# Optimized LSTM Networks with Evolutionary Algorithm for Stock Price Prediction

**Akif Nadeem*, Abdul Basit**, Salman Raza**, Muhammad Abdul Qayum**, Hamid Ali**, Mirza Muhammad Ayzed*****

**School of Intelligent Manufacturing and Control Engineering, Qilu Institute of Technology, Jinan, China (akifnadeem@qlit.edu.cn).*
***Department of Computer Science, National Textile University, Faisalabad, Pakistan, (abdulbasit@ntu.edu.pk, salmanraza@ntu.edu.pk, qayum@ntu.edu.pk, hamid.ali@ntu.edu.pk).*
****School of Computer and Information Engineering, Qilu Institute of Technology, Jinan, China (ayzed.mirza@gmail.com (Corresponding author)).*

**Abstract**: The stock market serves as a crucial venue for the exchange of shares of publicly traded companies, and it serves as an indicator of a country's economic well-being. Despite inherent risks, it also presents the possibility of long-term financial gains. In recent years, Long Short-Term Memory (LSTM) networks have been crucial in financial market analysis for predicting stock prices. This paper introduces a sophisticated LSTM network whose hyperparameters have been optimized using the Attributed Single-Objective Comprehensive Learning Particle Swarm Optimization (A-SOCLPSO) algorithm, resulting in a predictive model for stock price prediction. The model's design is optimized to enhance predicted accuracy by utilizing five years of historical data from 20 businesses listed in the Dow Jones Industrial Average (DJIA), with a 60-day look-back period. The proposed model LSTM-A-SOCLPSO outperformed existing state-of-the-art models, making it a valuable alternative for stock price prediction. This study highlights the efficacy of combining LSTM networks with sophisticated optimization techniques to enhance predictions of financial market trends.

*Keywords*: Stock, Stock price prediction, Evolutionary algorithms, LSTM, PSO.

## 1. INTRODUCTION

Stock ownership is a form of investing where the buyers can own stocks for firms listed on the stock exchange, a critical global business element. The stock market reveals a country's economic level, type, and condition. In the past, market analysis relied on traditional approaches; however, with the emergence of modern Machine Learning (ML) algorithms, the analytical landscape has transformed considerably. Given the vast amount of data and the rise of algorithmic trading, markets are increasingly susceptible to being influenced by automated systems. As a result, transforming the financial industry toward greater socioeconomic sustainability will enhance the effectiveness of ML algorithms in managing potential future disruptions and supporting informed decision-making. (Wang *et al.*, 2024).

Deep Learning (DL), as a key advancement in machine learning, leverages its layered nonlinear architecture to effectively extract meaningful patterns from complex financial time-series data.(Usmani *et al.*, 2024). Recurrent Neural Networks (RNNs) are exceptionally efficient in financial forecasting because they can consider historical data (Selvin et al., 2017; Sachdeva et al., 2019). Unlike traditional neural networks, Recurrent Neural Networks (RNNs) can retain information from previous steps, making them particularly well-suited for forecasting in economic and financial contexts (Lu and Xu, 2024). LSTM networks, a sophisticated form of RNN, address the usual constraints of RNNs by incorporating three gates. However, they also present issues such as intricate computations and challenges in interpreting the internal behavior of LSTM networks. To tackle these challenges, researchers have proposed hybrid models such as Artificial Neural Network (ANN)-Genetic Algorithm (GA) systems (Shin and Lee, 2002) and fuzzy rule extraction methods (Xue *et al.*, 2022). Nevertheless, LSTMs necessitate rigorous parameter tweaking, which involves significant processing resources.

Evolutionary Algorithms (EA) are useful for exploring solution spaces, managing multidimensional problems, and preventing suboptimal solutions. EA leverages its team's knowledge and talents to find nearly optimal solutions in vast search domains, thereby enhancing machine learning, engineering design, and other complex applications. Meanwhile, particle swarm optimization (PSO) is a search approach inspired by bird and fish behavior. It uses a population-based technique to distribute particles depending on individual and group experiences. This helps the swarm identify the best options. Engineering, economics, and artificial intelligence use both methods to solve complex optimization problems.

The optimal parameter configuration for stock price forecasting involves using the LSTM network architecture tuned by the A-SOCLPSO algorithm to construct an accurate prediction model. A-SOCLPSO is a modification of Attributed Multi-Objective Comprehensive Learning Particle Swarm Optimization (A-MOCLPSO) (Ali and Khan, 2013). We fine-tuned architectural factors, including hidden layer LSTM units, learning rate, dropout rate, epochs, and batch size, to influence temporal pattern identifications in datasets. A-

SOCLPSO optimizes hyperparameters by simulating a swarm of particles with one likely solution. The maximal solution (cognitive aspect) and the maximal solution among all particles (social aspect) modify the positions of the particles in the system. Through cooperation and learning in subsequent iterations, the goal is to find the optimum model hyperparameters that minimize the objective function.

The key contributions of this article are as follows:

• This paper proposes the LSTM-A-SOCLPSO model, which combines the LSTM network architecture, A-SOCLPSO optimization algorithm, and Stock Technical Indicators (STIs) to improve stock price prediction. The incorporation of STIs enables the model to extract meaningful features from historical stock data, improving prediction accuracy while addressing market complexity and volatility.

• The A-SOCLPSO algorithm is proposed to optimize key hyperparameters of the LSTM network, such as the number of units, learning rate, dropout rate, epochs, and batch size. The optimization process is guided by minimizing the Mean Squared Error (MSE), ensuring a robust and efficient predictive model.

• Extensive evaluations are conducted to analyze the influence of various STIs, i.e., Moving Average Convergence Divergence (MACD), Simple Moving Average (SMA), Accumulation/Distribution Oscillator (ADO), etc., on the proposed LSTM-A-SOCLPSO model, which integrates the LSTM network with A-SOCLPSO optimization. Comparative experiments with state-of-the-art techniques reveal that the proposed model significantly outperforms benchmark models in terms of prediction accuracy and error metrics, demonstrating its potential for financial market forecasting.

The remaining parts of this paper are organized as follows. Section II presents related works. Section III presents the background concepts and techniques. The proposed model is presented in Section IV. Implementation and evaluation of the proposed model are provided in Section V. Finally, Section VI concludes this paper.

## 2.RELATED WORKS

Deep Neural Network (DNN) play a vital role in the field of finance, namely in tasks like predicting stock market trends within a single day. Overfitting is a prevalent issue in stock market prediction, which detrimentally affects the performance of models, especially when dealing with financial time series that have limited data and challenging data augmentation. In addressing these problems, several deep-learning techniques have been proposed. (Baek and Kim, 2018) introduced the ModAugNet framework, integrating two LSTM modules to enhance forecasting accuracy, particularly for complex financial data like the Standard & Poor's 500 Index (S&P 500) and Korea Composite Stock Price Index 200 (KOSPI 200), (Longet al., 2019) proposed Multilayer Feedforward Neural Network (MFNN), a neural network combining convolutional and recurrent neurons to improve stock price prediction stability. (Chong et al., 2017) analyzed various DL models, highlighting their ability to extract features from complex data without prior knowledge, thereby enhancing prediction precision. (Liu *et al.*, 2022) demonstrated that deep neural networks outperform single-layer models in forecasting the Chinese stock market. However, these studies meet limitations like; the need for careful hyperparameter tuning, insufficient curation of the features, and the time-consuming nature of accurate models like Restricted Boltzmann Machines (RBMs).

Forecasting the future change in stock prices still presents a significant challenge due to the complexity and volatility of market information. To address this, (Manish et al., 2022) proposed an Evolutionary Deep Learning Model (EDLM) using Traditional STIs and a Correlation-Tensor model, which effectively identified patterns in Indian banking stocks. While the approach achieved strong results, it also introduced increased computational complexity due to the use of large word vectors. These challenges highlight the need for more advanced deep-learning techniques and bio-inspired algorithms that can balance accuracy with efficiency while adapting to dynamic market conditions.

