.9184

Figure 5 displays the visualization of predicted values vs. target values generated by all identified prediction techniques. The inferior performance of ARIMA becomes evident when the predicted values are unable to capture the non-linear patterns in stock prices. Meanwhile, for the MFO-ANN, it is apparent in the figure that the prediction values, indicated by dots, are mostly scattered and do not align well with the target values.

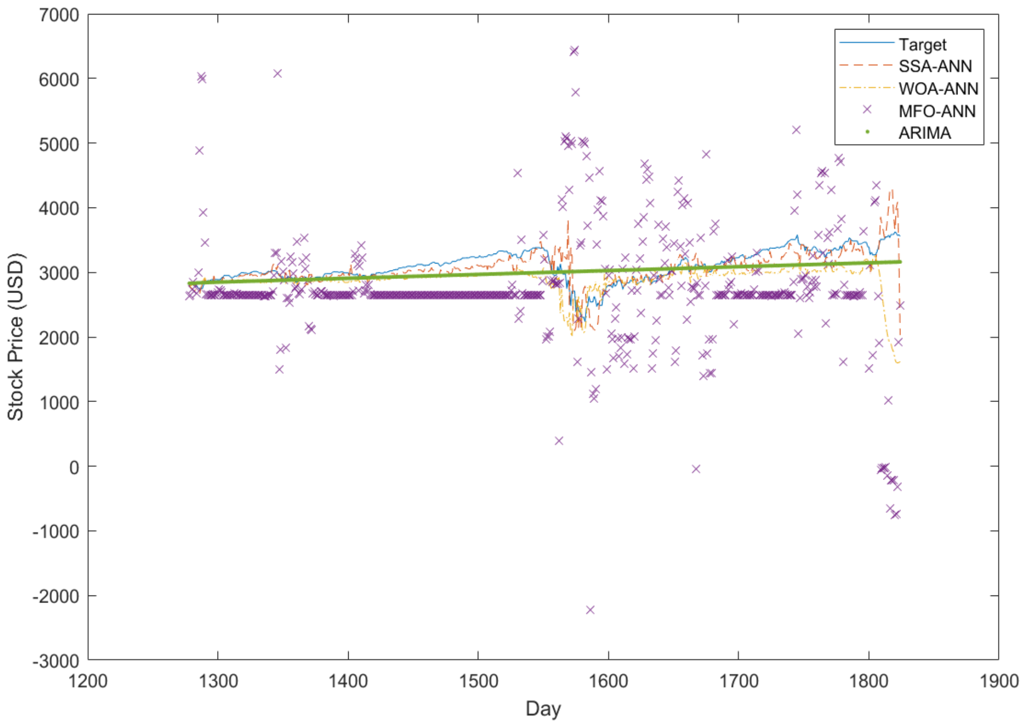

Figure 5: Graph Comparison for Stock Price Prediction: SSA-ANN vs. WOA-ANN vs. MFO-ANN

Furthermore, a paired sample T-test was employed to assess the effectiveness of the SSA-ANN model. At a significance level of 0.05%, Table 8 shows that the results show a big difference in the means between SSA-ANN and the other hybrid algorithms we looked at, which are WOA-ANN, MFO-ANN, and ARIMA.

Table 8: Significant Test: SSA-ANN vs. WOA-ANN vs. MFO-ANN vs. ARIMA

Hybrid Methods	**Sig. (2-tailed)**
SSA-ANN vs. WOA-ANN	0.0000
SSA-ANN vs. MFO-ANN	0.0000
SSA-ANN vs. ARIMA	0.0000

7. Conclusion

Over the years, numerous studies have been proposed for stock price prediction, and they are progressively ongoing today. This indicates the criticality of stock price prediction to many parties, be it the government, private sector, or individual. As the quest for accurate stock price prediction persists, this research contributes by introducing a novel hybrid prediction model. This model harnesses the capability of a metaheuristic algorithm known as SSA to optimize an ANN. This hybridization serves as an automated mechanism for fine-tuning the weights and biases within the ANN, resulting in enhanced generalization capabilities for stock price prediction. The experimentation phase of our proposed model was conducted using the Yahoo stock price dataset, with the target variable being the next day's stock prices.

Upon completion of the simulation and analysis, our findings demonstrate that the SSA-ANN hybrid model consistently achieved lower error rates, particularly when applied to a normalized dataset. Furthermore, when compared to alternative approaches such as WOA and MFO hybrid with ANN and a single method, ARIMA, the SSA-ANN model exhibited clear superiority in the context of stock price prediction. This study underscores the potential of hybridizing metaheuristic optimization techniques with machine learning algorithms to yield more accurate and robust predictions in the domain of stock price predictions. The results obtained here hold promise for improving decision-making processes and investment strategies in financial markets.

