Baghdad Science Journal

Volume 23 | Issue 5 Article 13

5-26-2026

Predicting Future Stock Closing Prices: A Five-Stage Approach Utilizing Historical Data and Enhanced Deep Learning Techniques With Multi-Strategies Improved Beluga Whale Optimization

Murahari Reddy
Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, TamilNadu, India, dmuraharireddy@gmail.com

R. Balamanigandan
Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, TamilNadu, India, drrbalamanigandan@gmail.com

How to Cite this Article

Reddy, Murahari and Balamanigandan, R. (2026) "Predicting Future Stock Closing Prices: A Five-Stage Approach Utilizing Historical Data and Enhanced Deep Learning Techniques With Multi-Strategies Improved Beluga Whale Optimization," *Baghdad Science Journal*: Vol. 23: Iss. 5, Article 13.
DOI: https://doi.org/10.21123/2411-7986.5297

BAGHDAD SCIENCE JOURNAL 2026;23(5):1711–1738

Scan the QR to view the full-text article on the journal website

RESEARCH ARTICLE

Predicting Future Stock Closing Prices: A Five-Stage Approach Utilizing Historical Data and Enhanced Deep Learning Techniques With Multi-Strategies Improved Beluga Whale Optimization

Murahari Reddy *, **R. Balamanigandan**

Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, TamilNadu, India

ABSTRACT

Predicting stock market movements is a complex and challenging task due to many factors that can influence stock prices. While there are various approaches and techniques that researchers and analysts have explored, it's crucial to note that accurately forecasting stock prices consistently is complicated. Financial data can be noisy and subject to errors. Minor inaccuracies in data can lead to inaccurate predictions. In this research, we propose a new deep learning technique with an optimization algorithm to overcome the limitations in stock market prediction. This research has five stages to predict closing prices using historical data. The input data is initially collected from three datasets (Apple,Microsoft,&Tesla). This data must undergo punctuation removal, outlier removal, replacing missing values, data normalization, and data cleaning performed in the pre-processing stage for accurate prediction. After that, the features are extracted using the Improved ResNeXt approach to improve the prediction performance. Finally,a novel DL technique of EfficientNet V2 is employed to predict the future stock closing prices based on historical data. To improve the prediction accuracy, the study utilizes the Multi-Strategies Improved Beluga Whale Optimization Algorithm (MSIBWOA) to optimize the hyper-parameters. For determining the results of the DL stock forecasting approaches, the Accuracy, Mean Squared Error (MSE), Pearson's Correlation (R), Normalized Root Mean Squared Error (NRMSE), and Root Mean Squared Error (RMSE) metrics were used. The performance of the proposed method is compared with other "state-of-the-art" techniques. In comparison to the most advanced baseline algorithms, the evaluations demonstrate a significant improvement in the prediction's performance.

Keywords: Deep learning, EfficientNet V2, Improved ResNeXt, Optimization, Stock datasets, Stock market prediction

Introduction

One of the foundational elements of a nation's economy is its stock market. Over time, the stock market's performance determines whether a nation's GDP will expand or contract. Because the financial markets are volatile, it is unknown if the investments will generate profits or result in massive investor losses.[1–3] As part of economic liberalization, stock markets play the most essential role in global company financial strategies. The most important question for traders is what should be done concerning a specific stock, i.e., whether to buy, hold, or sell the shares of a stock.[4–6] As a result, such prediction models that help predict stock values more correctly and effectively are required.

These prediction models have been developed using multiple factors that influence stock prices. The pre-

Received 9 April 2025; revised 10 November 2025; accepted 16 November 2025.
Available online 26 May 2026

* Corresponding author.
E-mail addresses: dmuraharireddy@gmail.com (M. Reddy), drrbalamanigandan@gmail.com (R. Balamanigandan).

https://doi.org/10.21123/2411-7986.5297

diction utilizing historical data is the most important of all the other parts. The historical data represents the changing pattern of the rate. Over time, it has been discovered that more than historical data is needed to improve model accuracy.[7–9] In addition to the historical data, sentiment data must be used as an extra factor to enhance the model's efficiency, since sentiment data analyzes current scenarios such as natural disasters, government policy changes, new foreign investments, etc.[10–12]

Shareholders and all stock market investors are interested in stock market forecasts. As a result of their effectiveness, deep learning models have proven to be possible replacements in forecasting stock price research. Linear models with classic machine learning approaches are used for predicting. While ensemble models may deliver more excellent performance,[13–15] deep learning models outperformed typical machine learning models. This research was conducted for successfully predicting stock market closing prices and performing well. So, in this case, this study utilizes an optimized deep learning technique to predict the stock market to get higher prediction accuracy in less time.

Novelty of the proposed research

- Compile historical stock price information from three well-known datasets (Microsoft, Tesla, and Apple). Strong predictive modeling that takes into account various stock behaviors and market conditions is made possible by this diverse dataset, which guarantees thorough market coverage.
- Use a number of preprocessing methods such as data normalization, data cleaning, outlier removal, punctuation removal, and missing value replacement. The quality and dependability of the dataset are improved by cleaning and preparing the data, which is essential for precise forecasts. This stage reduces data noise and biases, producing more reliable modeling outcomes.
- To extract useful features from the preprocessed data, apply the Improved ResNeXt method. This step improves the model's prediction performance by utilizing a cutting-edge feature extraction technique to identify intricate patterns and relationships in the data that conventional methods might overlook.
- Use the cutting-edge deep learning method EfficientNet V2 to forecast future stock closing prices. EfficientNet V2 allows the model to produce high-quality predictions without consuming excessive amounts of resources by optimizing accuracy while maintaining computational efficiency.
- To optimize the hyper-parameters of the EfficientNet V2 model, use the Multi-Strategies Improved Beluga Whale Optimization Algorithm (MSIBWOA). By methodically looking for the ideal hyper-parameter configurations, this sophisticated optimization algorithm improves the model's predictive accuracy and enhance stock price forecasting performance.

Related prior works

Very few studies have been conducted in the stock market prediction field. This section highlights some earlier studies based on stock market forecasts for various corporations here. Dael et al.[16] proposed "multi-model generative adversarial network hybrid prediction algorithm (MMGAN-HPA) for stock market price prediction." They developed a prediction model based on the GAN approach. They gathered the data, pre-processed it, retrieved its attributes, evaluated the system, and then used the GAN approach to create a stock price forecasting system. The GAN comprises a discriminator and a generator, which are both trained via adversarial learning procedures. Therefore, they used attributes such as open, high, date, close, volume, and low to train this system.

Aldhyani and Alzahrani's[17] study "Framework for predicting and modelling stock market prices based on deep learning algorithms" was suggested in this research. They suggested employing hybrid techniques to anticipate stock market closing prices. First, they gathered information from Tesla and Apple's records. When the amount of data being used was significant, normalization was an essential approach for information scaling; therefore, it falls into a precise interval. Normalization speeds up and improves the precision of gradient descent—normalization from minimum to maximum. The stock market closing prices were then predicted using a combination of LSTM and CNN approaches.

The paper "Stock Prediction Based on the Bidirectional Gated Recurrent Unit with Convolutional Neural Network and Feature Selection" was suggested by Zhou et al.[18] They proposed a hybrid convolutional recurrent model framework based on a convolutional recurrent (CNN-GRU) hybrid network for the stock-price forecasting model. The stock price dataset was chosen and prepared: Pre-process the data, divide it into testing and training sections, and convert input into supervised training information using the sliding window method with timestamps. The hybrid model design included parameter settings with recurrent and convolutional layers, and input of pre-processed information.

Yang et al. [19] presented this work, "Forecasting Apple Stock Closed Prices by LR and LSTM with Discrete Wavelet Transformation." This work suggested a hybrid forecasting model combining the advantages of denoising techniques and ML methods for forecasting Apple stock closing prices utilizing the closed prices five days ago to aid investors in predicting the future trend of stock prices. The attributes chosen were Apple stock closing prices. After min-max rescaling, all of the features' values are between 0 and 1 and hybrid LSTM and LR methods were trained using the denoised information and the acquired features.

Swathi et al.'s [20] paper "An optimal deep learning-based LSTM for stock price prediction using Twitter sentiment analysis" demonstrated a Teaching and Learning Optimization (TLBO) model based on LSTM sentiment analysis for stock price forecasting utilizing Twitter historical information. Because tweets are often short and have odd grammatical frameworks, stock data pre-processing is needed to remove irrelevant material. Furthermore, the LSTM method categorizes tweets into negative and positive views about stock costs. The Adam optimizer was utilized to select the learning rate to improve the predicted output of the LSTM model.

The above literature shows specific issues to be resolved in the proposed work. Stock price movements are typically complicated. Traders have long relied on their ability to forecast changes in stock values. It also took longer to complete, but the prediction accuracy was good. Therefore, to solve the flaws identified in prior studies, this study executes research in stock market prediction. It utilizes a feature selection technique to reduce the time complexity of the evaluation performance. A novel deep learning technique of EfficientNet V2 with an optimization algorithm was employed to get higher prediction accuracy for stock market prediction. This system analyzes the prediction of the closing costs of three companies (Apple, Microsoft, and Tesla).

Materials and methods

Stock market prediction is an exciting domain for traders to predict future prices. Previous years' research on stock market prediction has some issues with predicting accurately. So, to overcome the limitations, this study proposes a DL approach for predicting stock prices. First, collect the historical data from the three companies' datasets. Then, punctuation removal, outlier removal, replacing missing values, data normalization, and data cleaning are performed in the pre-processing stage. Fig. 1 demonstrates the structure of the proposed methodology.

Then, the Improved ResNeXt technique is used to extract all features from the pre-processed data. Then, EfficientNet V2 is employed to predict future stock closing prices with a Multi-Strategies Improved Beluga Whale Optimization Algorithm (MSIBWOA) to improve prediction accuracy by optimizing the proposed model's hyper-parameters. The evaluation process is performed on three datasets.

Pre-processing

This research focuses on predicting stock market trends for three major companies: Microsoft, Apple, and Tesla. An accurate prediction in such a volatile domain requires careful data handling, ensuring it is well-prepared. To do this, pre-processing is essential. To improve the stock data for prediction, a number of critical procedures are used:

Punctuation removal

When processing text data related to the stock market, such as news articles, financial reports, or social media posts, punctuation removal is crucial even though it is usually not a priority in numerical datasets like stock prices. Sentiment analysis and event-based insights from text data can have a big impact on stock prices. It can remove all extraneous punctuation symbols and special characters from textual elements. To guarantee cleaner data for any text-based models used in the prediction items such as news headlines and company reports are eliminated.

Outlier removal

Rare occurrences like abrupt market crashes or unanticipated surges can result in outliers in stock market data. Outliers can occasionally indicate significant market events, but if they are not managed properly, they can also skew forecasts. Predictions can become more stable when outliers are eliminated or adjusted to lessen dataset volatility. To identify and eliminate extreme outliers, statistical methods like the Interquartile Range (IQR) or Z-scores are used. Careful consideration is given to stock prices that show abrupt increases or decreases, in order to prevent the exclusion of crucial information that could represent real market behavior.

Replacing missing values

Market holidays, trading halts, or problems with data collection can all result in missing data in stock market datasets. Effective handling of missing values

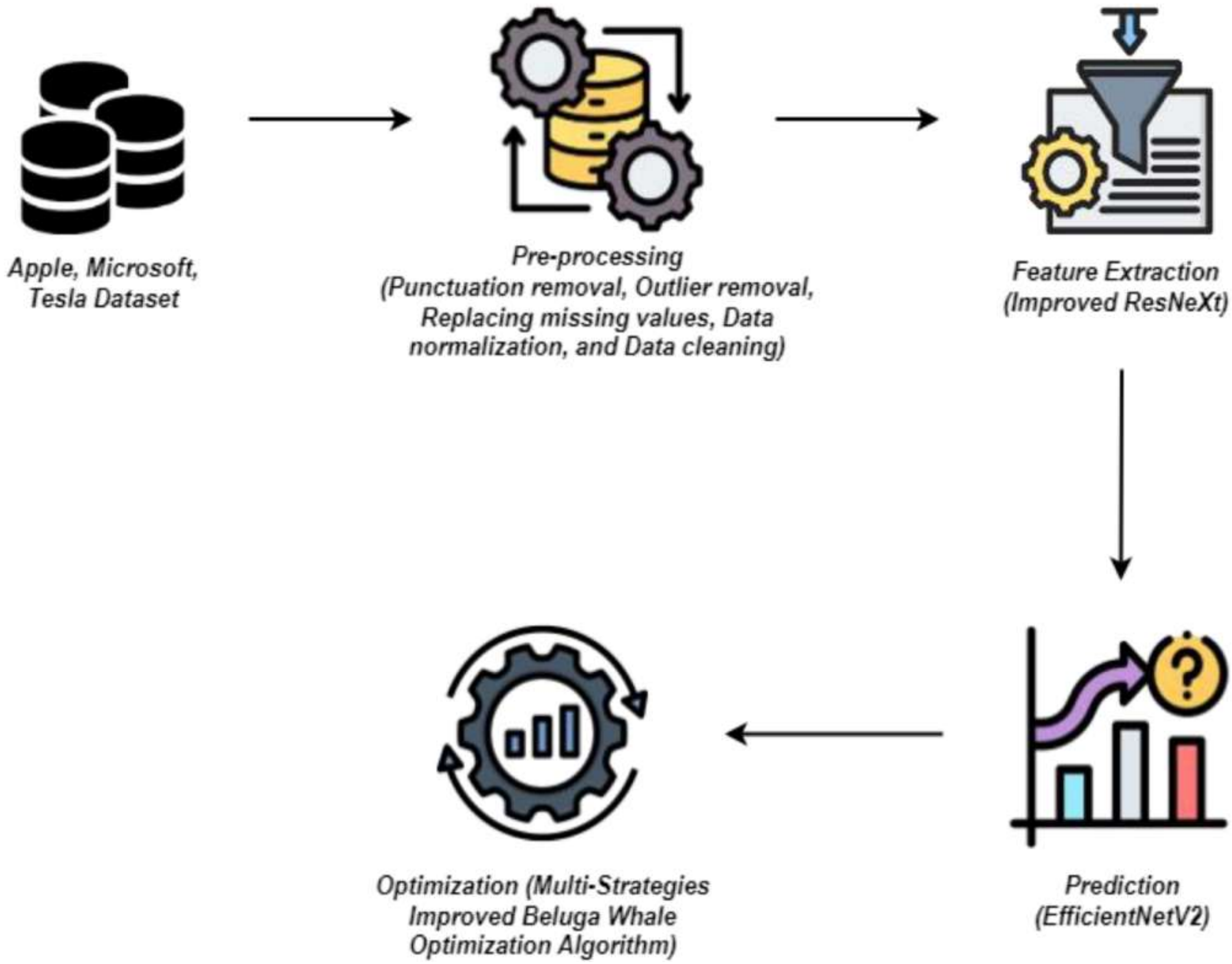


Fig. 1. The proposed methodology structure.

is essential for predictive models to function well. Therefore, attributing or replacing missing values is a critical step. In order to ensure continuous and consistent data for training models, imputation techniques are used to replace missing data in stock prices. These techniques include filling in missing values with the price of the previous day (forward fill) or using mean, median, or interpolation methods.

Data normalization

Prices of stocks for various companies (such as Tesla, Apple, and Microsoft) can differ greatly in terms of volatility and price scales. By allowing each feature to contribute equally, data normalization makes sure that these variations do not have a detrimental effect on the predictive model. Stock prices are scaled to a standard range (usually 0 to 1) using Min-Max normalization.[21] Neural networks and other models that are sensitive to scale benefit from this step because it keeps larger numbers from dominating smaller ones during training.

Data cleaning

The raw stock data must be cleaned before any meaningful prediction can be made. This step involves removing redundant, irrelevant, or incorrect entries, ensuring the data is accurate and suitable for analysis. Duplicate records are removed, irrelevant features are dropped, and errors in stock data entries (such as incorrect stock prices) are corrected. Fig. 2 depicts the normalized data for Apple, Microsoft, and Tesla.

The dataset is then restructured to a consistent format and made ready for analysis.