The main issue with using the models to predict stock markets is that the models must be optimized for parameters and changed according to the current market situation. (Zhou *et al.*, 2018) have formulated the problem of adversarial training for LSTM and Convolutional Neural Networks (CNN), which can help largely offset the problem. They used the index data, which is publicly available, and to enhance the accuracy of forecasting the direction of the stock prices, they employed the rolling partition training method. (Öztürk, 2023) proposed an ensemble strategy of metaheuristics to select hyperparameters that gave better performance than the existing methods based on several benchmarks. However, both studies have their drawbacks. For instance, it requires an updated model from previous data, and there are issues related to some probabilistic methods like Bayesian Optimization when utilizing metaheuristic initialization.

Due to these challenges, researchers have been encouraged to seek more advanced deep-learning techniques and bio-inspired algorithms to improve on the existing stock market prediction models. One such approach is the use of hybrid LSTM-GA models that optimize temporal window size and network architecture, showing strong performance on financial datasets. (Wang *et al.*, 2022) introduced the Artificial Rabbits Optimization (ARO) algorithm, inspired by rabbit survival strategies, proving its effectiveness in benchmark functions and practical applications like fault detection. (Gülmez, 2023b) demonstrated that the LSTM-ARO model outperforms other models in stock market prediction, validated using the DJIA index dataset. However, these approaches exhibit limitations, such as the restricted generation of new information in GA's crossover process, challenges in interpreting LSTM model outputs, the comparable performance of most existing optimization algorithms, and the necessity for a more comprehensive comparative analysis with other forecasting models.

In proir studies, challenges in tuning deep learning models like RNN-LSTM to capture market stochasticity, time delays, and architectural complexities have been evident. (Zhang and Yang, 2019) addressed this by integrating PSO with LSTM, allowing PSO to fine-tune model weights, loss functions, and other parameters to enhance predictive performance. (Kumar and Haider, 2021) extended this approach by combining PSO with the Flower Pollination Algorithm (FPA), achieving a 4–6% accuracy improvement in one-day stock market forecasting across six stock exchanges. (Ali and Khan, 2013) introduced A-MOCLPSO, which refines PSO by selecting the best position from randomly chosen particles, eliminating reliance on global or local best positions, and proving effective in multi-objective optimization, particularly in network security. Despite these advancements, challenges persist, including application limitations, computational complexity, dependence on specific optimization techniques, and issues with assessment metrics. Additionally, stock data fluctuations may impact predictability and forecast accuracy due to the inherent complexity of market patterns.

This paper integrates an Evolutionary Algorithm (EA) with LSTM networks to develop the LSTM-A-SOCLPSO model, comprising the LSTM network, optimized hyperparameters from A-SOCLPSO, and historical data, aiming to improve stock price prediction accuracy and address key gaps in financial forecasting. While deep learning models like LSTM, CNN, and hybrid approaches show promise, they often face challenges such as overfitting, the need for extensive hyperparameter tuning, and limitations in feature selection. Evolutionary strategies, such as GA, PSO, and ARO, have been used to optimize LSTM architectures, but issues like computational complexity and limited adaptability persist. We propose a novel framework that combines A-SOCLPSO with Stock Technical Indicators (STIs) for enhanced feature selection and model robustness, demonstrating superior predictive performance through extensive experimentation.

## 3. BACKGROUND CONCEPTS AND TECHNIQUES

This section provides an overview of the foundational concepts and methods used in this study, including STIs, LSTM networks, PSO, and the A-SOCLPSO algorithm. These elements form the core of the proposed framework for stock price prediction.

### *3.1 Stock Technical Indicators (STI)*

Technical indicators are one of the approaches to analyzing the stock market and are based on technical elements, namely price and volume indices collected from past data to help in the prediction of future market prices. Some of the most popular indicators are the Moving Average indicators, Relative Strength Index (RSI), MACD, and the Bollinger Bands. Such indicators are very useful in analyzing the status and trends in the market, which can assist the trader in his decisions. The STIs that are incorporated in this research are highlighted in Table 1 below.

**Table 1. Stock Technical Indicators (STI's) used.**

| Name | Mathematical Equation |
|---|---|
| Simple n-day Moving Average (SMA) | $\frac{C_t + C_{t-1} + \cdots + C_{t-n+1}}{n}$ |
| Weighted n-day Moving Average (WMA) | $\frac{n * C_t + (n-1) * C_{t-1} + \cdots + C_{t-}}{n + (n-1) + \cdots + 1}$ |
| Momentum (MOM) | $C_t - C_{t-n+1}$ |
| Stochastic K% (STCK) | $\frac{C_t - LL_{t\_t-n+1}}{HH_{t\_t-n+1} - LL_{t\_t-n+1}} \times 100$ |
| Stochastic D% (STCD) | $\frac{K_t + K_{t-1} + \cdots + K_{t-n+1}}{n} \times 100$ |
| Relative Strength Index (RSI) | $100 - \frac{100}{1 + \frac{\sum_{i=1}^{n-1} UP_{t-i}}{\sum_{i=1}^{n-1} DW_{t-i}}}$ |
| Moving Average Convergence Divergence (MACD)t | $EMA(12)_t - EMA(26)_t$ |
| Signal (n)t | $MACD_t \times \frac{2}{n+1} + Signal(n)_{t-1} \times \left(1 - \frac{2}{n+1}\right)$ |
| Larry William's R% (LWR) | $\frac{HH_{t\_t-n+1} - C_t}{HH_{t\_t-n+1} - LL_{t\_t-n+1}} \times 100$ |
| Accumulation/Distribution (A/D) Oscillator (ADO) | $\frac{H_t + C_t}{H_t - L_t}$ |
| Commodity Channel Index (CCI) | $\frac{M_t + SM_t}{0.015 D_t}$ |

### *3.2 Long Shot-Term Memory (LSTM)*

LSTMs, a specialized type of RNN, address vanishing gradient issues with gating mechanisms to manage information flow, enabling them to capture long-term dependencies in sequential data, making them ideal for tasks like time series forecasting and natural language processing (Srivastava and Mishra, 2021).

### *3.3 Particle Swarm Optimization (PSO)*

Inspired by the collective movement of birds and fish, PSO adapts candidate solutions based on individual and neighboring experiences. Recent innovations in PSO focus on improving convergence, flexibility, and robustness in complex, high-dimensional problems. Despite its successes in machine learning, engineering, and financial modeling, PSO faces challenges like premature convergence and lack of swarm diversity in multimodal and large-scale optimization.

### *3.4 Attributed Single-Objective Comprehensive Learning Particle Swarm Optimization (A-SOCLPSO)*

The A-SOCLPSO algorithm is particularly suitable for solving single-objective optimization problems, such as hyperparameter tuning for deep learning networks. A-SOCLPSO is especially suitable for many types of single-objective Optimization problems. The procedure involves continuous modification of the velocities and positions of the set of potential solutions to the problem by considering both the personal-best solution of the members in the group and the best experience recorded by any other random particle in the swarm. A-SOCLPSO optimizes the performance and reduces the time required to find the best solution by adjusting parameters and learning algorithms in multiple iterations.

## 4. PROPOSED FRAMEWORK: LSTM-A-SOCLPSO MODEL

### *4.1 Overview*

This section presents the proposed LSTM-A-SOCLPSO framework, which integrates an LSTM network for stock price forecasting and the A-SOCLPSO algorithm for hyperparameter optimization. The core contribution lies in the design and application of a single-objective evolutionary optimization algorithm (A-SOCLPSO) to systematically determine the optimal values for key hyperparameters of the LSTM network. The final model combines the optimized hyperparameters with the LSTM architecture and financial time-series data to construct a robust forecasting system. The A-SOCLPSO is an advanced version of the A-MOCLPSO algorithm developed based on the PSO algorithm. A-SOCLPSO also differs from A-MOCLPSO because the former employs the single-objective optimization method instead of the multi-objective one to optimize hyperparameters of machine learning models. This makes it most suitable in models that entail large and complicated search spaces such as the LSTM networks. A-SOCLPSO allows the compromise between the search of the hyperparameters space optimization and the target that is most often the MSE. It determines different hyperparameters such as LSTM units, learning rate, dropout rate, epoch number, and batch size. This focused optimization is deemed suitable for boosting the given model's performance because it yields accurate and reliable forecasts. Fig. (1) illustrates the workflow of the proposed LSTM-A-SOCLPSO.