Acknowledgement

This study received support from the Ministry of Education Malaysia (MOE) through the Fundamental Research Grant Scheme (FRGS/1/2019/ICT02/UMP/02/4).

References

[1] B. Gülmez, "Stock Price Prediction with Optimized Deep LSTM Network with Artificial Rabbits Optimization Algorithm," *Expert Systems with Applications*, vol. 227, p. 120346, 2023.
[2] L. Lu, Z. Pei, P. Chen, Z. Gao, Z. Gan, and K. Feng, "An Improved Quantile-Point-Based Evolutionary Segmentation Representation Method of Financial Time Series," *The International Arab Journal of Information Technology*, vol. 19, no. 6, pp. 873–883, 2022.
[3] G. Liu and W. Ma, "A quantum artificial neural network for stock closing price prediction," *Information Sciences*, vol. 598, pp. 75–85, 2022.
[4] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," in *Proceedings of the UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.
[5] J. Chen, Y. Wen, Y. A. Nanehkaran, M. D. Suzauddola, W. Chen, and D. Zhang, "Machine learning techniques for stock price prediction and graphic signal recognition," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 106038, 2023.
[6] P. Ping-Feng and L. Chih-Sheng, "A hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.
[7] M. Gocken, M. Ozcalici, A. Boru, and A. T. Dosdogru, "Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction," *Expert Systems with Applications*, vol. 44, no. February, pp. 320–331, 2016.
[8] K. C. S, "Hybrid models for intraday stock price forecasting based on artificial neural networks and metaheuristic algorithms," *Pattern Recognition Letters*, vol. 147, pp. 124–133, 2021.
[9] T. Liu, X. Ma, S. Li, X. Li, and C. Zhang, "A stock price prediction method based on meta-learning and variational mode decomposition," *Knowledge-Based Systems*, vol. 252, p. 109324, 2022.
[10] E. Liu, "Comparison of stock price prediction ability based on GARCH and BP_ANN," in *Proceedings of the 2nd International Conference on Computing and Data Science (CDS)*, Stanford, CA, USA, 2021, pp. 90-93, doi: 10.1109/CDS52072.2021.00021.
[11] B. Kumari and T. Swarnkar, "Forecasting Daily Stock Movement Using a Hybrid Normalization Based Intersection Feature Selection and ANN," *Procedia Computer Science*, vol. 218, pp. 1424–1433, 2023.
[12] A. Afandi, N. Lusi, I. G. N. B. Catrawedarma, Subono, and B. Rudiyanto, "Prediction of temperature in 2 meters temperature probe survey in Blawan geothermal field using artificial neural network (ANN) method," *Case Studies in Thermal Engineering*, vol. 38, p. 102309, 2022.
[13] J. Jagannathan and M. Y. Mohamed Parvees, "Security breach prediction using Artificial Neural Networks," *Measurement: Sensors*, vol. 24, p. 100448, 2022.
[14] I. Veza *et al.*, "Improved prediction accuracy of biomass heating value using proximate analysis with various ANN training algorithms," *Results in Engineering*, vol. 16, p. 100688, 2022.
[15] H. S. I. Harba, "Hybrid Approach of Prediction Daily Maximum and Minimum Air Temperature for Baghdad City by Used Artificial Neural Network and Simulated Annealing," *Iraqi Journal of Science*, vol. 59, no. 1C, pp. 591–599, 2018.
[16] A. K. A. Mohammed and M. K. Dhaidan, "Prediction of Well Logs Data and Estimation of Petrophysical Parameters of Mishrif Formation, Nasiriya Field, South of Iraq Using Artificial Neural Network (ANN)," *Iraqi Journal of Science*, vol. 64, no. 1, pp. 253–268, Jan. 2023.
[17] F. A. Al-Azazi and M. Ghurab, "ANN-LSTM: A deep learning model for early student

performance prediction in MOOC," *Heliyon*, vol. 9, no. 4, p. e15382, 2023.

[18] A. C. Cinar and N. Natarajan, "An artificial neural network optimized by grey wolf optimizer for prediction of hourly wind speed in Tamil Nadu, India," *Intelligent Systems with Applications*, vol. 16, p. 200138, 2022.

[19] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," *Procedia Computer Science*, vol. 167, pp. 599–606, 2020.

[20] Salps." https://en.wikipedia.org/wiki/Salp.

[21] Time Series Forecasting with Yahoo Stock Price." https://www.kaggle.com/datasets/arashnic/time-series-forecasting-with-yahoo-stock-price.

[22] A. L. Seyedali Mirjalili, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[23] S. Mirjalili, "Moth-flame optimization Algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.