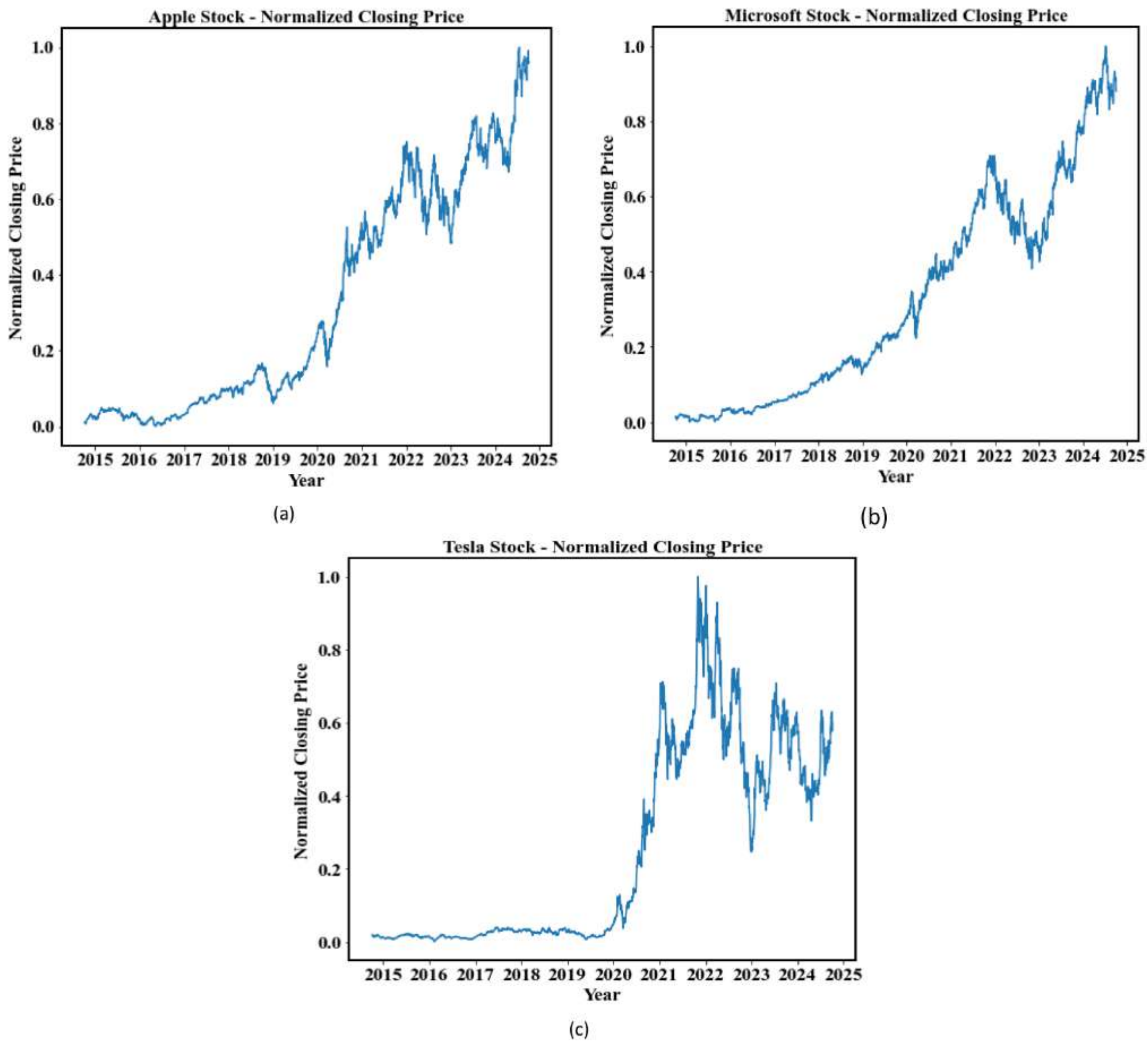


Fig. 2. Normalized closing prices of (a) Apple, (b) Microsoft, and (c) Tesla.

Feature extraction

After performing the pre-processing process on input stock history data, attributes are extracted from the pre-processed data. To extract the attributes For that, the Improved ResNeXt technique is used. ResNeXt is an upgraded version of ResNet. The ResNeXt network includes one fully connected layer, one convolutional layer, two pooling layers, four ResNeXt block structures, and one SoftMax classifier.[22] ResNeXt blocks are made up of deep residual networks of cardinality. They use the split-transform-merge approach to change the residual block, resulting in branching paths within a cell. The skip connection path delivers the ResNeXt block's result. The depth of the residual networks expands orthogonally, and the ResNeXt block's cardinality is dictated by the number of branching paths it contains. The ResNeXt block can be mathematically denoted as in the following Eq. (1):

$$\mathrm{OP} = \mathrm{a} + \sum_{\mathrm{i}=1}^{\mathrm{ca}} \tau_{\mathrm{i}}\,(\mathrm{a}) \tag{1}$$

Convolutional layers are classified into three types: convolution in sequence, group convolution, and convolution. The input features provided by the shortcut are mixed with the result of the final convolution layer, which is then activated utilizing the ReLU activation function. The set of convolution strides for the ResNeXt Block1 phase is 2.

In the pooling layer, max pooling is utilized to achieve spatial invariance and accelerate training while maintaining accuracy. During the pooling process, the most significant value of the neighborhood is determined by Eq. (2):

$$v_{i,L}^{x,y} = \max_{m \in [0, m_i - 1], n \in [0, n_i - 1]} v_{(i-1),L}^{(x+m),(y+n)} \tag{2}$$

The feature map of the residual learning module is converted into a 128-dimensional feature vector and transferred to the fully connected layer through global average pooling. The average pooling layer is utilized to retrieve the features from the dataset. Because of their broader and deeper architectures than typical convolutional neural networks (CNNs), improved ResNeXt models are recognized for their improved ability to learn complex patterns and representations from data.

Recognizing insignificant yet significant changes in historical data is critical for stock market prediction activities since it can significantly increase forecasting accuracy. By utilizing Improved ResNeXt, complicated, high-dimensional data from financial time series may be handled more skillfully, effectively capturing both short- and long-term trends and fluctuations. Furthermore, the model is a strong option for extracting significant features that support precise stock market predictions because of its multi-path architecture's capacity to use comprehensive feature learning techniques, which improves the model's generalization across various market situations. The heatmap of features is shown in Fig. 3.

Prediction

To predict the future stock closing prices in the stock market, EfficientNetV2 is utilized in this framework. EfficientNetV2, an upgraded version of EfficientNetV1, is an additional family of convolutional neural networks with a particular emphasis on improving the speed of training and enhancing parameter efficiency.[23] A mix of training-aware searches for neural structures and compound scaling were utilized to achieve this goal.

The EfficientNetV2 networks are made up of seven blocks. All blocks in EfficientNetV1 are MBConv. In the first three blocks, MBConv is substituted with Fused-MBConv to improve training time and reduce model complexity. The next three blocks include MBConv blocks. Finally, block seven comprises a regular 1×1 convolution layer, a medium pooling layer, and an entire connection layer. Eq. (3) and Eq. (4) show the convolutional layer equations.

$$z^l = a^{l-1} * W^l + b^l \tag{3}$$

$$a^l = f^l \left(z^l \right) \tag{4}$$

where z^l is the result of the l^{th} convolution, is the convolution operation, W^l represents the weight of the l^{th} layer, b^l represents the offset of the l^{th} layer, a^l represents the outcome of the l^{th} layer, and f^l demonstrates the activation function of the l^{th} layer.

The SE module comprises two fully connected layers and a global average pool. To improve feature representations, the SE block incorporates an attention technique. The network employs the h-swish function as an activation function rather than ReLU since it discards values less than zero, potentially resulting in the loss of critical information. The following Eq. (5) and Eq. (6) are the description of the h-Swish:

$$h - swish\,(x) = x.\sigma\,(x) \tag{5}$$

$$\sigma\,(x) = \frac{ReLU6\,(x + 3)}{6} \tag{6}$$

Here $\sigma(x)$ represents the piecewise linear complex imitation function. This network fused-MBconv block is another essential element. As the input data passes through the layers and blocks of the network, it undergoes transformation and convolution. Feature maps, which show the input data at different levels of abstraction, are produced by these transformations. Because EfficientNetV2 strikes a balance between performance and computational efficiency, it provides a number of benefits for stock market prediction. Compared to conventional models, it can achieve greater accuracy with fewer parameters by using a compound scaling technique that uniformly scales depth, width, and resolution. Because of this, it can process big financial datasets with quicker training and inference times. Prediction accuracy in volatile markets is enhanced by its capacity to capture intricate patterns using attention mechanisms and optimized convolutional layers. Additionally, real-time analysis and forecasts in stock market applications are made possible by the lightweight architecture, which lowers computational costs.

Optimization

The BWO algorithm was inspired by the beluga whale, which is skilled at hunting and navigating in

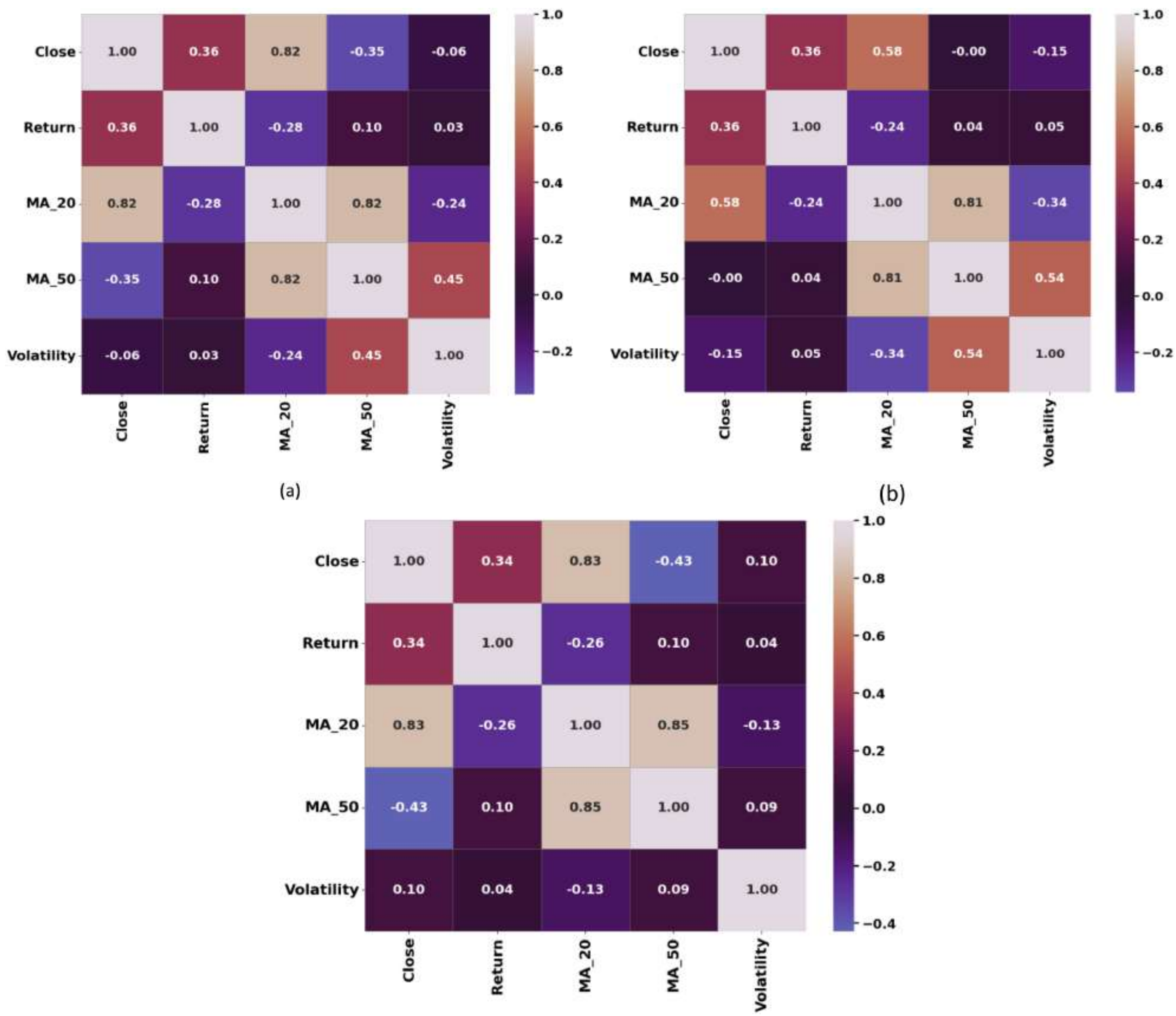


Fig. 3. Heatmap of features of (a) Apple, (b) Microsoft, and (c) Tesla.

aquatic habitats. This new biomimetic optimization technique solves optimization problems by modelling beluga whales' social interactions and hunting techniques. In BWO, the exploration stage represents the algorithm's worldwide search for the best solutions and is comparable to how belugas utilize echolocation to find and track prey. On the other hand, after a target has been found, the exploitation phase represents the algorithm's refined search in a potentially productive area. It is similar to the targeted hunting strategies utilized by these whales. Another feature of the whale fall phase is introducing a novel mechanism replicating the natural phenomena of a whale falling to the ocean floor upon its death. This mechanism functions as an ecological disturbance, potentially opening up new search spaces and allowing escape from local optima.

Initialization

Beluga whale sites are considered search agents. The initial population is created randomly during the initialization stage, and every individual's fitness value is calculated. The following Eq. (7) is the population initialization model:

$$X = \mathrm{rand}\,(m, n) \cdot \left(Ub - Lb\right) + Lb \tag{7}$$

Like the Harris Hawks Optimizer (HHO), BWO moves from exploration to exploitation according to the balancing factor B_f in every iteration. This is the model for the balance factor B_f calculated by Eq. (8):

$$B_f = B_0 \left(\frac{1 - T}{2T_{max}}\right) \tag{8}$$

T_{max} is the best iterative number, B_0 is a random number between 0 and 1, and T is the current iteration. Beluga whale pairs perform synchronized or mirrored swimming when the balancing factor $B_f > 0.5$ indicates they are exploring. The exploitation phase begins when $B_f \leq 0.5$ initiates in beluga whale predatory behaviour.

Exploration phase

Beluga whale pair swimming is replicated in the BWO exploration phase. They travel in an unpredictable, mirrored, or synchronized manner, described as in the following Eq. (9):

$$\begin{cases} X_{t,f}^{T+1} = X_{t,p_f}^{T} + \left(X_{t,p_1}^{T} - X_{t,p_f}^{T}\right) \\ \qquad \times (1 + r_1)\sin(2\pi r_2), \quad j = \text{even} \\ X_{t,f}^{T+1} = X_{t,p_f}^{T} + \left(X_{t,p_1}^{T} - X_{t,p_f}^{T}\right) \\ \qquad \times (1 + r_1)\cos(2\pi r_2), \quad j = \text{even} \end{cases} \tag{9}$$

Exploitation phase

The Beluga Whale Organization (BWO) exploitation phase imitates beluga whale preying behaviour. To forage and move together by the whereabouts of beluga whales in the area, beluga populations exchange signals and information about their placements. In the exploitation phase, the LF approach improves the algorithm's convergence. This can be expressed as in Eq. (10):

$$X_i^{T+1} = r_3 X_{best}^{T} - r_4 X_i^{T} + C_1 \times L_F \cdot \left(X_r^{T} - X_i^{T}\right) \tag{10}$$

Where r_3 and r_4 are randomly selected numbers between (0, 1), and $C_1 = 2r_4(1 - T/Tmax)$ is the random leap strength. The LF function, or L_F, is computed in this way as Eqs. (11) and (12):

$$L_F = 0.05 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{11}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1/2)}}\right)^{1/\beta} \tag{12}$$

Where the random variables u and v are normally distributed, the Γ function is represented by $\Gamma(x)$, and B is the default constant set to 1.5.

Whale fall phase

Whale fall happens if the balancing factor gives the chance of a single beluga whale falling. During migration and foraging, beluga whales are threatened by killer whales and humans, an occurrence known as the whale fall. The deceased whale plummets to the ocean's depths. After that, the updated location is determined using the current beluga whale's location, random beluga whale locations, and the whale fall's step size to maintain the population size, formulated as Eq. (13):

$$X_i^{T+1} = r_5 X_i^{T} - r_6 X_i^{T} + r_7 X_{step} \tag{13}$$

Where the random values r_5, r_6, and r_7 fall between 0 and 1, and X_{step} represents the whale fall's step size, which is determined by Eq. (14):

$$X_{step} = \left(Ub - Lb\right) e^{-C_2 T/T_{max}} \tag{14}$$

C_2 is the step factor, and $C_2 = 2Wf \times m$ is a function of population size and the likelihood of a whale falling.

Proposed multi-strategies improved beluga whale optimization algorithm

The proposed MSBWO introduces four successful strategies: the EP mechanism, the SLFSUP, the ICMDOBL population diversity strategy, and the Gold-SA population update mechanism. Additionally, upgrading MSBWO to a binary version can resolve the feature selection issue.