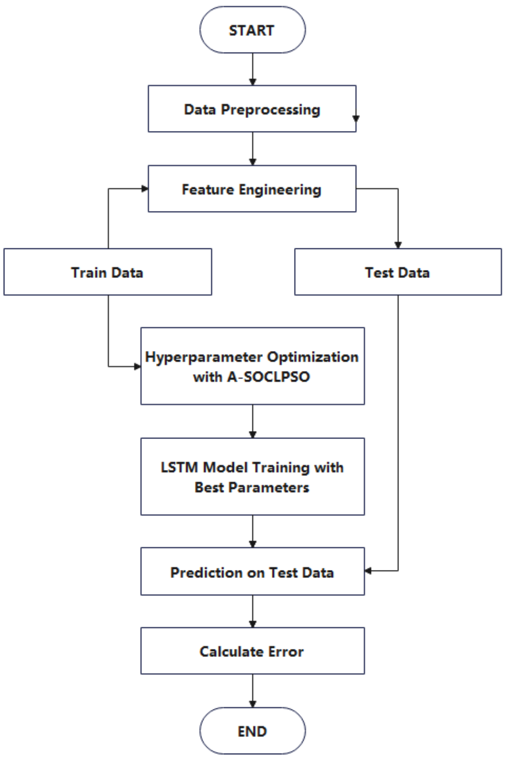


Fig. 1. Workflow of the LSTM-A-SOCLPSO framework including optimization and model training phases.

### *4.2 Optimization*

It also means that in this method, every 'particle' within the swarm is a potential solution. Such potential solutions have distinct hyperparameters which include the number of LSTM units, learning rate, dropout rate, numbers of epochs, size of a batch. In this case, an iterative approach is used to update these particles in such a way that they may search through space and come up with the best solution possible.

The positions of the particles in the hyperparameter space are optimized as influenced not only by their own best solutions (called personal best or pbest), but also the best experience recorded by any other random particle in the swarm. The algorithm also uses inertia weight to decide between exploitation and exploration. Moreover, there are cognitive and social coefficients that help to regulate the impact of the social and personal high scores.

In each iteration, the velocities and positions of the particles are adjusted based on the motion within the search space. It is also necessary to include some random values to maintain diversity to prevent the formation of locally optimal solutions. The fitness of the particles is determined by calculating the MSE of the prediction done by the model on a set of samples. These particles are contained within the bounds given for each hyperparameter that has been defined. Equations (1) and (2) explain how the velocity and position updates are done, while (3) explains how the particles are kept within bounds.

*Algorithm 1: A-SOCLPSO-based LSTM Hyperparameter Optimization*

Input: Search space for hyperparameters (units, learning rate, dropout, epochs, batch size)

Output: Optimal hyperparameter configuration

1. Initialize swarm population with random hyperparameter values
2. Evaluate fitness (MSE) of each particle by training an LSTM with its parameters
3. For each particle:
    a. Update personal best (pbest) if current fitness is better
    b. Randomly select another particle to determine the social best
    c. Update velocity and position using Equations (1)–(3)
4. Enforce boundary constraints
5. Repeat Steps 2–4 until the stopping criterion is met (e.g., max iterations)
6. Return best-found hyperparameters

$$v_{i,d}^{t+1} = w.v_{i,d}^{t} + c_1.r_1.\left(pbest_i^d - x_i^d\right) + c_2.r_2.\left(pbest_j^d - x_i^d\right) \tag{1}$$

$$x_{i,d}^{t+1} = x_{i,d}^{t} + v_{i,d}^{t+1} \tag{2}$$

$$x_{i,d}^{t+1} = \left(x_{i,d}^{t+1}, x_{min}, x_{max}\right) \tag{3}$$

Frequently used notations in this paper are listed in Table 2.

The algorithm repeatedly reconstructs the swarm throughout several cycles and remembers which solutions were found to be better than others at a certain stage. The idea is to minimize the MSE as much as possible and find the best set of hyperparameters that will give a high-accuracy model. In any case, the process stops as soon as the algorithm has been iterated several times.

**Table 2. Frequently used Notations.**

| Symbols | Description |
|---|---|
| $n$ | is the number of days |
| $C_t$ | is the closing price at time t |
| $L_t$ | is the low price at time t, |
| $H_t$ | is the high price at time t, |
| $LL_{t\_t-n+1}$ | the lowest, low prices in the last n days, |
| $HH_{t\_t-n+1}$ | the highest, high prices in the last n days, |
| $M_t$ | $\frac{H_t + L_t + C_t}{3}$ |
| $SM_t$ | $\frac{\sum_{i=0}^{n-1} M_{t-i}}{n}$ |
| $D_t$ | $\frac{\sum_{i=0}^{n-1} \lvert M_{t-i} - SM_t \rvert}{n}$ |
| $v_{i,d}^{t+1}$ | updated velocity of the particle $i$ in dimension $d$ at iteration $t+1$ |
| $w$ | inertia weight |
| $c_1, c_2$ | cognitive and social coefficients |
| $r_1, r_2$ | random numbers |
| $pbest_i^d$ | personal best position of the particle $i$ in dimension $d$ |
| $pbest_j^d$ | best-known position of any other particle j in dimension d |
| $x_{i,d}^t$ | current position of the particle $i$ in dimension $d$ |
| $x_{i,d}^{t+1}$ | updated velocity of the particle $i$ in dimension $d$ at iteration $t+1$ |
| $x_{min}, x_{max}$ | lower and upper bounds for the position |

At that final point, the set of hyperparameters that yields the best accuracy, or error is returned as the answer.

*4.3 Model Construction*

The next step involves training the final LSTM model using the set of hyperparameters tuned through the A-SOCLPSO method. After optimization, the values from the identified parameters are fetched and converted into proper data types. This means that the number of LSTM units, epochs, and batch size must be converted to integers, while the learning and dropout rates are in the form of floats. It is crucial because the hyperparameters define the architecture and learning process of the LSTM model. They also affect the accuracy within which the model can predict future values.

Once the optimum parameters have been identified, the LSTM model is created by defining the structure and setting the parameters of the model using the extracted values. They comprise an LSTM layer followed by a dropout layer, repeated twice to make the entire model. This dropout layer effectively reduces overfitting, where the model is made to add more noise

to some of the input units while training them. The number of LSTM units, which determines the best LSTM units, is used to identify the number of units present in each LSTM layer. Furthermore, the dropout rate is checked using the best dropout rate. The output is generated by the final dense layer of the model, which, in this case, is the expected closing price.

*4.4 Training & Evaluation*

Subsequently, after the process of model architectural configuration is finalized, the Adam optimizer is employed to compile the model with the hyper-tuned parameter used as the learning rate hyperparameter. While training the network, the Adam optimizer is chosen since it offers flexibility when adjusting the learning rate to manage loss. The MSE, which stands for the mean squared error, is the loss function used. MSE is often used for regression problems such as stock price prediction since it determines the average squared difference between the estimated values and the actual observations.

The training process includes using the LSTM model to fit the training dataset, and A-SOCLPSO selected the number of epochs and the number of batches as the best. When going through the training data, the model fine-tunes its parameters, that is, the weights and the biases to minimize the loss function. During each epoch, the model runs through the data in batches and tweaks its settings to improve the system's ability to predict outcomes. After the model's training, its accuracy is then tested using the test dataset to evaluate its efficiency. After creating forecasts for closing prices by the trained model, these values are transformed back to the scale from which the normalization was done. Moreover, the real test data is also inverse transformed for comparison. The model's performance is estimated with the help of MSE, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These measures show the performance of the model by both the average inaccuracy and the overall inaccuracy of the fit of the predictions to the actual data.

*4.5 Evaluation Criteria*

Predictive models used in time series forecasting tasks, such as stock price prediction, are typically evaluated using accuracy metrics, including MSE, RMSE, MAE, and MAPE.