ICMDOBL population diversity

The BWO algorithm's random population creation may result in an unequal distribution of the population, which could impair population quality and variety and hinder the algorithm's convergence. The features of chaotic mapping include unpredictability, irreducibility, and uncertainty. This approach may provide a more homogeneous population distribution compared to probability-dependent random generation. With chaotic mapping, MSBWO creates an initial population that broadens the range of possible answers as in the following Eq. (15).

$$X_{i+1,j} = \mathrm{mod}\left(3.85X_{i,j} + 0.4 - \frac{0.7}{3.85\pi} \times \sin(3.85\pi X_{i,j}), 1\right) \tag{15}$$

Next, the values undergo scaling and shifting in order to produce an X_{Circle} for each dimension, with values ranging from Lb to Ub as Eq. (16):

$$X_{Circle} = \left(Ub - Lb\right) \cdot X_{i,j} + Lb \tag{16}$$

To enhance the quality of the initial solutions and diversify the population even more, DOBL is implemented. Here is how the particular solution is stated:

$$X_{DOBL} = X_{Circle} + r_8 \times \left(r_9 \times (L_b + U_b - X_{Circle}) - X_{Circle}\right) \tag{17}$$

Where r_8 and r_9 are random values between 0 and 1, and X_{Circle} is the population established using the ICM approach, as illustrated in Eq. (17). The DOBL creates the opposition population X_{DOBL} and the generated population X_{Circle}. These two populations are then combined to form $X_{new} = \{X_{DOBI} \cup X_{Circle}\}$. The first population is then chosen from the top N individuals. MSBWO improves convergence by starting to iterate with the fittest individuals using ICMDOBL.

EP strategy

Regarding updating their location, beluga whales usually take the best whale for prey. A hierarchical approach was proposed for the GWO algorithm to update the locations based on the mean area of the top three best grey wolves to overcome the drawbacks of using a single search agent. The EP approach is incorporated into MSBWO and modelled after GWO. The EP contains the candidate elites, the top three solutions, and their weighted average. The algorithm's ability to break out from local optima is enhanced by the agent randomly chosen from the EP to direct position updates.

The EP strategy is modelled as Eq. (18):

$$\begin{cases} Elite^T = \left[X^T_{best\_1}, X^T_{best\_2}, X^T_{best\_3}, X^T_{mean}\right] \\ X^T_{mean} = \sum\limits_{i-1}^{best\_num} \theta_i X^T_{best\_i}, \ \theta_i = \frac{\omega_i}{\sum_{j-1}^{best\_num} \omega_j} \\ \omega_i = \frac{3f_{best\_3} - f_{best\_i}}{3f_{best\_3} - f_{best\_1}} \end{cases} \tag{18}$$

Where best_num is set to 3, and f_{best_i} is the fitness value.

SLFSUP strategy

BWO employs LF with a fixed step throughout the exploitation stage to enhance its convergence. However, the expected step of LF may change depending on where the algorithm is. Finding the best result is more accessible with more significant LF steps. Consequently, MSBWO employs the step-adaptive LF approach to increase its convergence accuracy and exploitation. To fully utilize the solution space, MSBWO employs LF with a more significant step in the early phases of the iteration as the following Eq. (19).

$$L'_F = 0.05 \times (1 - T/(T_{max}/2)) \times e^{-T/T_{max}} \cdot \frac{u \times \sigma}{|v|^{1/\beta}} \tag{19}$$

A random agent, the current agent, and the best agent are involved in location update in the BWO exploitation, as indicated by Eq. (20). More details need to be included in the potential fix. When seeking prey, the whale's position adjusts the distance using the spiral updating location method in WOA, which is based on the prey's location and the best solution found. This approach can enhance search skills and fully utilize the local data. MSBWO presents this technique to improve the algorithm's precision and rigour in regional space development and to fortify local search capability.

$$X_i^{T+1} = r_3 X^T_{EP} - r_4 X_i^T + C_1 \times L'_F \cdot \left(X_r^T - X_i^T\right) \times \left|X^T_{EP} - X_i^T\right| \times \cos(2\pi l) \times e^{bl} \tag{20}$$

Where X_r is its current location of a randomly picked beluga whale to preserve its variety. b in MSBWO is set to 1.

Golden-SA update mechanism

The Gold-SA scans all sine function values, drawing inspiration from the connection between a sine function and a circle with a unit radius. The algorithm is quite good at searching globally. Gold-SA uses the golden section ratio in its location updating process to accelerate convergence and avoid local optima by thoroughly scanning the search space.

The following Eq. (21) is the updated position with Gold-SA:

$$X_i^{T+1} = X_i^T|\sin(r_{10})| + r_{11}\sin(r_{10})|x_1 \times X^T_{EP} - x_2 \times X_i^T| \tag{21}$$

Where the random numbers r_{10} and r_{11} are in the intervals $[0, 2\Pi]$ and $[0, \Pi]$, respectively. The following Eq. (22) is an expression for them:

$$\begin{cases} x_1 = a\tau + b(1 - \tau) \\ x_2 = a(1 - \tau) + b \end{cases} \tag{22}$$

where $\tau = (\sqrt{5}-1)/2$ is the golden amount and a and b have beginning values of $-\Pi$ and Π, respectively.

Binary MSBWO

The proposed MSBWO has a far better search performance than the original BWO because it is an upgraded algorithm. Consequently, it is used in this work to produce a better feature subset. However, the agents' search space must be limited when applying MSBWO. Furthermore, binary transformation is necessary to transfer the continuous values to their equivalent binary values.

The initialization is done using Eqs. (23) and (24) to address the issues above:

$$X_i = \left[X_i^f\right],\ 1 \le i \le m,\ 1 \le j \le d \tag{23}$$

$$X_i^f = \begin{cases} 0, & if\ rand < 0.5 \\ 1, & if\ rand \ge 0.5 \end{cases},\ 1 \le i \le m,\ 1 \le j \le d \tag{24}$$

Discretization is carried out using the sigmoid function after position update as Eqs. (25) and (26).

$$S\left(X_i^j(t)\right) = \frac{1}{1+\exp\left(X_i^j(t)\right)} \tag{25}$$

$$X_i^j(t+1) = \begin{cases} 0, & if\ rand < S\left(X_i^j(t+1)\right) \\ 1, & if\ rand \ge S\left(X_i^j(t+1)\right) \end{cases} \tag{26}$$

The FS has two primary objectives as a combination optimization. Enhancing the performance of categorization is one goal; reducing the number of selected attributes is the other. Eq. (27) thus displays the fitness function.

$$\text{Fitness} = r_{12}E_R(D) + r_{13}|S|/|D| \tag{27}$$

Where D is the number of attributes in the original dataset, and $E_R(D)$ is the error rate for the prediction.

Computational complexity

To improve comprehension of the MSBWO algorithm's implementation process as presented in this research, the computational complexity of MSBWO is examined in the following manner. The three primary phases that determine the computing complexity of the MSBWO are the beluga whale's initiation, fitness assessment, and updating. During the MSBWO initialization phase, it is assumed that each agent's computational complexity is (d), where d is the problem value. Considering n as the population size, the computational complexity of ICMDOBL is ($n\times d$). After the repetition, ($n\times logn+n$) is the computing complexity of EP. During the discovery and extraction stages, the innovative exploitation method takes the role of the initial exploration method, and the computing complexity is comparable to BWO, denoted as ($n\times d\times Tmax$). Similar to BWO, the whale fall stage computation can also be approximated as ($0.1\times n\times Tmax$).

Furthermore, another searching approach is Gold-SA, whose computational complexity is ($n\times d$). As a result, the approximate evaluation of MSBWO's overall computational complexity is ($n\times d+n\times Tmax\times(1.1+logn+2\times d)$). As a result, compared to the original BWO technique, the MSBWO approach presented in this study has more computational complexity.

Using the MSIBWOA for hyper-parameter optimization in stock market prediction offers several advantages. It enhances its capacity to navigate intricate high-dimensional hyper-parameter spaces and steer clear of local optima by combining several search strategies to strike a balance between exploration and exploitation. As a result, crucial parameters like learning rate, batch size, and network architecture are tuned more precisely. Because of its ability to adapt to dynamic environments, MSIBWOA is especially useful for stock market prediction where data is extremely volatile and patterns are frequently non-linear, increasing model performance and prediction accuracy.

Results and discussion

In the first half of this section, we forecast stock market prices using the datasets analysis and procedure while comparing our methodology to state-of-the-art techniques.

Hyper-parameter configuration

Hyper-parameters of the model, including learning rate, batch size, and dropout, are obtained in a methodical manner, usually with optimization algorithms like MSIBWOA. These techniques aid in determining the ideal hyper-parameter combination for the best possible model performance. The optimization dynamics unique to this model support the selection of a learning rate of 0. 1 even though it may seem high in comparison to more common values like 1e-3 or 1e-4. When combined with learning rate scheduling or warm-up techniques to avoid overshooting during gradient descent, a higher learning rate can occasionally speed up convergence in the early stages of training or work well with specific architectures like Improved ResNeXt and EfficientNet V2. To guarantee stability and improve performance in the stock market prediction task, this method was empirically validated.

Experimental setup

The experiment were conducted on a device running windows 10 equipped with 4 GB RAM and an Intel i5 2.60 GHz processor. In the Anaconda3 environment, the investigations are carried out using Python, KERAS, and TensorFlow. This study uses the Microsoft stock,[24] Apple, Inc.,[25] and Tesla stock[26] datasets to evaluate the efficacy of our suggested methodology. To accelerate the training of deep learning models, including EfficientNet V2 and feature extraction with Improved ResNeXt, the experiments are performed using an NVIDIA GeForce RTX 3060 GPU with CUDA support.

Dataset description

Microsoft, Inc. data

The daily price history of Microsoft Stock (MSFT) is provided in this dataset. There are descriptions for every column. USD is the currency. From March 13, 1986 to 2024, the Microsoft Historical Dataset provides all of the lifetime stock data. Seven columns make up this dataset: dates, opening, high, low, closing, adj_close, and volume.

Apple, Inc. dataset

A worldwide innovator, Apple, Inc. creates, manufactures, and markets various electronic products, including wearable technologies, smartphones, tablets, and laptops. The Mac series of laptops, desktop computers, the iPhone, the iPad, the apple TV, and the Watch are just a few of the company's well-known and essential product lines.

Tesla, Inc. data

Tesla, Inc. is an American business headquartered in Palo Alto, California, focusing on electric automobiles and renewable energy. The historical information on Tesla Inc. is provided in this dataset. Every day, one can get the information. The currency is the USD. The dataset was gathered between August 4, 2014 and 2024. Table 1 demonstrates every dataset's validation, training, and testing variables.

Table 1. Validation, training, and testing variables for every dataset.

Datasets	Training	validation	Testing
Apple, Inc. dataset	80%	10%	10%
Microsoft stock dataset	80%	10%	10%
Tesla stock dataset	80%	10%	10%

Performance measures

For testing, Optimized EfficientNet V2 and EfficientNet V2 method's performances are determined by using the Accuracy, R^2, RMSE, NRMSE, R%, and MSE metrics by Eqs. (28) to (33):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{28}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_{i,\text{exp}} - y_{i,\text{pred}}\right)^2 \tag{29}$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n}\frac{\left(y_{i,\text{exp}} - y_{i,\text{pred}}\right)^2}{n}} \tag{30}$$

$$R\% = \frac{n\left(\sum_{i=1}^{n} y_{i,\text{exp}} \times y_{i,\text{pred}}\right) - \left(\sum_{i=1}^{n} y_{i,\text{exp}}\right)\left(\sum_{i=1}^{n} y_{i,\text{pred}}\right)}{\sqrt{\left[n\sum_{i=1}^{n}(y_{i,\text{exp}})^2 - \left(\sum_{i=1}^{n} y_{i,\text{exp}}\right)^2\right] \times \left[n\sum_{i=1}^{n}(y_{i,\text{pred}})^2 - \left(\sum_{i=1}^{n} y_{i,\text{pred}}\right)^2\right]}} \times 100 \tag{31}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_{i,\text{exp}} - y_{i,\text{pred}}\right)^2}{\sum_{i=1}^{n}\left(y_{i,\text{exp}} - y_{\text{avg},\text{pred}}\right)^2} \tag{32}$$

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(y_{i,\text{exp}} - y_{i,\text{pred}}\right)^2}}{y_{i,\text{pred}}} \tag{33}$$

Table 2. Results of DL for forecasting the closing cost of Apple for training.

Method	NRMSE	RMSE	MSE	R
Optimized EfficientNetV2	0.07294	0.02312	0.00082	99.88%
EfficientNetV2	0.02108	0.01021	0.00107	98.89%

Since accuracy and error rates are crucial for decision-making in real-world applications like stock market prediction, the selected error rate was chosen in place of the more popular loss metric in deep learning systems because it offers a more straightforward and comprehensible measure of the models prediction performance. The error rate makes it evident how well the model generalizes to new data, even though loss functions are essential for fine-tuning model parameters during training. Furthermore, because it represents the true percentage of inaccurate predictions, the error rate can be more significant when comparing models or evaluating real-world results, which is more in line with the goals of financial forecasting.

Performance metrics

The accuracy, R^2, NRMSE, RMSE, and MSE values were calculated throughout the tests using the difference between the accurate stock closing price and the forecasted stock closing cost. A comparison of the difference between that day's actual stock cost and the predicted stock cost is done to determine the degree of alteration or variation in the stock price on the predicted day. This calculation aims to quantify the discrepancy between the predicted and actual values. Here, this study processes the stock data of Apple, Microsoft, and Tesla.

Results of Apple stock data

Table 2 displays Apple's closing price forecasts using the EfficientNetV2 and Optimized EfficientNetV2 methods. The results show how reliable and robust the DL technique is for predicting closing costs based on training. Optimized EfficientNetV2 had a low forecasting error rate based on the measurements of NRMSE (0.07294) and RMSE (0.02312).

Fig. 4 depicts the precise combination of the forecasted data for Apple's historical stock amount and the actual amounts and R% during the training stage of the stock market dataset. This is demonstrated by the fact that there is no appreciable variation between the actual and displayed figures.

Due to very high R values (98.89% and 99.88%, respectively) and extremely low MSE and RMSE values, EfficientNetV2 and Optimized EfficientNetV2 were eligible to be tested during the training phase. Fig. 5 demonstrates the error in histogram graphs.

After training, a balance of 30% of the collected information is employed to test the predictions made by the DL techniques, as indicated in Table 3. Additionally, the testing stage is essential for evaluating and identifying the advantages of the proposed approaches to estimate Apple's closing amount. The Optimized EfficientNetV2 model produced the best outcomes (MSE = 0.00008; RMSE = 0.01084).

Table 3. The outcomes of the testing for forecasting Apple's closing cost.

Method	MSE	RMSE	NRMSE	R
Optimized EfficientNetV2	0.00008	0.01084	0.02864	99.52%
EfficientNetV2	0.00054	0.02167	0.06810	97.18%

The effectiveness of the DL techniques during the testing stage is demonstrated in Fig. 6. Optimized EfficientNetV2 has a better R percentage (99.52%) than EfficientNetV2 (97.18%). Fig. 7 demonstrates the error histogram graphs.

The next stage was to check whether the closing amounts of the forecasting amounts were consistent with the actual amounts.

Results of Microsoft data

After that, Microsoft's stock market closing cost was predicted using DL algorithms. The outcomes of the DL techniques during the training stage are shown in Table 4 (using 70% of the data). The Optimized EfficientNetV2 model had a lower forecast of 2.927×10^{-5}, as shown by the MSE measurements.

Table 4. Results of DL for forecasting Microsoft's closing cost for the training stage.

Method	NRMSE	RMSE	MSE	R
Optimized EfficientNetV2	0.01576	0.00487	2.927×10^{-5}	99.98%
EfficientNetV2	0.0298	0.00941	0.0001243	99.89%

The efficiency of the Optimized EfficientNetV2 and EfficientNetV2 approaches is shown in Fig. 8.

Fig. 9 demonstrates the error histogram graphs.

Table 5 and Fig. 10 display the MSE and RMSE of the DL approaches used to forecast Microsoft's closing cost due to the testing period. Even though the RMSE and MSE were the lowest possible costs, the R percentage of Optimized EfficientNetV2 (99.81%) was close to 100. In this study, Optimized EfficientNetV2 outperformed EfficientNetV2 in terms of MSE and RMSE. Fig. 11 demonstrates the error histogram graphs.

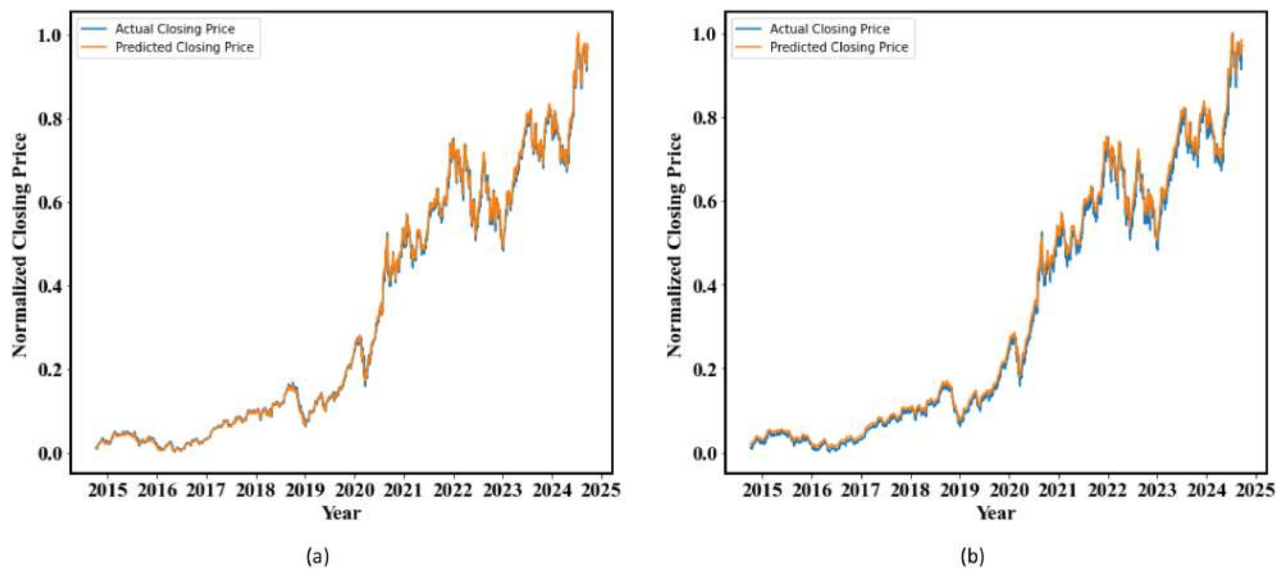


Fig. 4. During training, the following two DL methods were evaluated based on the way they predicted the closing price of the Apple stock market: (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

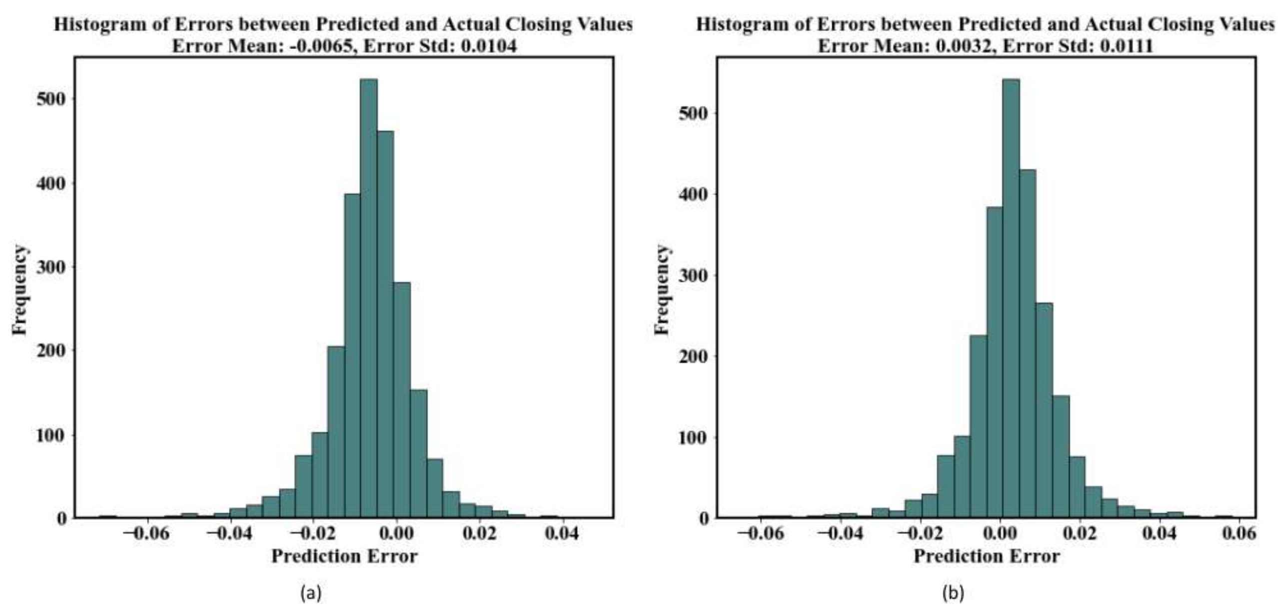


Fig. 5. The DL models' histograms to estimate Apple's closing value for the training phase are (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

Table 5. The results of the testing were used to predict Microsoft's closing price.

Method	MSE	RMSE	NRMSE	R
Optimized EfficientNetV2	2.673×10^{-5}	0.00522	0.01420	99.81%
EfficientNetV2	0.0001246	0.00947	0.0310	99.76%

For EfficientNetV2, the R percentage was 99.76%, or about one. The outcome for Optimized EfficientNetV2 was superior to those for EfficientNetV2.

Results of Tesla data

After that, Tesla's stock market closing cost was predicted using DL techniques. The outcomes of the DL techniques during the training stage are shown in Table 6 (using 70% of the data). The Optimized EfficientNetV2 technique had lower forecasting of 2.976×10^{-5}, as demonstrated by the MSE measurements.

The efficiency of the Optimized EfficientNetV2 and EfficientNetV2 techniques is shown in Fig. 12.

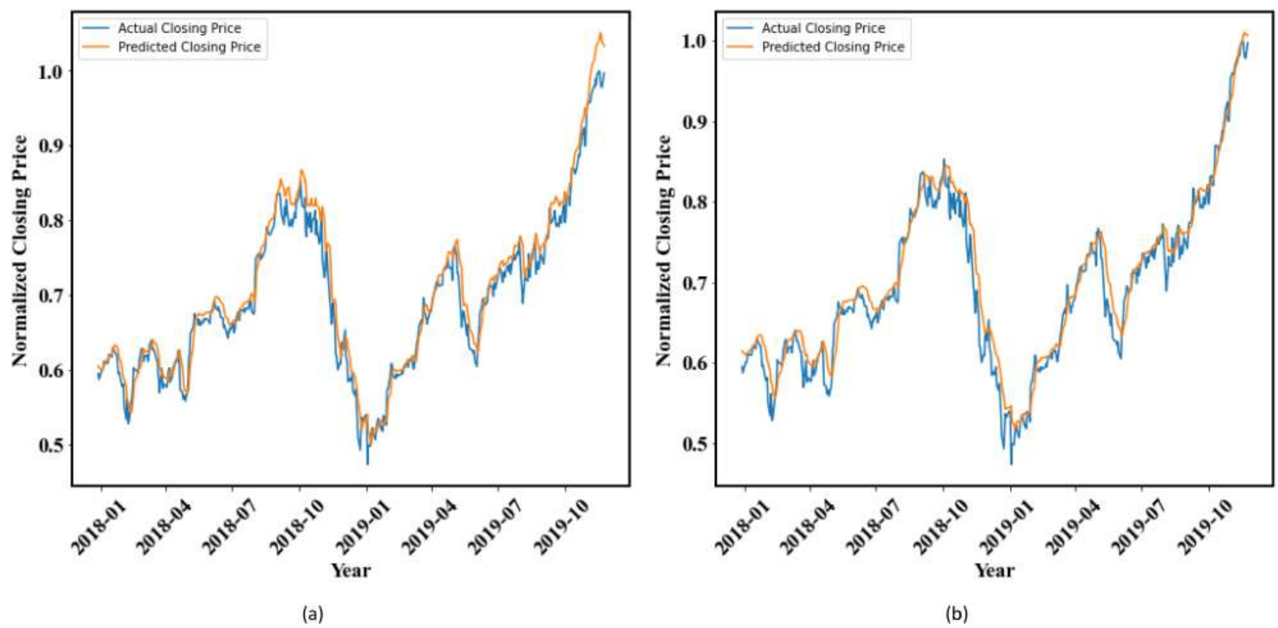


Fig. 6. During testing, the following two DL methods were evaluated based on the way they predicted the closing price of the Apple stock market: (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

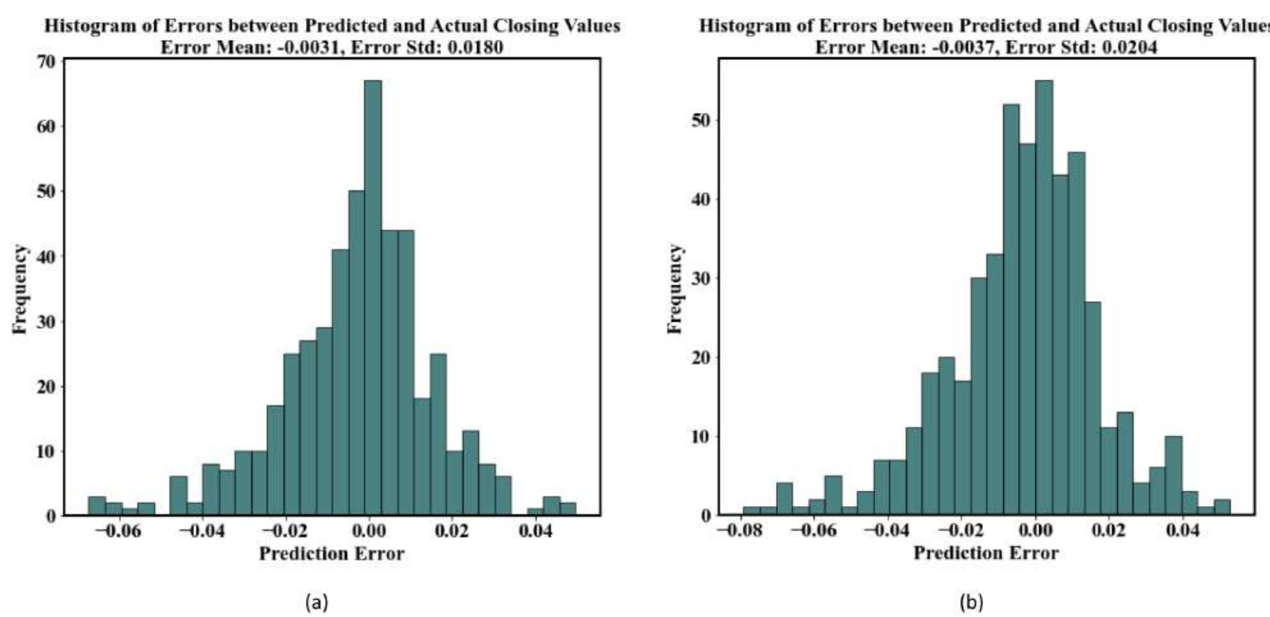


Fig. 7. The DL models' histograms for predicting Apple's closing cost for testing are (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

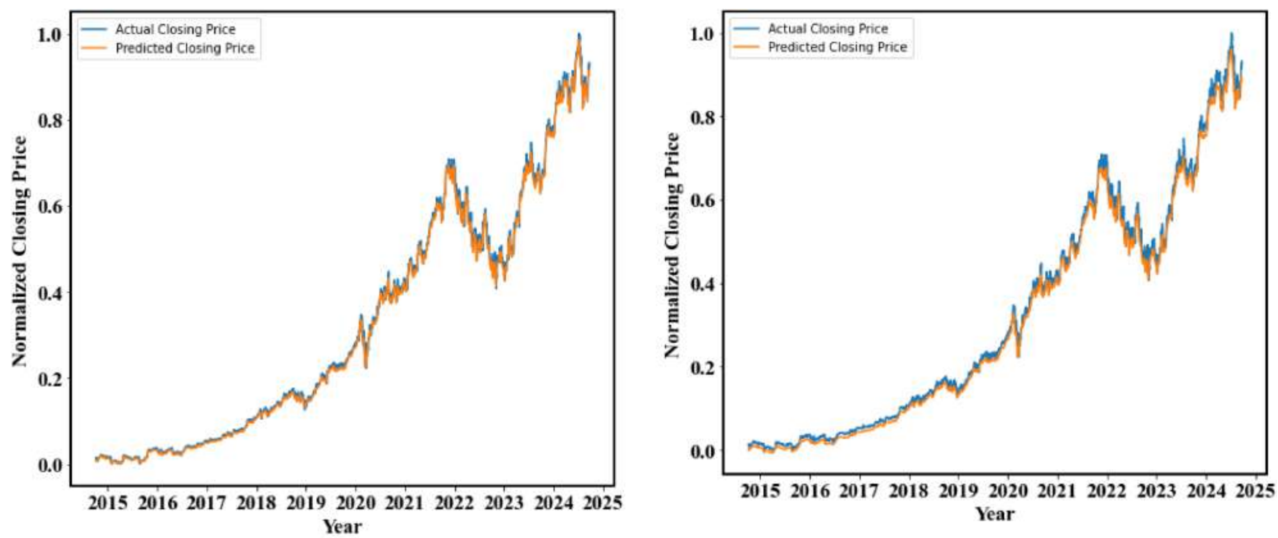
Table 6. Results of DL for forecasting the closing cost of Tesla for training.

Method	NRMSE	RMSE	MSE	R
Optimized EfficientNetV2	0.01246	0.00507	2.976×10^{-5}	99.85%
EfficientNetV2	0.0294	0.00840	0.0001245	99.72%

Fig. 13 demonstrates the error histogram graphs.

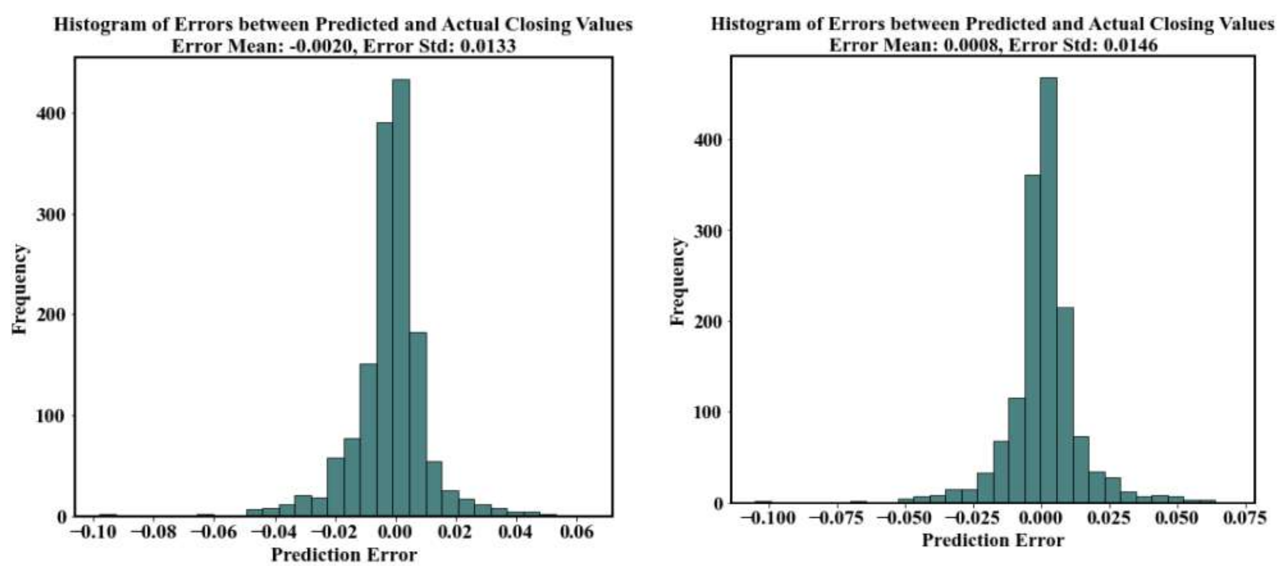
For forecasting the closing cost of Tesla for the testing stage, Table 7 and Fig. 14 display the MSE and RMSE of the DL approaches. Even though the RMSE and MSE were the lowest possible costs, the R percentage of Optimized EfficientNetV2 (99.86%) was close to 100. Fig. 15 demonstrates the error histogram graphs.

In this research, Optimized EfficientNetV2 both demonstrated decreased MSE and RMSE. EfficientNetV2 had an R percentage of 99.72%, or almost one. Results for Optimized EfficientNetV2 were superior to those for EfficientNetV2.



(a) (b)

Fig. 8. During training, the following two DL methods were evaluated based on the way they predicted the closing price of the Microsoft stock market: (a) Optimized EfficientNetV2 and (b) EfficientNetV2.



(a) (b)

Fig. 9. The DL models' histograms for forecasting Microsoft's closing value during the training stage are (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

Predicting out-of-sample periods of Apple, Microsoft, and Tesla data

Figs. 16 to 18 illustrate the predictive performance of the proposed model for out-of-sample periods across three major stocks — Apple, Microsoft, and Tesla. Each figure depicts the model's forecasting capability during bull and bear market phases. Specifically, Fig. 16(a–b), Fig. 17(a–b), and Fig. 18(a–b) represent the model's prediction outcomes for Apple, Microsoft, and Tesla, respectively, under bullish and bearish conditions.