*4.5.1 Mean Squared Error (MSE)*

MSE measures the average squared differences between predicted and actual values, with outliers having a significant impact due to their larger error weights. Equation (4) provides the mathematical equivalent of MSE.

$$\mathrm{MSE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2 \quad (4)$$

Where, n represents the number of observations, $\mathrm{y_i}$ is the actual value of $i$ and $\hat{\mathrm{y}}_\mathrm{i}$ is the anticipated value of i.

*4.5.2 Root Mean Squared Error (RMSE)*

RMSE quantifies the disparity between predicted and actual values, with a lower RMSE indicating better model accuracy. Equation (5) provides the mathematical equivalent of RMSE.

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (5)$$

Where, n represents the number of observations, $y_i$ is the actual value of $i$ and $\hat{y}_i$ is the anticipated value of i.

*4.5.3 Mean Absolute Error (MAE)*

MAE measures the average absolute deviation between predicted and actual values, offering greater stability than MSE and RMSE in the presence of outliers. Equation (6) provides the mathematical equivalent of MAE.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (6)$$

Where, n represents the number of observations, $y_i$ is the actual value of i, $\hat{y}_i$ is the anticipated value of i and $|x|$ represents the absolute value of x.

*4.5.4 Mean Absolute Percentage Error (MAPE)*

MAPE measures the average percentage difference between predicted and actual values, enabling model comparison across datasets, but it may face issues with division by zero or large errors near zero values. Equation (7) provides the mathematical equivalent of MAPE.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \quad (7)$$

Where, n represents the number of observations, $y_i$ is the actual value of i, $\hat{y}_i$ is the anticipated value of i and $|x|$ represents the absolute value of x.

## 5. RESULTS AND DISCUSSION

*5.1 Dataset*

The DJIA is one of the most widely recognized stock indices, representing a broad cross-section of the U.S. economy. For this study, historical stock data from 01/01/2018 to 01/01/2023 was collected using the finance Python library, which retrieves accurate, publicly available financial data from Yahoo Finance (Garita and Garita, 2021), (Yahoo!, 2024). The dataset includes key features such as Open, High, Low, Close, Adjusted Close, and Volume. Among these, the 'Close' price, representing the final traded price at the end of each session, was selected for forecasting due to its importance in summarizing daily market behavior.

To ensure data integrity, all rows with missing values in the 'Close' column were removed during preprocessing. The cleaned 'Close' series was then used to compute a variety of STIs, including SMA, Weighted Moving Average (WMA), Momentum (MOM), Stochastic Oscillator (%K) (STCK), Smoothed Stochastic Oscillator (%D) (STCD), Relative Strength Index (RSI), Exponential Moving Average (EMA), Larry Williams %R (LWR), ADO, Commodity Channel Index (CCI), MACD, and Signal Line (SIG). These indicators quantify different temporal and volatility-related aspects of market behavior, enhancing the learning capacity of forecasting models.

Additional missing values introduced during the calculation of these indicators (due to rolling window operations) were also removed to maintain dataset quality. After this, all numerical features were normalized to the range [0, 1] using MinMaxScaler. This step ensures that all inputs contribute proportionally to training and helps accelerate the convergence of the LSTM network.

Finally, the dataset was split into training and testing sets, with 80% used for training and 20% for testing. This structured preprocessing pipeline, comprising missing value handling, feature engineering, and normalization, ensures data consistency and plays a vital role in improving the model's generalization performance.

*5.2 Evaluation Strategy*

The predictive performance of the proposed LSTM-A-SOCLPSO model is evaluated using the four metrics described in Section 4.5. These metrics allow for a comprehensive assessment of accuracy, stability, and robustness. The same evaluation criteria are applied to all baseline models for fair comparison.

*5.3 Results*

This section presents the results of the proposed LSTM-A-SOCLPSO model in comparison with LSTM models optimized using LSTM-ARO (Gülmez, 2023b) and LSTM-GA (Gülmez, 2023a), as previously introduced. Each model was executed for 50 iterations using identical training and testing datasets. Evaluation was conducted using four standard metrics, MSE, RMSE, MAE, and MAPE, as described in Section 4.5.

The values reported in Tables 3 to 6 represent the average performance of each model across selected DJIA constituent stocks. The tickers and corresponding companies include: AXP (American Express), AMGN (Amgen), AAPL (Apple Inc.), BA (Boeing), CAT (Caterpillar), CSCO (Cisco Systems), CVX (Chevron), GS (Goldman Sachs), HD (Home Depot), HON (Honeywell), IBM (International Business Machines), INTC (Intel), JNJ (Johnson & Johnson), KO (Coca-Cola), JPM (JPMorgan Chase), MCD (McDonald's), MSFT (Microsoft), CRM (Salesforce), V (Visa), and DIS (Walt Disney).

These results provide a comprehensive view of the predictive performance of the tested models. Alongside the tabulated metrics, comparative plots are included to visualize the relative accuracy of each method. Overall, the findings demonstrate the effectiveness of the A-SOCLPSO optimization strategy in improving forecast precision and model robustness across diverse financial instruments.

To compare the performance of four optimization algorithms, LSTM-A-SOCLPSO, LSTM-ARO, LSTM-GA, LSTM-STI, and basic LSTM, we assessed the MSE to predict the stock prices for different tickers. The analysis shows that the LSTM-A-SOCLPSO algorithm maintains a consistently high accuracy rate compared to the other algorithms. For instance, as for JNJ's forecasting, the proposed LSTM-A-SOCLPSO model yields an MSE of 4.761, which is significantly less than the MSE values of LSTM-ARO (5.703), LSTM-GA (5.023), LSTM-STI (6.766) and LSTMs (4.806). As for IBM's LSTM-A-SOCLPSO, the MSE amounts to 6.201, which is less than LSTM-ARO (7.333), LSTM-GA (7.292), LSTM-STI (8.240) and LSTM (7.689). With the aid of the LSTM-A-SOCLPSO model, the MSE of KO is 0.718. Hence, it outperforms LSTM-ARO with a value of 0.957, LSTM-GA with 0.931, LSTM-STI with a value of 2.480, and LSTM with a value of 1.198.

**Table 3. Comparison of MSE Results for Different Models.**

| Ticker | LSTM-A-SOCLPSO | LSTM-ARO | LSTM-GA | LSTM-STI | LSTM |
|---|---|---|---|---|---|
| AXP | 13.735 | 24.287 | 25.869 | 27.451 | 26.707 |
| AMGN | 18.944 | 19.774 | 20.272 | 26.519 | 20.744 |
| AAPL | 14.400 | 22.731 | 24.109 | 17.264 | 29.026 |
| BA | 23.907 | 33.658 | 40.216 | 42.237 | 53.865 |
| CAT | 28.472 | 38.438 | 36.329 | 35.639 | 40.835 |
| CSCO | 1.131 | 0.990 | 1.143 | 1.054 | 1.333 |
| CVX | 24.102 | 23.970 | 21.953 | 47.332 | 26.168 |
| GS | 54.658 | 80.137 | 80.827 | 75.649 | 103.298 |
| HD | 22.710 | 62.531 | 59.791 | 65.965 | 86.977 |
| HON | 9.182 | 13.264 | 14.589 | 17.123 | 17.501 |
| IBM | 6.201 | 7.333 | 7.292 | 8.240 | 7.689 |
| INTC | 1.917 | 1.835 | 1.990 | 5.311 | 5.683 |
| JNJ | 4.761 | 5.703 | 5.023 | 6.766 | 4.806 |
| KO | 0.724 | 0.957 | 0.931 | 2.480 | 1.198 |
| JPM | 7.764 | 10.404 | 12.926 | 8.037 | 11.825 |
| MCD | 58.216 | 20.599 | 21.524 | 23.234 | 18.880 |
| MSFT | 50.665 | 63.322 | 79.253 | 60.032 | 94.190 |
| CRM | 33.647 | 39.904 | 52.945 | 43.754 | 55.754 |
| V | 15.484 | 24.406 | 25.667 | 28.409 | 50.538 |
| DIS | 10.897 | 13.959 | 14.591 | 17.076 | 16.549 |