The modeling assumptions of the suggested predictive framework, which employs EfficientNet V2 with Improved ResNeXt feature extraction and MSIB-WOA optimization, are supported by pertinent financial theories. The Efficient Market Hypothesis (EMH) states that stock prices reflect all available information, making consistent prediction difficult. Nevertheless, the model is able to extract signifi-

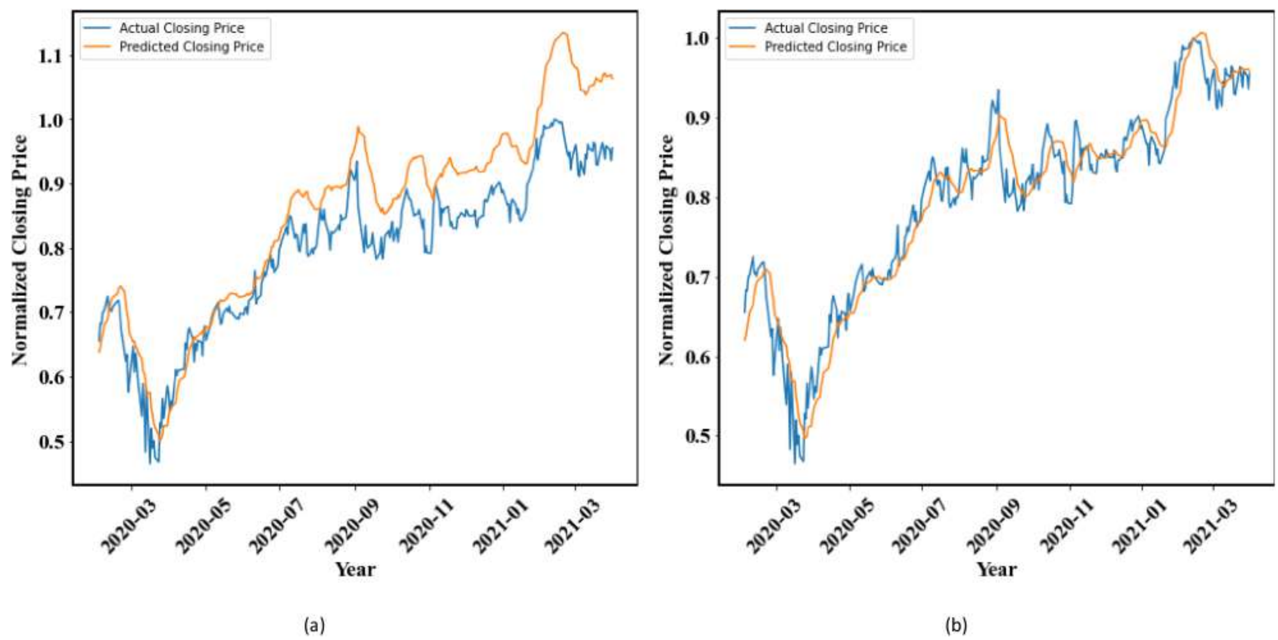


Fig. 10. During testing, the following two DL methods were evaluated based on the way they predicted the closing price of the Microsoft stock market: (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

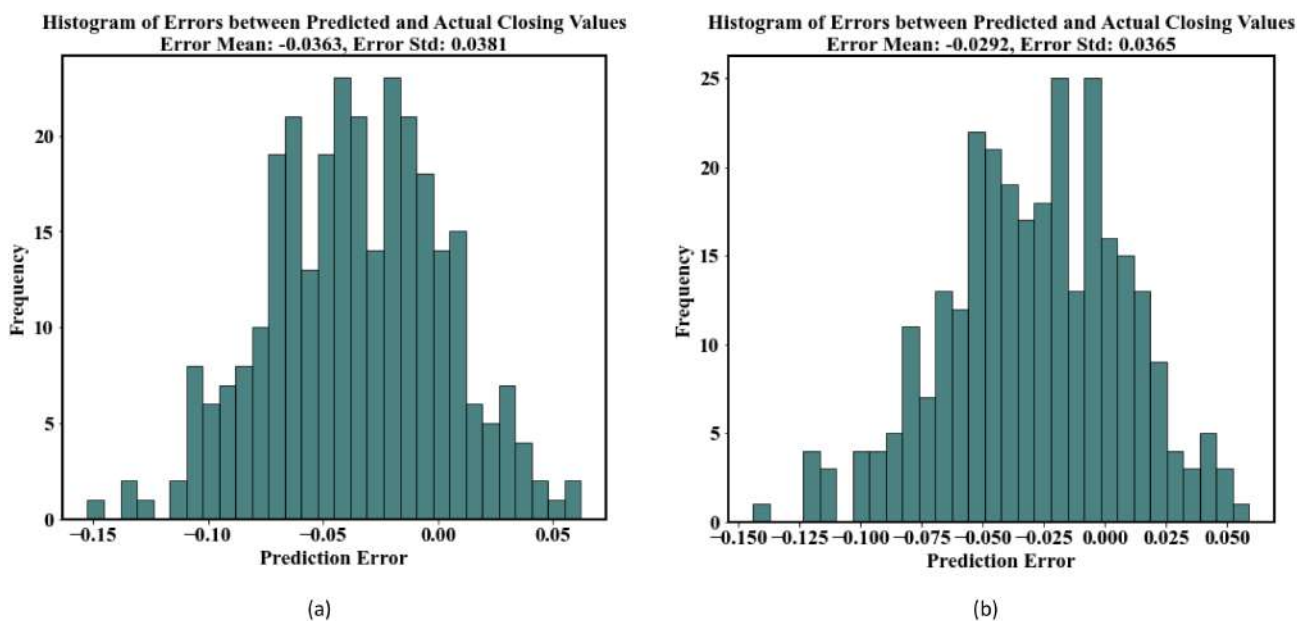


Fig. 11. The DL models' histograms for forecasting Microsoft's closing value during the testing stage are (a) Optimized EfficientNetV2 and (b) EfficientNetV2

cant trends because of short-term inefficiencies and patterns found in historical price data. Furthermore, the inclusion of previous closing prices and other historical characteristics as predictive indicators is supported by the mean-reversion theory, which contends that stock prices eventually tend to return to their long-term average. By incorporating these financial concepts, the methodology becomes both theoretically and data-driven, strengthening the justification for its feature selection and predictive approach. The suggested MSIBWOA algorithm is contrasted with a number of other well-known meta-heuristic optimization techniques in Table 8, such as Standard BWOA, Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Artificial Bee Colony (ABC). According to the results, MSIBWOA performs

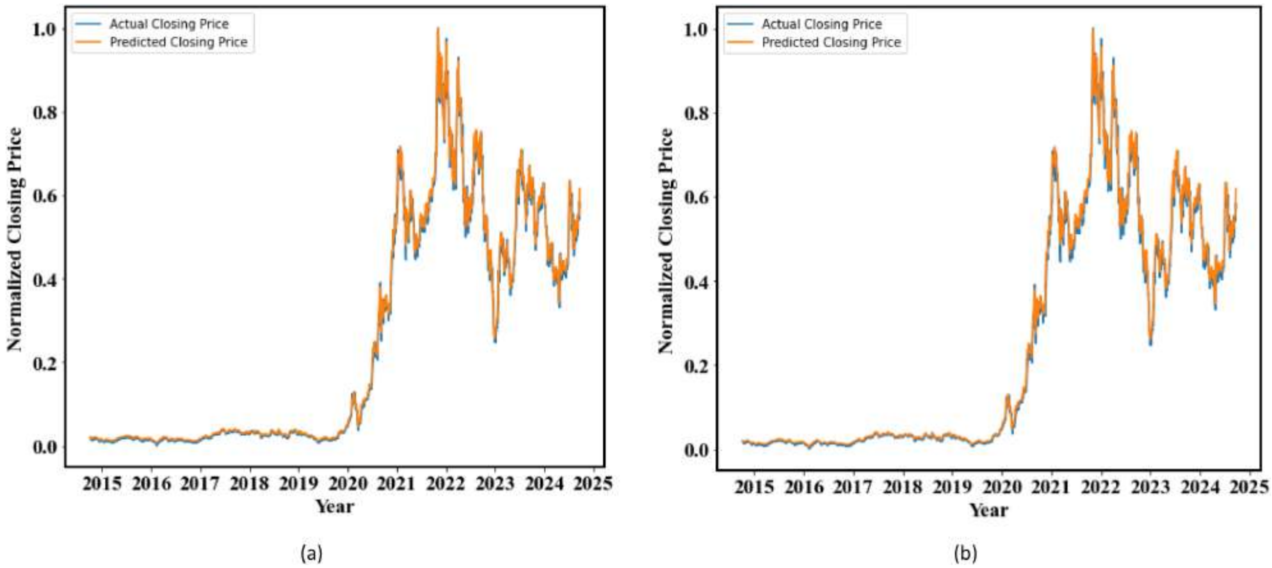


Fig. 12. During training, the following two DL methods were evaluated based on the way they predicted the closing price of the Tesla stock market: (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

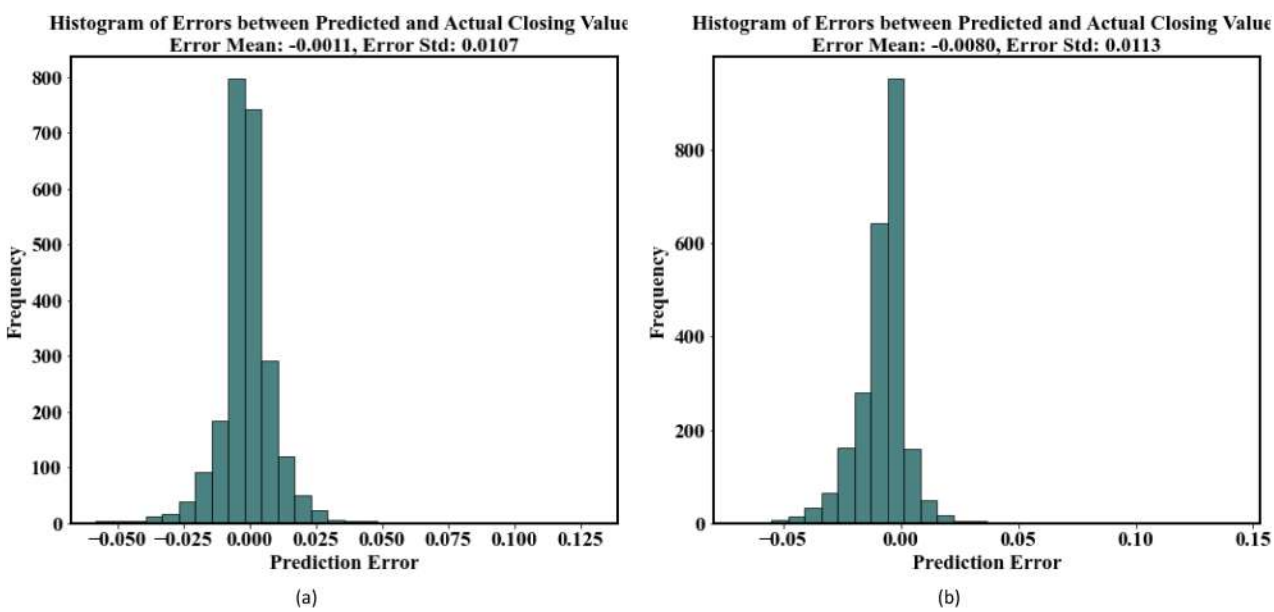


Fig. 13. The DL models' histograms for forecasting Tesla's closing value during the training stage are (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

better than the other algorithms in terms of validation accuracy (98. 5 %) achieves the lowest RMSE (0. 021) and exhibits faster convergence. MSIBWOA's multiple integrated strategies—EP improves exploration to avoid local optima, SLFSUP refines solutions in promising regions, and ICMDOBL improves global search efficiency—are responsible for its superior performance.

In contrast, while the other algorithms are effective for optimization, they either converge more slowly or are prone to getting trapped in suboptimal solutions.

Table 7. The results of the testing were used to predict Tesla's closing price.

Method	MSE	RMSE	NRMSE	R
Optimized EfficientNetV2 EfficientNetV2	2.973×10^{-5}	0.00486	0.01428	99.86%
EfficientNetV2	0.0000987	0.00941	0.0298	99.72%

These results highlight the robustness and efficiency of the proposed MSIBWOA in hyperparameter tuning for the predictive deep learning model.

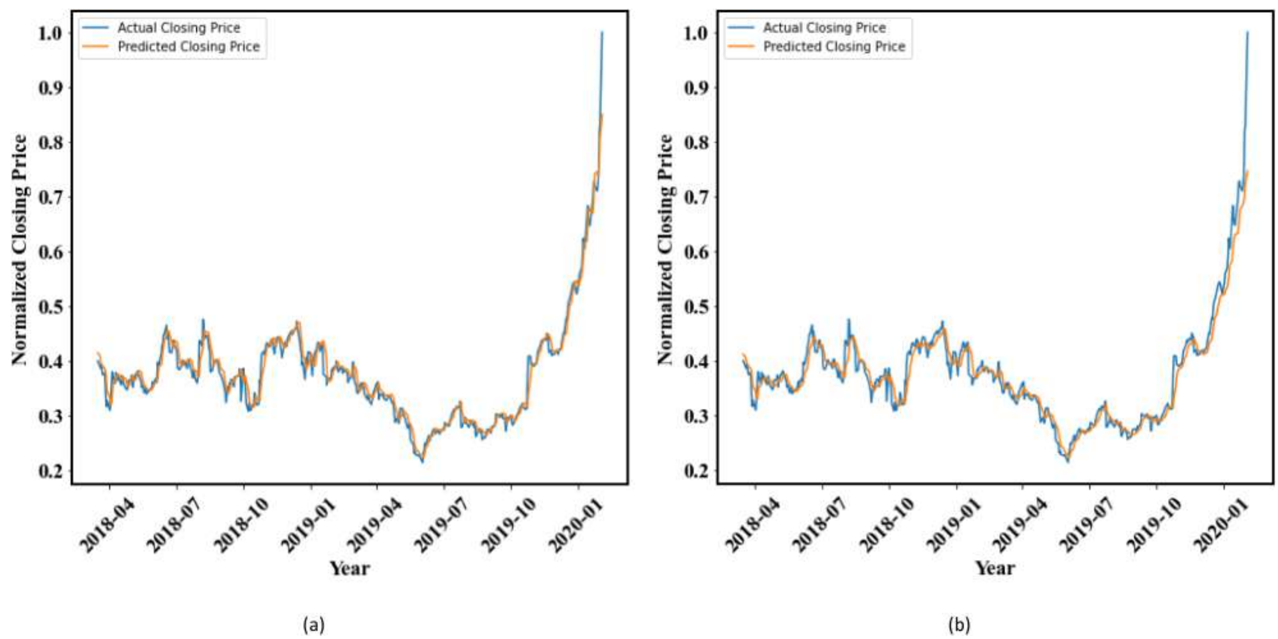


Fig. 14. During testing, the following two DL methods were evaluated based on the way they predicted the closing price of the Tesla stock market: (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

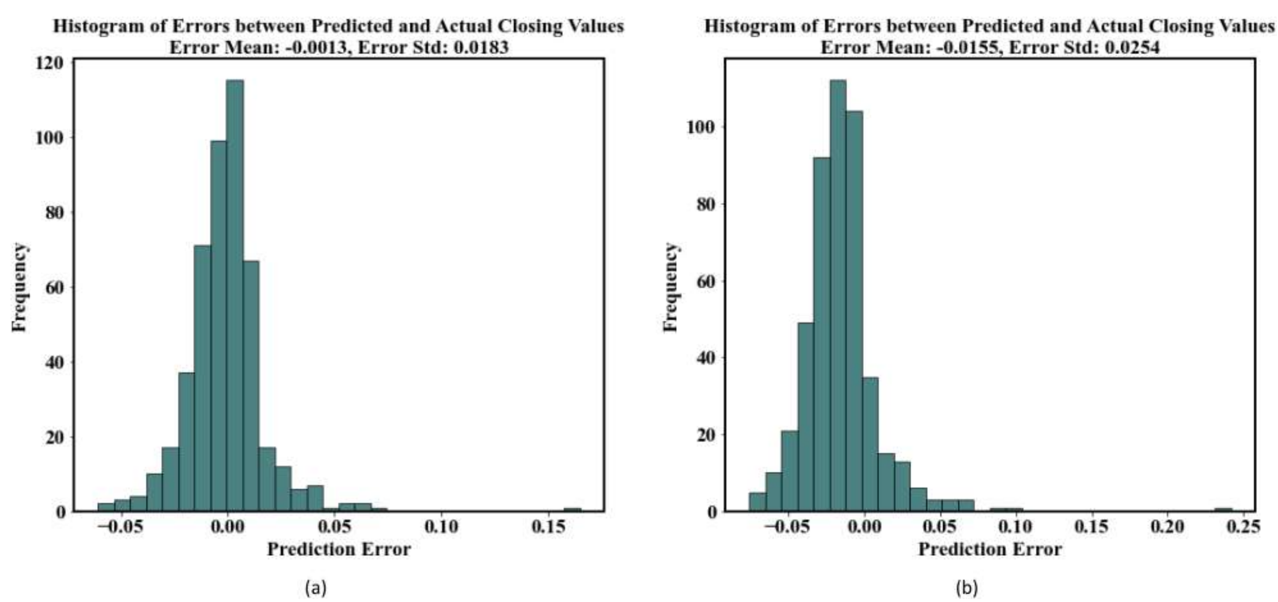


Fig. 15. The DL models' histograms for forecasting Tesla's closing value during the testing stage are (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

Table 8. Compares proposed MSIBWOA algorithm with some other algorithms.

Optimization Algorithm	Key Features / Strategies	Best Validation Accuracy (%)	RMSE	Convergence Speed
MSIBWOA (Proposed)	EP, SLFSUP, ICMDOBL	98.5	0.021	Fast
Standard BWOA [21]	Basic whale optimization	95.2	0.035	Moderate
Particle Swarm Optimization (PSO) [22]	Swarm-based search	94.8	0.038	Moderate
Genetic Algorithm (GA) [23]	Selection, crossover, mutation	93.5	0.042	Slow
Artificial Bee Colony (ABC) [27]	Foraging-based search	92.8	0.045	Slow

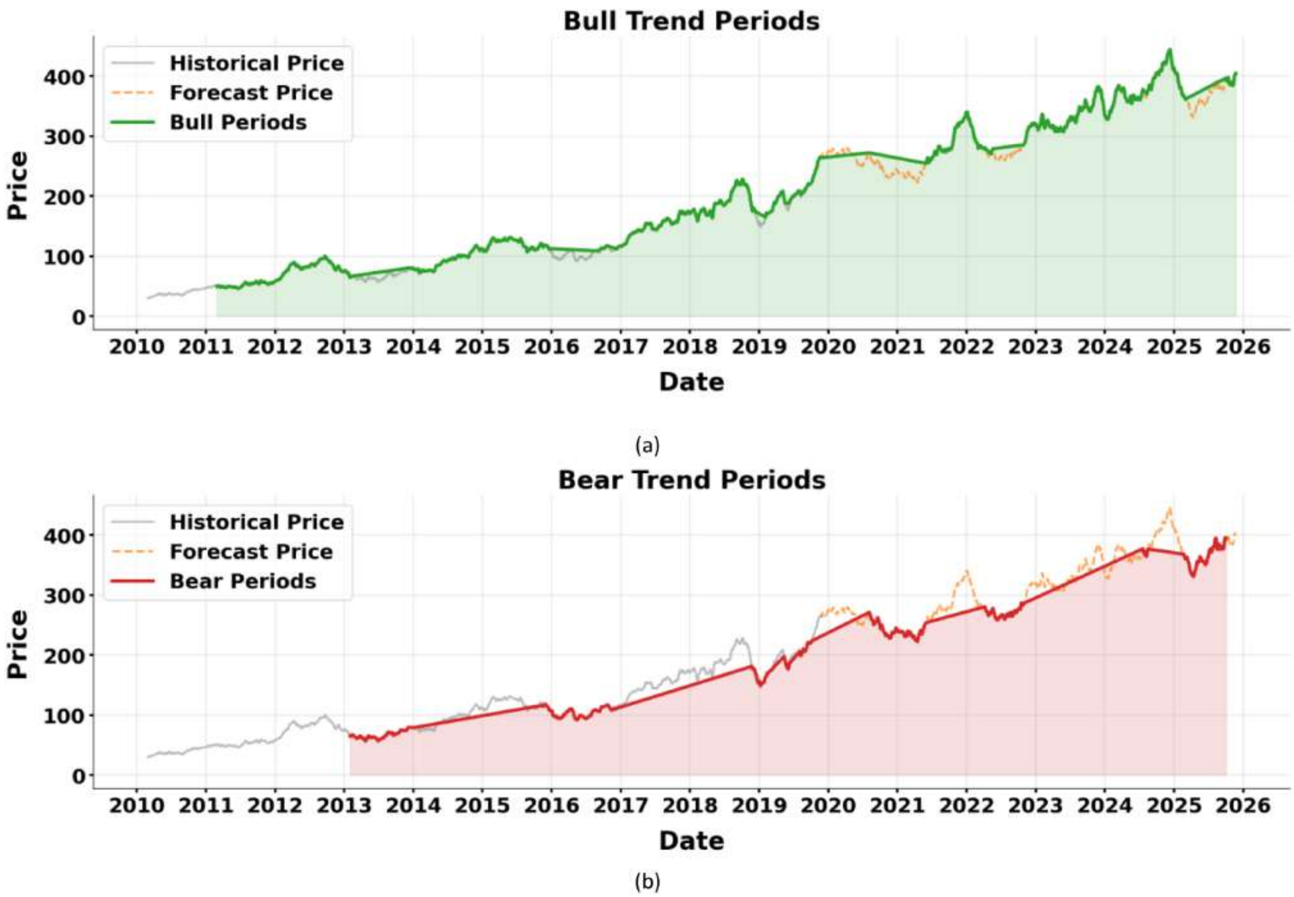


Fig. 16. Predicting the out of the periods of Apple employing (a) Bull periods and (b) Bear periods.

Predicting future values of Apple, Microsoft, and Tesla data

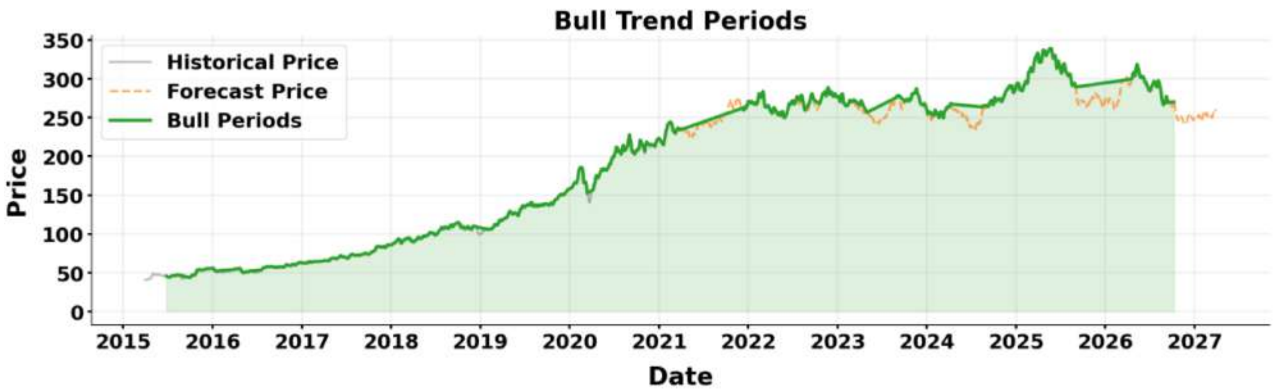
To assess the Optimized EfficientNetV2 and EfficientNetV2 algorithms, this study predicted future costs for Apple, Microsoft, and Tesla. Fig. 19 illustrates this when using the DL technique to forecast the values of Apple Corporation. The predicted results for the Microsoft Corporation generated using the DL model are shown in Fig. 20. The predicted results for the Tesla Corporation generated using the DL model are shown in Fig. 21.

Investors can easily acquire access to more stocks, and dividends provided as a component of the corporation's investor incentive scheme will benefit them greatly. Investors want to acquire stocks on the stock market whose values are expected to rise and sell stocks expected to fall. The more precisely they predict how a stock will behave, the more money they make. Developing an autonomous algorithm that can correctly forecast market movements is essential for assisting traders in maximizing their profits.

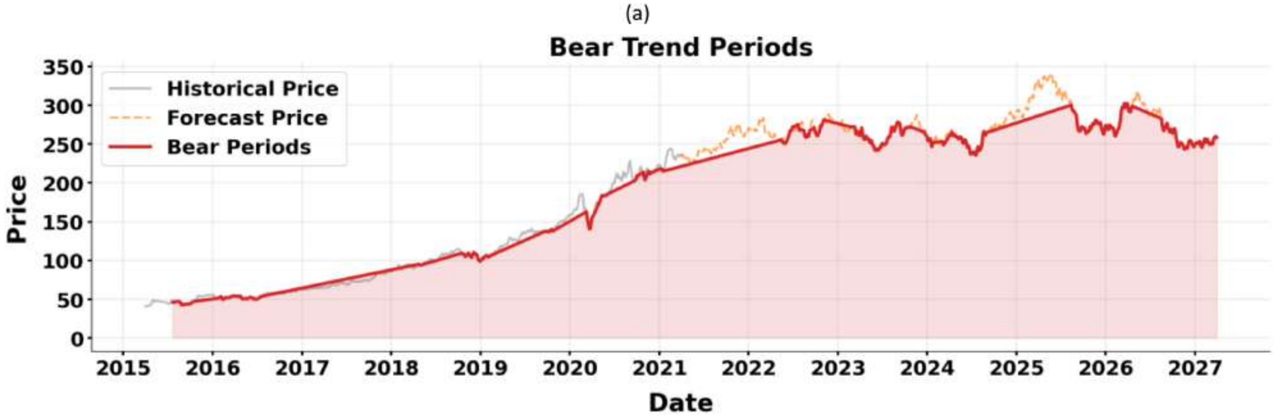
Comparative analysis of the proposed model with existing models in the literature

To show the effectiveness of EfficientNetV2, we compared the results of this proposed DL structure with those of earlier studies. Table 9 compares the outcomes of the EfficientNetV2 approach utilized in this research to those obtained using prior models. The EfficientNetV2system outperforms the models employed in another study based on the MSE metric. The accuracy for Microsoft stock prediction reached 98.85%, marking a substantial improvement over other approaches.

In Tesla stock prediction, the proposed model also delivered robust performance, achieving an R-value of 0.9985, RMSE of 0.00507, and MSE of 2.976×10^{-5}, along with an accuracy of 99.07%. Compared to CNN-LSTM's R-value of 0.9973, the proposed model demonstrates both superior correlation and significantly lower error rates. Overall, the Optimized EfficientNetV2 system exhibits exceptional predictive accuracy and lower error metrics across all datasets when compared to prior models, making it a highly



(a)



(b)

Fig. 17. Predicting the out of the periods of Microsoft employing (a) Bull periods and (b) Bear periods.

Table 9. Compares the predictions made by the proposed Optimized EfficientNetV2 system to previous studies.

Models	Dataset	R	RMSE	MSE	Accuracy
GAN-HPA [16]	TCS	0.99586647	0.001087	0.00003490	-
CNN-LSTM [17]	Apple	0.9926	0.01143	5.725×10^{-5}	-
	Tesla	0.9973	0.00756	0.0001308	-
CNN-GRU [18]	Apple	0.9793	4.9636	-	-
	Microsoft	0.9717	5.1701	-	-
	Tesla	0.9328	6.5112	-	-
DWT + LSTM [19]	Apple	-	3.61	-	-
TLBO-LSTM [20]	Twitter	-	-	-	94.73%
Proposed (Optimized EfficientNet V2)	Apple	0.9988	0.02312	0.00082	99.23%
	Microsoft	0.9998	0.00487	2.927×10^{-5}	98.85%
	Tesla	0.9985	0.00507	2.976×10^{-5}	99.07%

effective approach for stock market prediction. Due to the potential benefits, which are long-term goals for most economies and people, it might provide accurate future predictions. Those curious about the future of the stock market may find it helpful to learn how to predict changes in stock prices. Additionally, their precision will increase as algorithms become more accurate and technological advances occur. Our proposed approach produced a more precise value than previous procedures. Additionally, with less computing complexity, the closing price is predicted.

To provide deeper insights beyond raw performance metrics, a comprehensive analysis of the graphical outputs and comparative behavior of the proposed Optimized EfficientNetV2 model was conducted. Excellent generalization and low prediction variance are indicated by the visual plots of predicted versus actual stock prices across the Apple, Microsoft, and Tesla datasets. The prediction curves nearly overlap the ground truth values. Because of its better feature extraction capabilities and optimized parameter configuration, the suggested model maintained

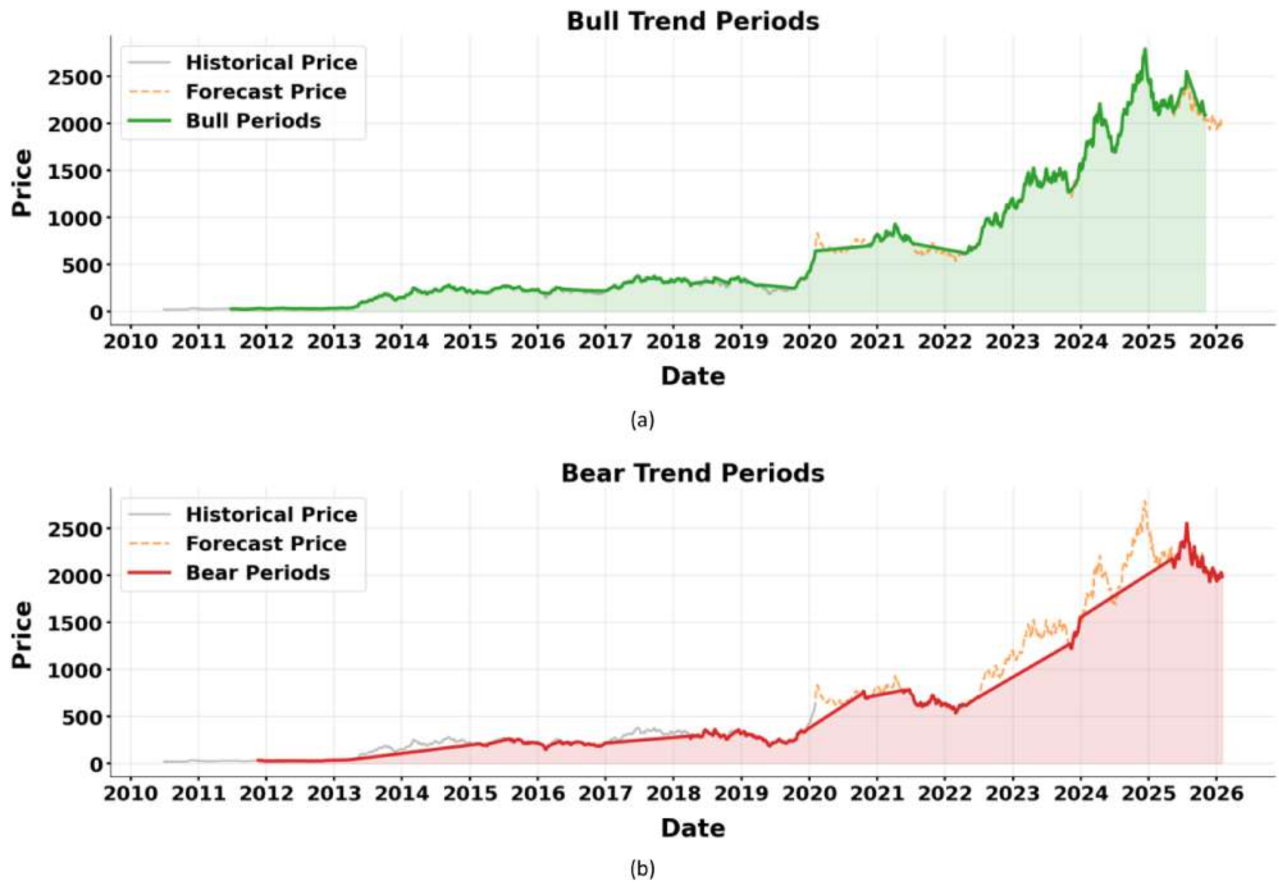


Fig. 18. Predicting the out of the periods of Tesla employing (a) Bull periods and (b) Bear periods.