**Table 4. Comparison of RMSE Result for Different Models.**

| Ticker | LSTM-A-SOCLPSO | LSTM-ARO | LSTM-GA | LSTM-STI | LSTM |
|---|---|---|---|---|---|
| AXP | 3.706 | 4.928 | 5.086 | 5.239 | 5.167 |
| AMGN | 4.353 | 4.446 | 4.502 | 5.150 | 4.555 |
| AAPL | 3.794 | 4.767 | 4.910 | 4.155 | 5.388 |
| BA | 4.889 | 5.801 | 6.341 | 6.499 | 7.339 |
| CAT | 5.336 | 6.199 | 6.027 | 5.970 | 6.390 |
| CSCO | 1.063 | 0.994 | 1.069 | 1.027 | 1.154 |
| CVX | 4.909 | 4.895 | 4.685 | 6.879 | 5.115 |
| GS | 7.393 | 8.951 | 8.990 | 8.698 | 10.164 |
| HD | 4.765 | 7.907 | 7.732 | 8.122 | 9.326 |
| HON | 3.030 | 3.641 | 3.819 | 4.138 | 4.183 |
| IBM | 2.490 | 2.707 | 2.700 | 2.870 | 2.773 |
| INTC | 1.384 | 1.354 | 1.410 | 2.304 | 2.384 |
| JNJ | 2.182 | 2.388 | 2.241 | 2.601 | 2.192 |
| KO | 0.851 | 0.978 | 0.964 | 1.575 | 1.094 |
| JPM | 2.786 | 3.225 | 3.595 | 2.835 | 3.439 |
| MCD | 7.630 | 4.538 | 4.639 | 4.820 | 4.345 |
| MSFT | 7.118 | 7.958 | 8.903 | 7.748 | 9.705 |
| CRM | 5.801 | 6.317 | 7.277 | 6.615 | 7.467 |
| V | 3.935 | 4.941 | 5.066 | 5.330 | 7.112 |
| DIS | 3.301 | 3.736 | 3.820 | 4.132 | 4.068 |

**Table 5. Comparison of MAE Result for Different Models.**

| Ticker | LSTM-A-SOCLPSO | LSTM-ARO | LSTM-GA | LSTM-STI | LSTM |
|---|---|---|---|---|---|
| AXP | 2.921 | 3.804 | 3.848 | 4.197 | 3.964 |
| AMGN | 3.328 | 3.318 | 3.425 | 3.757 | 3.374 |
| AAPL | 2.997 | 3.846 | 3.955 | 3.376 | 4.351 |
| BA | 3.815 | 4.425 | 4.805 | 5.288 | 5.406 |
| CAT | 4.296 | 4.947 | 4.748 | 4.588 | 4.870 |
| CSCO | 0.750 | 0.744 | 0.813 | 0.740 | 0.866 |
| CVX | 3.925 | 3.830 | 3.531 | 5.459 | 3.955 |
| GS | 5.874 | 7.060 | 7.272 | 7.084 | 7.913 |
| HD | 3.817 | 6.206 | 6.026 | 6.268 | 7.414 |
| HON | 2.400 | 2.876 | 3.068 | 3.230 | 3.284 |
| IBM | 1.990 | 2.113 | 2.096 | 2.340 | 2.171 |
| INTC | 1.171 | 1.092 | 1.072 | 1.944 | 2.034 |
| JNJ | 1.744 | 1.912 | 1.795 | 2.183 | 1.753 |
| KO | 0.652 | 0.738 | 0.751 | 1.367 | 0.811 |
| JPM | 2.223 | 2.523 | 2.894 | 2.197 | 2.716 |
| MCD | 6.306 | 3.630 | 3.646 | 3.779 | 3.375 |
| MSFT | 5.587 | 6.584 | 7.583 | 6.314 | 8.011 |
| CRM | 4.584 | 5.024 | 5.879 | 5.307 | 5.951 |
| V | 3.043 | 3.724 | 3.849 | 4.355 | 5.705 |
| DIS | 2.346 | 2.923 | 2.935 | 3.360 | 3.208 |

**Table 6. Comparison of MAPE Result for Different Models.**

| Ticker | LSTM-A-SOCLPSO | LSTM-ARO | LSTM-GA | LSTM-STI | LSTM |
|---|---|---|---|---|---|
| AXP | 0.018 | 0.024 | 0.024 | 0.026 | 0.024 |
| AMGN | 0.013 | 0.014 | 0.014 | 0.015 | 0.014 |
| AAPL | 0.019 | 0.025 | 0.026 | 0.021 | 0.029 |
| BA | 0.025 | 0.027 | 0.030 | 0.034 | 0.035 |
| CAT | 0.021 | 0.025 | 0.024 | 0.023 | 0.024 |

| | | | | | |
|---|---|---|---|---|---|
| CSCO | 0.016 | 0.016 | 0.017 | 0.016 | 0.018 |
| CVX | 0.024 | 0.025 | 0.023 | 0.033 | 0.025 |
| GS | 0.018 | 0.022 | 0.022 | 0.022 | 0.024 |
| HD | 0.026 | 0.021 | 0.020 | 0.021 | 0.024 |
| HON | 0.012 | 0.015 | 0.016 | 0.017 | 0.017 |
| IBM | 0.014 | 0.017 | 0.016 | 0.017 | 0.016 |
| INTC | 0.035 | 0.031 | 0.031 | 0.061 | 0.063 |
| JNJ | 0.010 | 0.011 | 0.011 | 0.013 | 0.011 |
| KO | 0.010 | 0.012 | 0.013 | 0.021 | 0.013 |
| JPM | 0.018 | 0.020 | 0.024 | 0.019 | 0.022 |
| MCD | 0.024 | 0.016 | 0.016 | 0.016 | 0.013 |
| MSFT | 0.023 | 0.025 | 0.029 | 0.024 | 0.031 |
| CRM | 0.028 | 0.029 | 0.034 | 0.031 | 0.035 |
| V | 0.015 | 0.018 | 0.019 | 0.021 | 0.027 |
| DIS | 0.023 | 0.026 | 0.026 | 0.031 | 0.029 |

Specifically, the LSTM-A-SOCLPSO model achieves an MSE of 1.131 for CSCO, which is relatively higher than LSTM-ARO (0.990) but comparatively better than LSTM-GA (1.143), LSTM-STI (1.054), LSTM 1.333. In each case, the evaluation results indicate that LSTM-A-SOCLPSO yields a better fit and a lower MSE than the other investigated algorithms, which makes LSTM-A-SOCLPSO the most optimized model. However, for the MCD stock, the LSTM-A-SOCLPSO model recorded a marginally higher MSE compared to LSTM-ARO and LSTM-GA. Although this result deviates from the model's overall superior performance, it may suggest a minor limitation in adapting to specific stock dynamics or volatility.

The graphical representations of the real and projected stock prices across several models compare their forecasting accuracy. Fig. 2 displays the performance of the LSTM-A-SOCLPSO model. The close alignment between the actual and forecast prices for different stocks reveals the model's exceptional accuracy. Fig. 3 shows the LSTM-ARO model, which exhibits overall satisfactory alignment but occasional deviations, especially in stocks more prone to fluctuations. Fig. 4 displays the LSTM-GA model, which accurately predicts the prices but not as closely as the LSTM-A-SOCLPSO model. Fig. 5 shows the LSTM-STI model, which works satisfactorily but shows more prominent variations in specific equities, such as AMGN and CSCO. Fig. 6 presents the outcomes of the conventional LSTM model, which, although successful, exhibits more frequent and noticeable deviations than the other models. The statistics collectively highlight the superior prediction skills of the LSTM-A-SOCLPSO model, especially in accurately tracking actual stock values.