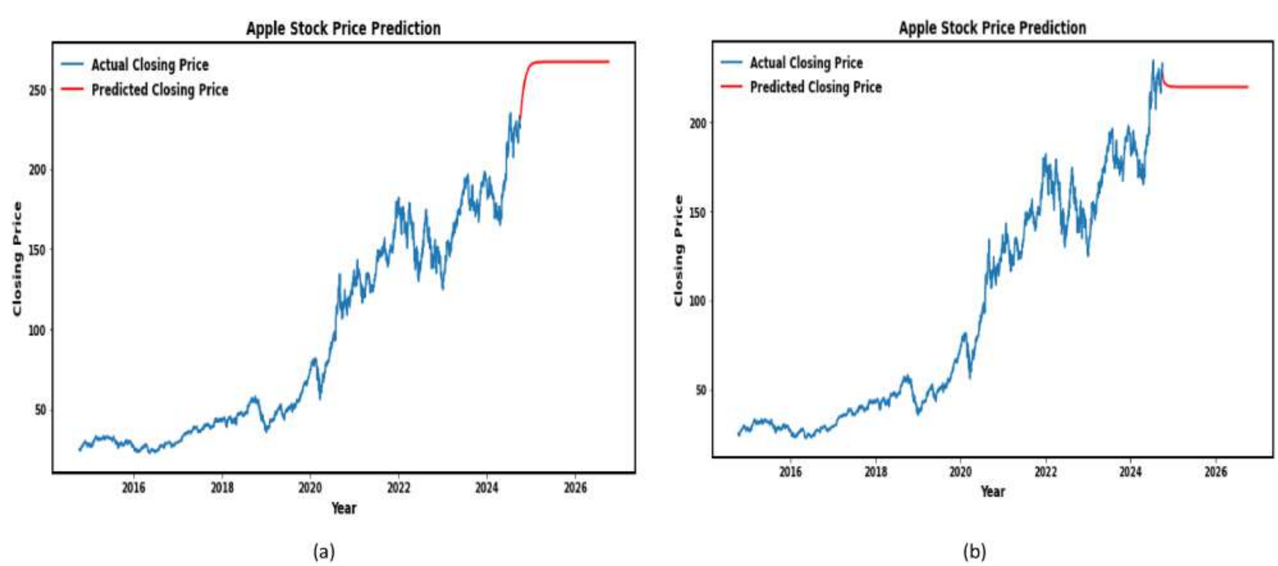


Fig. 19. Predicting the future prices of Apple employing (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

smoother tracking of price trends than previous models like CNN-LSTM and CNN-GRU, which frequently showed lag or deviation during abrupt market fluctuations. Additionally, residual error plots show that EfficientNetV2s prediction errors are closer to zero and more uniformly distributed, indicating lower bias and improved model stability. Although the model performs exceptionally well in terms of accuracy

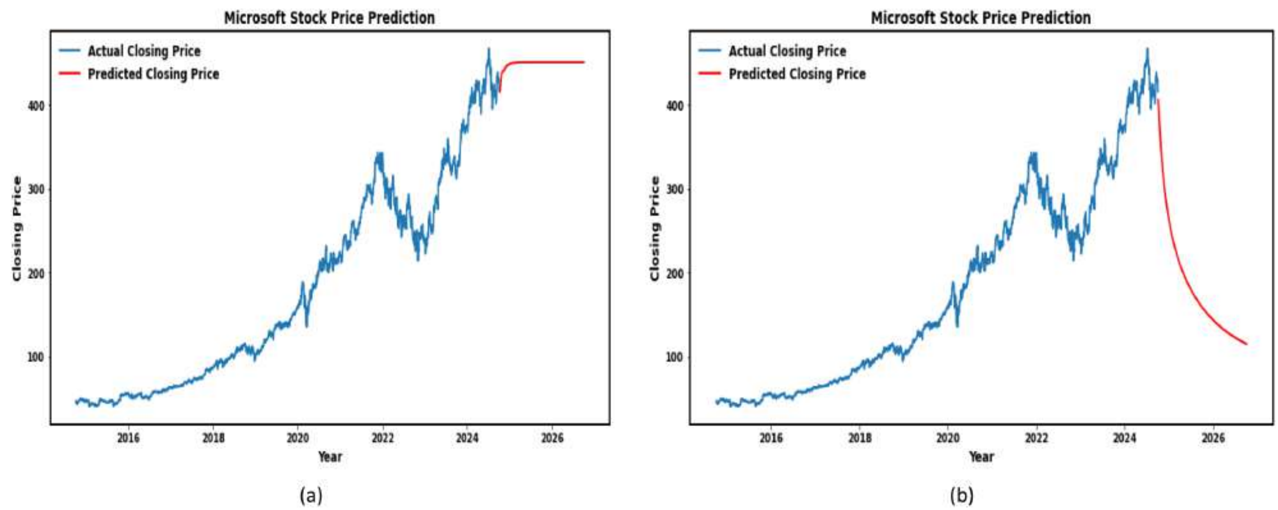


Fig. 20. Predicting the future prices of Microsoft employing (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

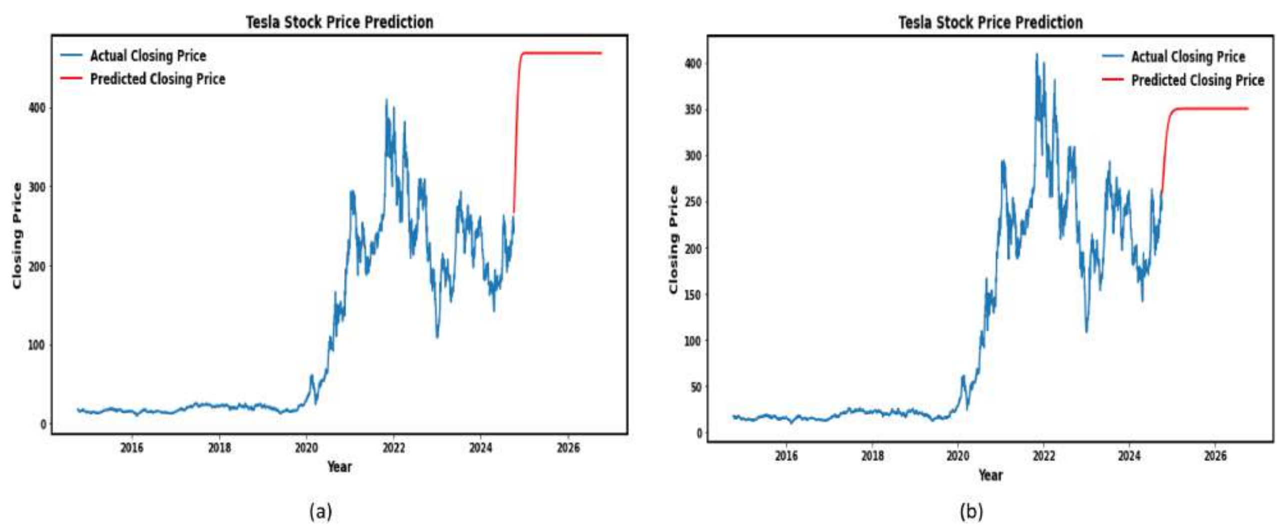


Fig. 21. Predicting the future prices of Tesla employing (a) Optimized EfficientNetV2 and (b) EfficientNetV2.

and convergence speed, EfficientNetV2s compound scaling necessitates more resources during initial training, which could be a limitation. However, the model effectively performs real-time inference once it has been optimized. Overall, the graphical and comparative analyses confirm that the suggested approach shows improved robustness, stability, and adaptability to erratic market conditions in addition to outperforming earlier models in quantitative metrics. The results of comparison are shown in Fig. 22.

Training and testing set performance evaluation

Figs. 23 to 25 display a graph of the accuracy of prediction and loss value as the amount of iteration steps increased. The graph shows the positive outcomes of applying the research's proposed converging strategy.

The suggested model is trained in 0.53 seconds, while the proposed method is tested in 0.24 seconds. The proposed approach is trained to utilize the prepared training set for 200 epochs during the training phase. It has been determined that the learning rate is 0.1. When comparing the proposed method with existing models in terms of prediction accuracy, the proposed method achieved higher accuracy.

Computational time complexity

Table 10 compares the running time complexity between the proposed model and various previously studied models. The results demonstrate that the pro-

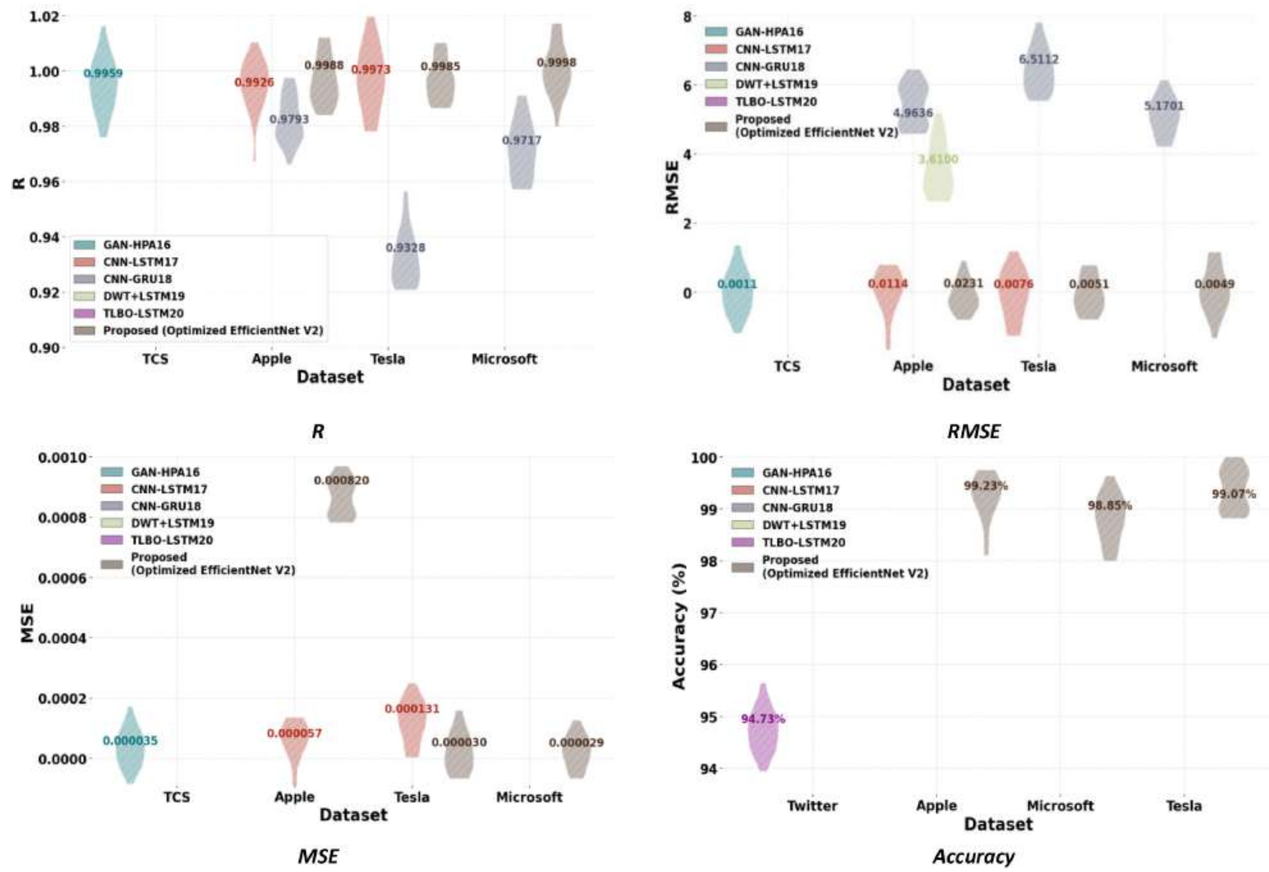


Fig. 22. Results of comparison of proposed and existing models.

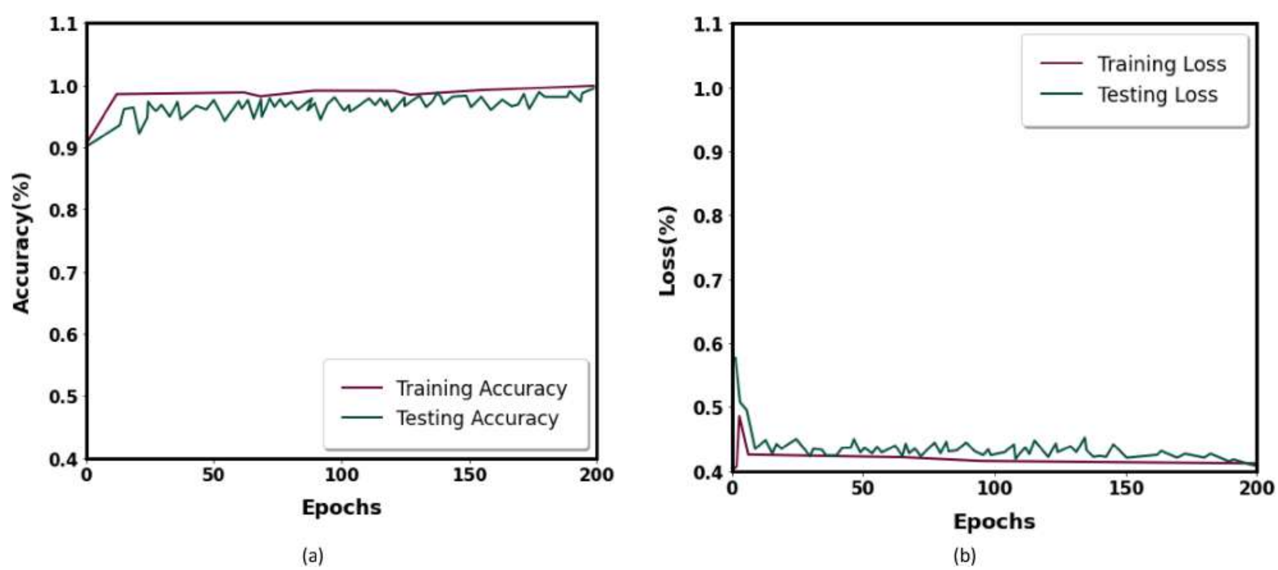


Fig. 23. For the Apple Dataset, (a) Accuracy of testing and training, (b) Loss of testing and training.

posed model significantly outperforms the others in terms of computational efficiency, achieving the lowest running time of 0.223 ms.

Comparatively, models like GAN-HPA, CNN-LSTM, CNN-GRU, DWT + LSTM, and TLBO-LSTM have higher running times, with values of 0.345 ms, 0.452 ms, 0.298 ms, 0.419 ms, and 0.385 ms, respectively. The suggested system's optimization for quicker performance is highlighted by this enhancement. The suggested model can be incorporated into a live

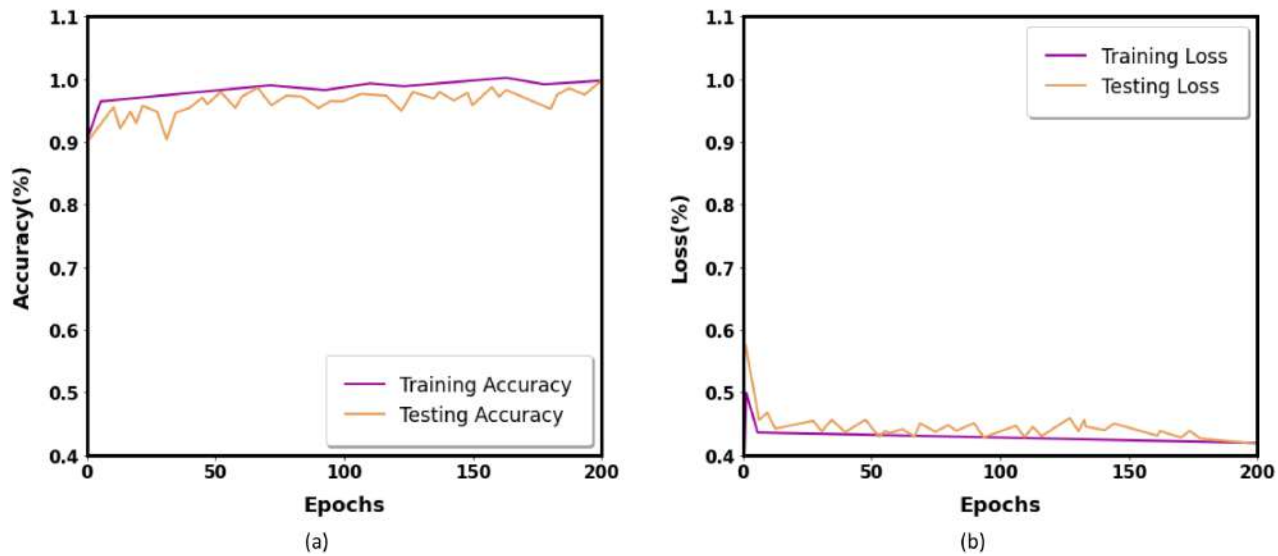


Fig. 24. For Microsoft Stock Dataset, (a) Accuracy of testing and training, (b) Loss of testing and training.

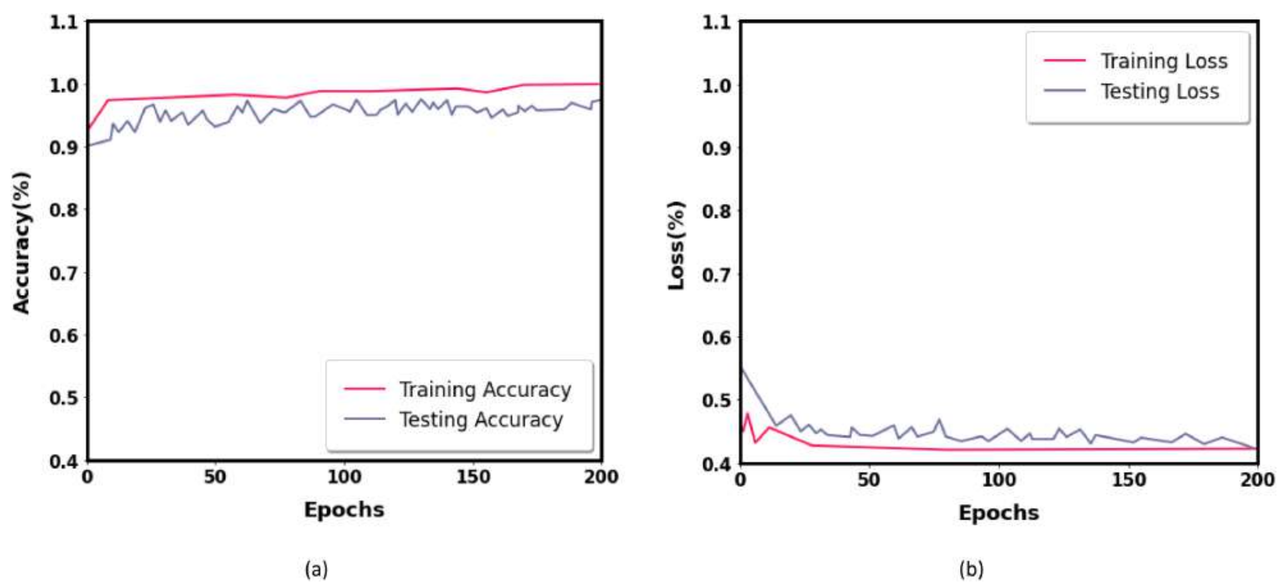


Fig. 25. For Tesla Stock Dataset, (a) the accuracy of testing and training and (b) the loss of testing and training.

Table 10. Compares the running time complexity made by the proposed system to previous studies.

Models	Running time (ms)
GAN-HPA [16]	0.345
CNN-LSTM [17]	0.452
CNN-GRU [18]	0.298
DWT + LSTM [19]	0.419
TLBO-LSTM [20]	0.385
Proposed Model	0.223

prediction system with a streamlined data pipeline, appropriate hardware, and stringent latency management to enable real-time deployment. It is possible to continuously gather real-time market data from financial APIs and send it via message-queue systems for quick feature extraction, preprocessing, and forecasting. The system can run on GPU-enabled hardware for effective inference either on-premise (e. g. The g. cloud-based servers) utilizing scalable GPU instances—such as the NVIDIA RTX series—to manage high data throughput during active trading periods. The end-to-end latency from data ingestion to prediction should stay between one and three seconds in order to satisfy real-time operational requirements, allowing traders and analysts to make decisions on time. Asynchronous data handling, model quantization, and CUDA-accelerated inference are examples of optimizations that help keep la-

Table 11. Firedman's average ranks.

Model	R^2 average rank	MSE average rank
GS-ARIMA	4.78	4.44
PSO-ESN	3.00	2.00
GS-MLP	2.33	4.00
Proposed Model	1.23	1.21
p-value	1.32×10^{-4}	1.02×10^{-4}

tency low while guaranteeing reliable performance. Lastly, using WebSocket-based services or REST APIs, the prediction outputs can be sent to trading dashboards or automated decision-support tools, enabling smooth integration with current financial platforms and real-time market workflows.

Statistical significance test

The Friedman nonparametric test is used to test the null hypothesis that no models under comparison differ significantly from one another. When the Friedman test rejects the null hypothesis, it is interesting to see which models differ significantly. Next, this work uses the Nemenyi test. In Friedman's test, the models are ranked according to their forecasting accuracy; the best-performing model is given rank 1. In the case of a tie, the average rank is provided. The Friedman statistic is then calculated as in Eq. (34).

$$p = \frac{12T}{k\,(k+1)}\left[\sum_{j} R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{34}$$

If the *P*-value is less than 0. 05, which rejects the assumption that all models are comparable with 95 % confidence, the post-hoc Nemenyi test is used to perform pairwise comparisons across the available approaches. If the average ranks of the models differ by at least a critical distance (CD), then the models can be considered statistically distinct. The CD is calculated using Eq. (35).

$$CD = q_a\sqrt{\frac{k\,(k+1)}{6T}} \tag{35}$$

Table 11 displays the test results for the firemans average rank.

Using historical data from Apple, Microsoft, and Tesla, the suggested predictive model shows great promise for incorporation into useful trading systems and investment platforms. Using real-time stock market data feeds from APIs, the model's forecasts can be updated on a regular basis and converted into buy or sell signals. With this kind of integration, traders and investors can use the model's results directly to make decisions in real-time market settings. Latency or the amount of time it takes to generate a prediction, is crucial in financial markets because even slight delays can make signals less valuable in quickly moving markets. By ensuring computational efficiency in comparison to more complex deep learning models, EfficientNet V2 lowers prediction latency. Even though training might require a lot of resources, the model can produce predictions almost instantly once it is put into use, which makes it appropriate for trading environments with tight deadlines.

Ethical considerations

There are various ethical issues with using deep learning models for financial prediction, such as the suggested EfficientNet V2 with Improved ResNeXt feature extraction and MSIBWOA optimization. In order to prevent giving certain investors or institutions an unfair advantage, algorithmic predictions must be used responsibly, making fairness crucial. Maintaining adherence to financial regulations is also necessary to prevent market manipulation and trading law violations using the model's predictions. The societal effects of algorithm-driven markets should be taken into account in addition to individual fairness and compliance, extensive automated trading may increase market volatility or worsen financial inequality. To encourage the responsible and ethical application of the suggested predictive framework, mechanisms like keeping an eye out for biased predictions, evaluating possible market impact, and guaranteeing regulatory compliance will be investigated in subsequent work.

Limitations & future works

Despite the model's high predictive accuracy, there are a number of obstacles to overcome when implementing it on actual trading platforms. Because financial markets are dynamic and volatile, the model must be able to manage constant data streams, adjust to abrupt changes in market regimes, and remain resilient in both bull and bear markets. A crucial issue for real-world implementation is making sure the model and brokerage or trading systems are connected securely and effectively. This study plan to expand on the suggested work in subsequent studies by investigating the predictive framework's real-time implementation in real-world trading scenarios. This will entail tackling important issues like incorporating streaming stock market data, guaranteeing low-latency reactions in erratic market circumstances, and constantly retraining the model to adjust to changing financial dynamics. In order to facilitate deployment across numerous stocks and indices concurrently without sacrificing computational efficiency, scalability will also be investigated. The model can be converted from a purely predictive framework into a reliable decision-support tool for practical financial applications by addressing these deployment issues.

Conclusion

Predicting the future price of a company's stock or any other monetary product traded on an exchange is a complex task. A stock's accurate future price prediction could result in a significant profit. This study employed a deep learning technique to predict future stock prices more accurately in pre-processing. Feature were then extracted using the Improved ResNeXt method. Finally, the future stock closing prices were predicted using the EfficientNet V2 technique with an optimization algorithm for higher prediction accuracy. The evaluation performance of this research achieved high accuracy. With the help of the feature selection method and the optimization algorithm, it achieved higher prediction accuracy with less computation time. Here, this research overcomes the previous research problems. and it gained 99.23%, 98.85%, and 99.07% accuracy for Apple, Microsoft, and Tesla, respectively.

Acknowledgements

We declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere.

Authors' declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images that are not ours have been included with the necessary permission for re-publication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at Saveetha University.

Authors' contributions statement

M. R. Conceptualization, Methodology, Software, Formal Analysis, Investigation, Resources, Writing – Original Draft, Writing - Review & Editing, Visualization. R. B. Conceptualization, Writing - Review & Editing, Original Draft, Writing - Review & Editing, Visualization.

Data availability

The authors will provide the data upon reasonable request.

References

1. Jiang W. Applications of deep learning in stock market prediction: recent progress. Expert Syst Appl. 2021;184:115537. https://doi.org/10.1016/j.eswa.2021.115537.
2. Chhajer P, Shah M, Kshirsagar A. The applications of artificial neural networks, support vector machines, and long–short-term memory for stock market prediction. Decis Anal J. 2022;2:100015. https://doi.org/10.1016/j.dajour.2021.100015.
3. Rouf N, Malik MB, Arif T, Sharma S, Singh S, Aich S, Kim HC. Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. Electronics. 2021;10(21):2717. https://doi.org/10.3390/electronics10212717.
4. Alkanhel R, El-kenawy ESM, Abdelhamid AA, Ibrahim A, Alohali MA, Abotaleb M, Khafaga DS. Network Intrusion Detection Based on Feature Selection and Hybrid Metaheuristic Optimization. Comput Mater Continua. 2023;74(2). https://doi.org/10.32604/cmc.2023.033273.
5. Abdelhamid AA, El-Kenawy ESM, Ibrahim A, Eid MM, Khafaga DS, Alhussan AA, Shams MY. Innovative feature selection method based on hybrid sine cosine and dipper throated optimization algorithms. IEEE access. 2023;11:79750–79776. https://doi.org/10.1109/ACCESS.2023.3298955.
6. El-Kenawy ESM, Khodadadi N, Mirjalili S, Makarovskikh T, Abotaleb M, Karim FK, Khafaga DS. Metaheuristic optimization for improving weed detection in wheat images captured by drones. Mathematics. 10(23):2022;4421. https://doi.org/10.3390/math10234421.
7. Atteia G, El-kenawy ESM, Samee NA, Jamjoom MM, Ibrahim A, Abdelhamid AA, Shams MY. Adaptive dynamic dipper throated optimization for feature selection in medical data. Comput Mater Continua. 2023;75(1), 1883–1900. https://doi.org/10.32604/cmc.2023.031723.
8. Koukaras P, Nousi C, Tjortjis C. Stock market prediction using microblogging sentiment analysis and machine learning. In Telecom. 2022;3(2):358–378. MDPI. https://doi.org/10.3390/telecom3020019.
9. Gunduz H. An efficient stock market prediction model using a hybrid feature reduction method based on variational autoencoders and recursive feature elimination. Financ innovation. 2021;7(1):28. https://doi.org/10.1186/s40854-021-00243-3.
10. Abdullah HS, Ali NH, Jassim AH, Hussain SH. An Analytical Comparison of the Behavior of Machine Learning and Deep Learning in Stock Market Prediction. Baghdad Sci. J. 2025;22(1):297–309. https://doi.org/10.21123/bsj.2024.10017.
11. Ananthi M, Vijayakumar K. Retracted article: stock market analysis using candlestick regression and market trend prediction (CKRM). J Ambient Intell Hum Comput.

2021;12(5):4819–4826. https://doi.org/10.1007/s12652-020-01892-5.
12. Sisodia PS, Gupta A, Kumar Y, Ameta GK. Stock market analysis and prediction for NIFTY50 using LSTM Deep Learning Approach. In 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM). 2022;2:156–161. IEEE. https://doi.org/10.1109/ICIPTM54933.2022.9754148.
13. Albahli S, Nazir T, Mehmood A, Irtaza A, Alkhalifah A, Albattah W. AEI-DNET: a novel DenseNet model with an autoencoder for the stock market predictions using stock technical indicators. Electronics. 2022;11(4):611. https://doi.org/10.3390/electronics11040611.
14. Ingle V, Deshmukh S. Ensemble deep learning framework for stock market data prediction (EDLF-DP). Global Transitions Proc. 2021;2(1):47–66. https://doi.org/10.1016/j.gltp.2021.01.008.
15. Kumar K, Haider MTU. Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN–LSTM network. New Gener Comput. 2021;39:231–272. https://doi.org/10.1007/s00354-020-00104-0.
16. Dael FA, Yavuz ÖÇ, Yavuz U. Stock Market Prediction Using Generative Adversarial Networks (GANs): Hybrid Intelligent Model. Comput Syst Sci Eng. 2023;47(1):19–35. https://doi.org/10.32604/csse.2023.037903.
17. Aldhyani TH, Alzahrani A. Framework for predicting and modeling stock market prices based on deep learning algorithms. Electronics. 2022;11(19):3149. https://doi.org/10.3390/electronics11193149.
18. Zhou Q, Zhou C, Wang X. Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection. Plos one. 2022;17(2):1–20. https://doi.org/10.1371/journal.pone.0262501.
19. Yang Y. Forecasting Apple Stock Closed Prices by LR and LSTM with Discrete Wavelet Transformation. In 2022 2nd International Conference on Economic Development and Business Culture (ICEDBC 2022). 2022;1:935–943. Atlantis Press. https://doi.org/10.2991/978-94-6463-036-7_138.
20. Swathi T, Kasiviswanath N, Rao AA. An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis. Appl Intell. 2022;52(12):13675–13688. https://doi.org/10.1007/s10489-022-03175-2.
21. Gökhan AKSU, Güzeller CO, Eser MT. The effect of the normalization method used in different sample sizes on the success of the artificial neural network model. Int J Assess Tool Educ. 2019;6(2):170–192. https://doi.org/10.21449/ijate.479404.
22. Wang G, Wang J, Yu H, Sui Y. Research on identification of corn disease occurrence degree based on improved ResNeXt network. Int J Pattern Recognit Artif Intell. 2022;36(02):2250005. https://doi.org/10.1142/S0218001422500057.
23. Sunil CK, Jaidhar CD, Patil N. Cardamom plant disease detection approach using EfficientNetV2. IEEE Access. 2021;10:789–804. https://doi.org/10.1109/ACCESS.2021.3138920.
24. Kottarathil P. Microsoft Historical Dataset [Internet]. 2020.
25. Kottarathil P. Apple Historical Dataset [Internet]. 2019.
26. Bozsolik T. Tesla stock data from 2010 to 2020 [Internet]. 2020.
27. Singh A, Sharma A, Rajput S, Mondal AK, Bose A, Ram M. Parameter extraction of the solar module using the sooty tern optimization algorithm. Electronics. 2022;11(4):564. https://doi.org/10.3390/electronics11040564.

التنبؤ بأسعار إغلاق الأسهم المستقبلية: نهج من خمس مراحل باستخدام البيانات التاريخية وتقنيات التعلم العميق المُحسّنة مع استراتيجيات متعددة لتحسين أداء حوت بيلوغا

موراهاري ريدي، ر. بالامانيجاندان

كلية سافيثا للهندسة، معهد سافيثا للعلوم الطبية والتقنية، جامعة سافيثا، تشيناي، تاميل نادو، الهند.

الخلاصة

يُعدّ التنبؤ بتحركات سوق الأسهم مهمة معقدة وصعبة نظرًا لتعدد العوامل المؤثرة على أسعار الأسهم. ورغم وجود العديد من المناهج والتقنيات التي استكشفها الباحثون والمحللون، إلا أنه من الضروري الإشارة إلى أن التنبؤ الدقيق والمستمر بأسعار الأسهم أمرٌ بالغ الصعوبة. فالبيانات المالية قد تكون مشوشة وعرضة للأخطاء، حتى أن أدنى خطأ فيها قد يؤدي إلى تنبؤات غير دقيقة. في هذا البحث، نقترح تقنية جديدة للتعلم العميق مع خوارزمية تحسين للتغلب على هذه القيود في التنبؤ بسوق الأسهم. يتألف هذا البحث من خمس مراحل للتنبؤ بأسعار الإغلاق باستخدام البيانات التاريخية. تُجمع بيانات الإدخال مبدئيًا من ثلاث مجموعات بيانات (أبل، مايكروسوفت، وتسلا). تخضع هذه البيانات لعمليات معالجة مسبقة تشمل إزالة علامات الترقيم، وإزالة القيم الشاذة، واستبدال القيم المفقودة، وتطبيع البيانات، وتنظيفها، وذلك لضمان دقة التنبؤ. بعد ذلك، تُستخرج الميزات باستخدام منهج ResNeXt المُحسّن لتحسين أداء التنبؤ. وأخيرًا، تُستخدم تقنية جديدة للتعلم العميق، وهي EfficientNet V2، للتنبؤ بأسعار إغلاق الأسهم المستقبلية بناءً على البيانات التاريخية. لتحسين دقة التنبؤ، تستخدم هذه الدراسة خوارزمية تحسين حيتان البيلوغا متعددة الاستراتيجيات (MSIBWOA) لتحسين المعلمات الفائقة. ولتحديد نتائج أساليب التنبؤ بمخزون الحيتان باستخدام التعلم العميق، تم استخدام مقاييس الدقة، ومتوسط مربع الخطأ (MSE)، ومعامل ارتباط بيرسون (R)، وجذر متوسط مربع الخطأ المعياري (NRMSE)، وجذر متوسط مربع الخطأ (RMSE). تمت مقارنة أداء الطريقة المقترحة مع أحدث التقنيات. وبالمقارنة مع أكثر الخوارزميات الأساسية تطورًا، أظهرت التقييمات تحسنًا ملحوظًا في أداء التنبؤ.

الكلمات المفتاحية: التعلم العميق، ResNeXt، EfficientNet V2 المحسّن، التحسين، مجموعات بيانات الأسهم، التنبؤ بسوق الأسهم.