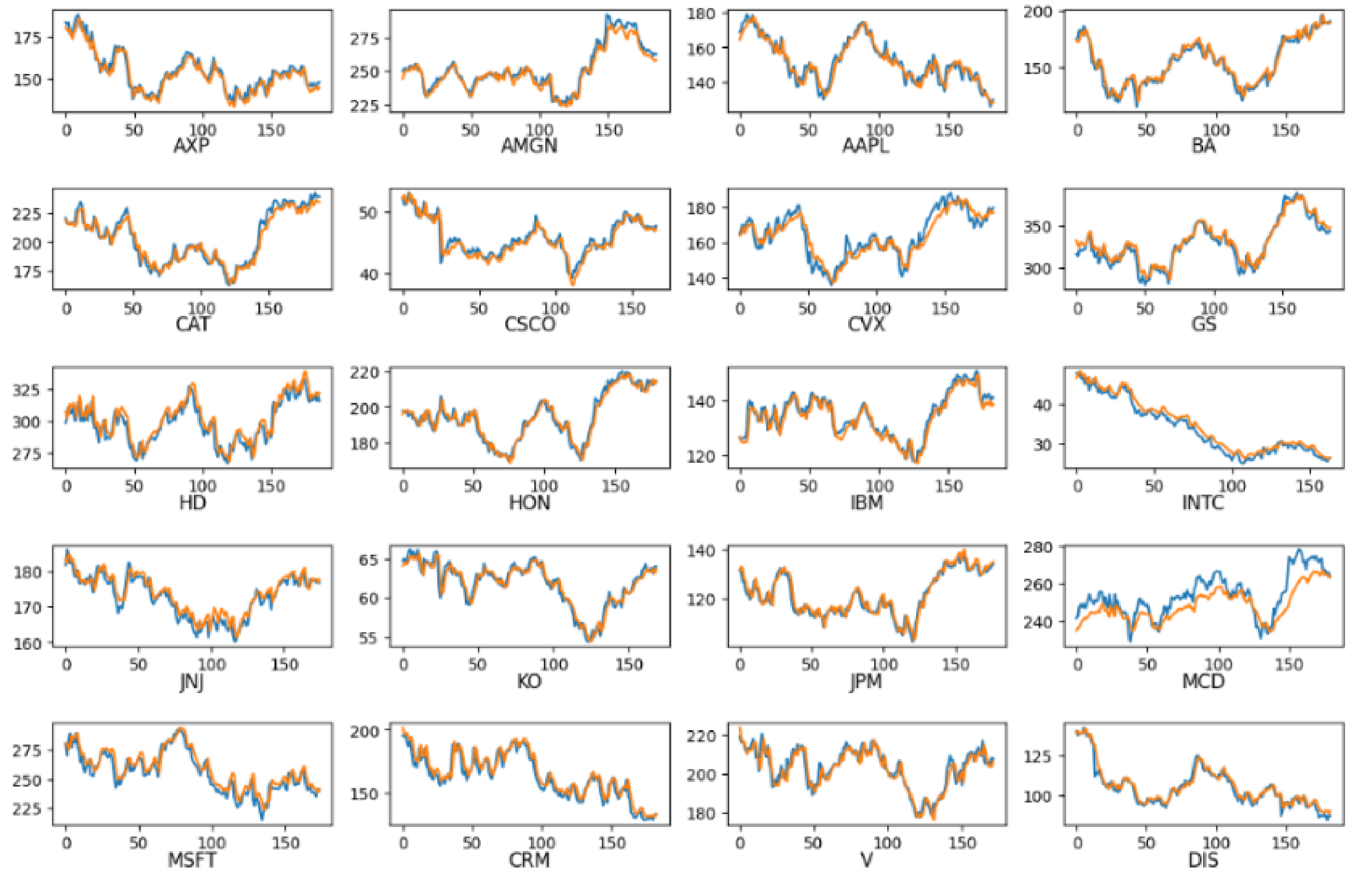


Fig. 2. LSTM-A-SOCLPSO Stock Price Prediction Plots.

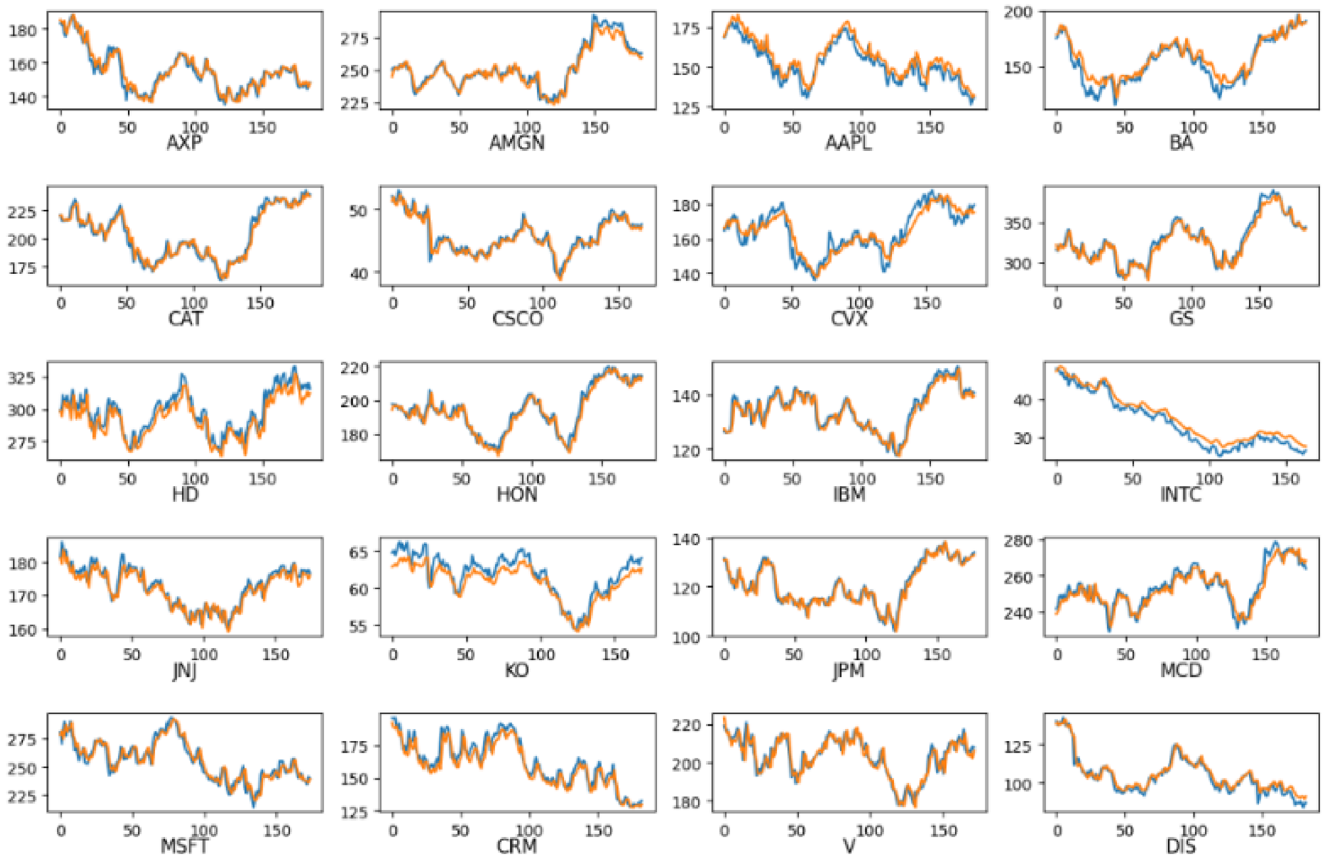


Fig. 3. LSTM-ARO Stock Price Prediction Plots.

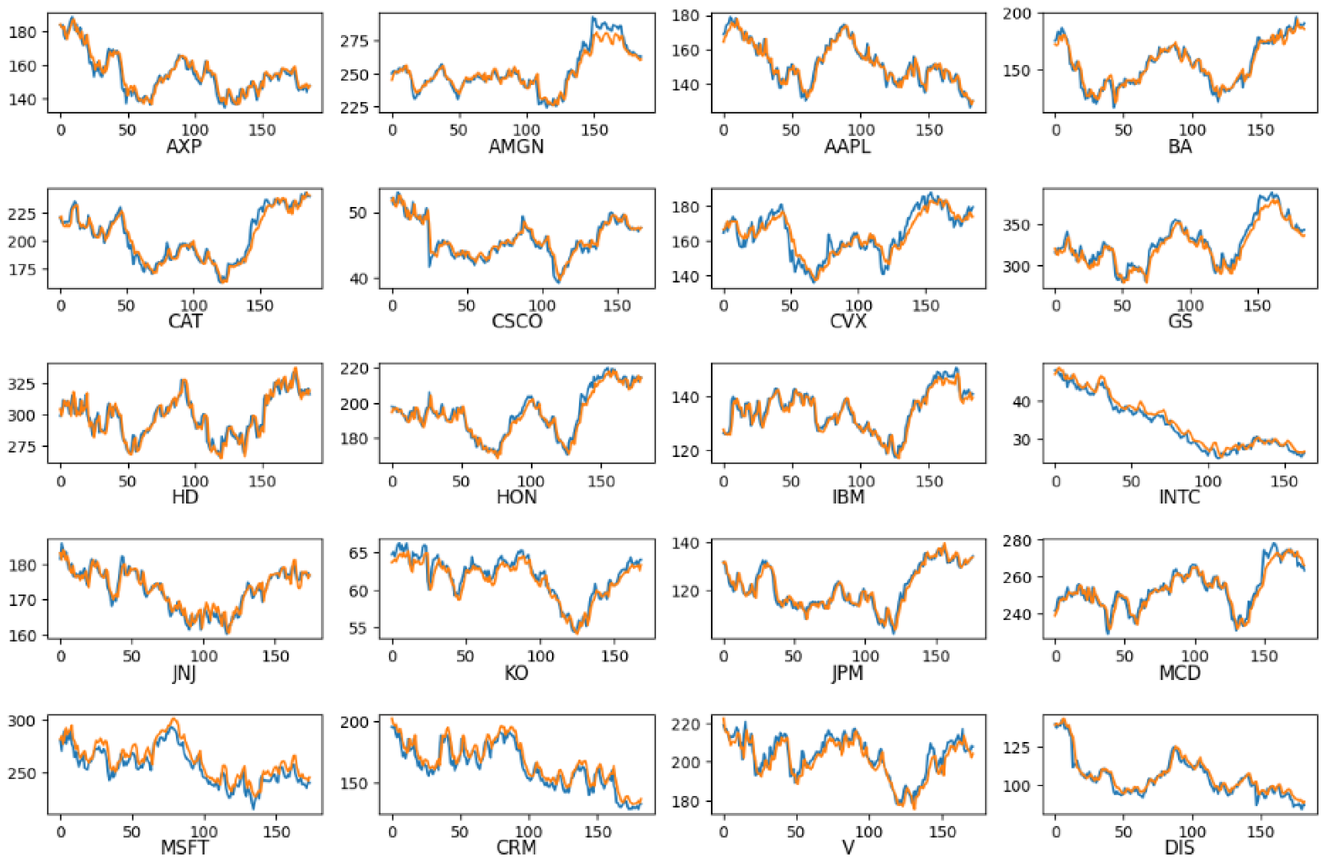


Fig. 4. LSTM-GA Stock Price Prediction Plots.

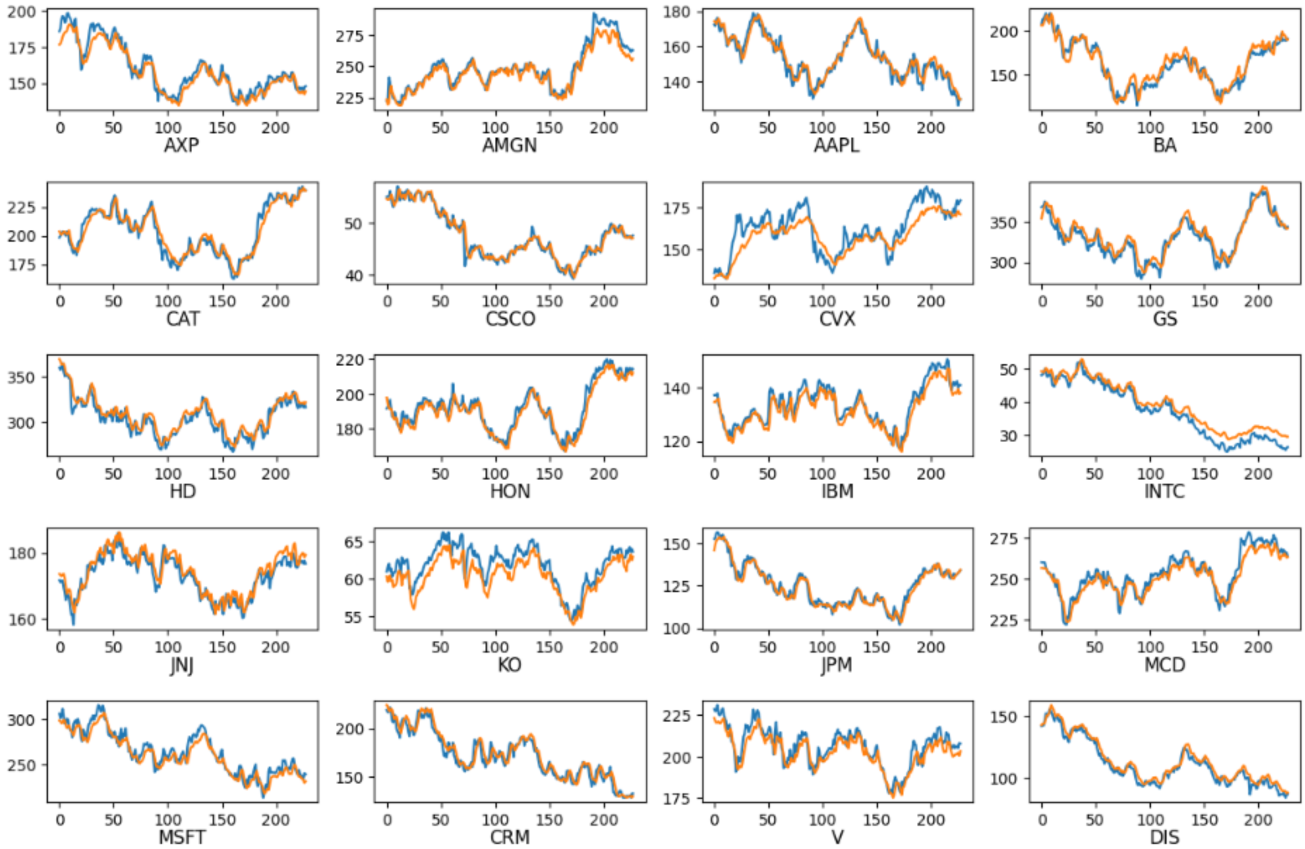


Fig. 5. LSTM-STI Stock Price Prediction Plots.

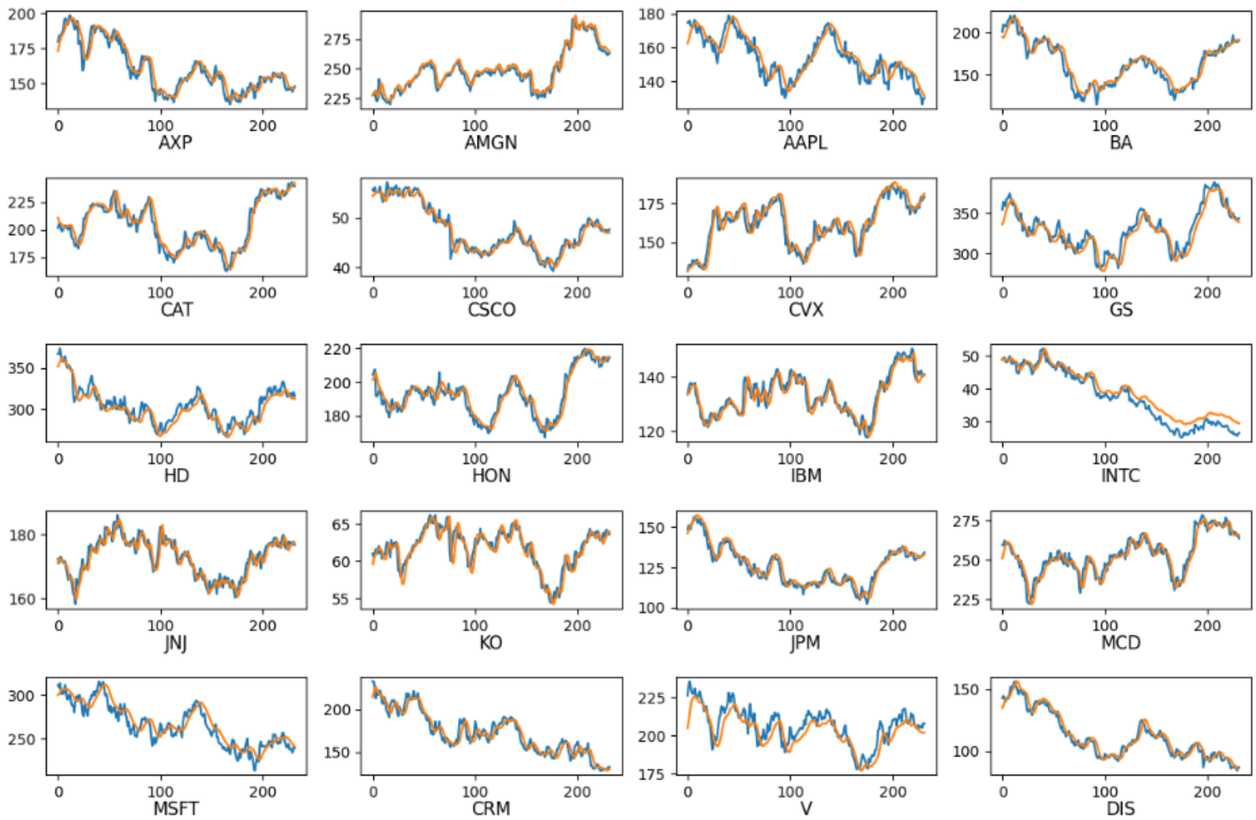


Fig. 6. LSTM Stock Price Prediction Plots.

## 6. CONCLUSIONS

This paper proposed the LSTM-A-SOCLPSO framework, combining Long Short-Term Memory (LSTM) networks with a single-objective optimization algorithm (A-SOCLPSO) to enhance stock price prediction using technical indicators. Evolutionary optimization of hyperparameters improved model performance over benchmark approaches. Through diverse technical indicators and structured preprocessing, the model demonstrated strong forecasting capabilities across multiple DJIA stocks. The key contribution lies in validating hybrid deep learning and evolutionary optimization as a viable strategy for financial time series prediction, offering a foundation for adaptive models that respond to changing market dynamics.

However, several limitations should be acknowledged. The model's performance depends on the quality of historical data and may be affected by unexpected market changes or macroeconomic events. Its reliance on a fixed set of technical indicators and a 60-day look-back window may limit adaptability across different asset types. Additionally, using only closing prices excludes other informative inputs such as sentiment or macroeconomic factors. Future research may explore integrating sentiment analysis, diverse financial data sources, and reinforcement learning to improve adaptability, as well as applying hybrid optimization methods to address convergence challenges. Expanding the framework for real-time forecasting and model interpretability could further enhance its practical relevance in financial applications.

## REFERENCES

Ali, H. and Khan, F.A. (2013) 'Attributed multi-objective comprehensive learning particle swarm optimization for optimal security of networks', *Applied soft computing*, 13(9), pp. 3903–3921.

Baek, Y. and Kim, H.Y. (2018) 'ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module', *Expert Systems with Applications*, 113, pp. 457–480. Available at: https://doi.org/https://doi.org/10.1016/j.eswa.2018.07.019.

Chong, E., Han, C. and Park, F.C. (2017) 'Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies', *Expert Systems with Applications*, 83, pp. 187–205. Available at: https://doi.org/https://doi.org/10.1016/j.eswa.2017.04.030.

Garita, M. and Garita, M. (2021) 'Using Stock Market Data in Python', *Applied Quantitative Finance: Using Python for Financial Analysis*, pp. 71–83.

Gülmez, B. (2023a) 'A novel deep learning model with the Grey Wolf Optimization algorithm for cotton disease detection', *Journal of Universal Computer Science*, 29(6), p. 595.

Gülmez, B. (2023b) 'Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm', *Expert Systems with Applications*, 227, p. 120346. Available at: https://doi.org/https://doi.org/10.1016/j.eswa.2023.120346.

Kumar, K. and Haider, Md.T.U. (2021) 'Enhanced Prediction of Intra-day Stock Market Using Metaheuristic Optimization on RNN–LSTM Network', *New Generation Computing*, 39(1), pp. 231–272. Available at: https://doi.org/10.1007/s00354-020-00104-0.

Liu, Q. *et al.* (2022) 'Stock market prediction with deep learning: The case of China', *Finance Research Letters*, 46, p. 102209. Available at: https://doi.org/https://doi.org/10.1016/j.frl.2021.102209.

Long, W., Lu, Z. and Cui, L. (2019) 'Deep learning-based feature engineering for stock price movement prediction', *Knowledge-Based Systems*, 164, pp. 163–173. Available at: https://doi.org/https://doi.org/10.1016/j.knosys.2018.10.034.

Lu, M. and Xu, X. (2024) 'TRNN: An efficient time-series recurrent neural network for stock price prediction', *Information Sciences*, 657, p. 119951.

Manish Agrawal Piyush Kumar Shukla, R.N.A.N.M.M. (2022) 'Stock Prediction Based on Technical Indicators Using Deep Learning Model', *Computers, Materials & Continua*, 70(1), pp. 287–304. Available at: https://doi.org/10.32604/cmc.2022.014637.

Öztürk, M.M. (2023) 'Initializing hyper-parameter tuning with a metaheuristic-ensemble method: a case study using time-series weather data', *Evolutionary Intelligence*, 16(3), pp. 1019–1031. Available at: https://doi.org/10.1007/s12065-022-00717-y.

Sachdeva, A. *et al.* (2019) 'An Effective Time Series Analysis for Equity Market Prediction Using Deep Learning Model', *2019 International Conference on Data Science and Communication (IconDSC)*, pp. 1–5. Available at: https://api.semanticscholar.org/CorpusID:201812219.

Selvin, S. *et al.* (2017) 'Stock price prediction using LSTM, RNN and CNN-sliding window model', in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1643–1647. Available at: https://doi.org/10.1109/ICACCI.2017.8126078.

Shin, K.-S. and Lee, Y.-J. (2002) 'A genetic algorithm application in bankruptcy prediction modeling', *Expert systems with applications*, 23(3), pp. 321–328.

Srivastava, P. and Mishra, P.K. (2021) 'Stock Market Prediction Using RNN LSTM', in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, pp. 1–5. Available at: https://doi.org/10.1109/GCAT52182.2021.9587540.

Usmani, U.A. *et al.* (2024) 'Deep Learning for Anomaly Detection in Time-Series Data: An Analysis of Techniques, Review of Applications, and Guidelines for Future Research', *IEEE Access* [Preprint].

Wang, A. *et al.* (2024) 'Against predictive optimization: On the legitimacy of decision-making algorithms that optimize predictive accuracy', *ACM Journal on Responsible Computing*, 1(1), pp. 1–45.

Wang, L. *et al.* (2022) 'Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems', *Engineering*

*Applications of Artificial Intelligence*, 114, p. 105082. Available at: https://doi.org/https://doi.org/10.1016/j.engappai.2022.105082.

Xue, G. *et al.* (2022) 'An adaptive neuro-fuzzy system with integrated feature selection and rule extraction for high-dimensional classification problems', *IEEE Transactions on Fuzzy Systems*, 31(7), pp. 2167–2181.

Yahoo! (2024) *Yahoo! Finance*. Available at: https://finance.yahoo.com/ (Accessed: 24 April 2024).

Zhang, Y. and Yang, S. (2019) 'Prediction on the highest price of the stock based on PSO-LSTM neural network', in *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*. IEEE, pp. 1565–1569.

Zhou, X. *et al.* (2018) 'Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets', *Mathematical Problems in Engineering*. Edited by Q. Zhang, 2018, p. 4907423. Available at: https://doi.org/10.1155/2018/4907423.