# A Financial Time Series Forecasting Model Using Quasi-Recurrent Neural Networks and the Crown Porcupine Optimizer for Stock Market Risk Prediction

**Ilyos Abdullayev**
Department of Business and Management, Urgench State University, Urgench, Uzbekistan
ilyos.a@urdu.uz

**Elvir Akhmetshin**
Department of Scientific Research Innovations and Scientific and Pedagogical Personnel Training, Mamun University, Khiva, Uzbekistan | Faculty of Economics, RUDN University, Moscow, Russia | Khorezm University of Economics, Urgench, Uzbekistan
akhmetshin@mamunedu.uz (corresponding author)

**Emil Hajiyev**
Department of Business Management, Azerbaijan State University of Economics (UNEC), Baku, Azerbaijan
hajiyev.emil@unec.edu.az

**Zokir Mamadiyarov**
Department of Finance and Tourism, Termez University of Economics and Service, Termez, Uzbekistan | Department of Finance, Alfraganus University, Tashkent, Uzbekistan | Department of Bank Accounting and Auditing, Tashkent State University of Economics, Tashkent, 100066, Uzbekistan
z.mamadiyarov@afu.uz

**Tatyana Khorolskaya**
Department of Money Circulation and Credit, Kuban State Agrarian University named after I.T. Trubilin, Krasnodar, Russia
xorolskaya.t@edu.kubsau.ru

**E. Laxmi Lydia**
Department of Computer Science and Engineering, Vignan's Institute of Engineering for Women, Visakhapatnam, Andhra Pradesh, India
elaxmi2002@yahoo.com

*Received: 12 July 2025 | Revised: 1 August 2025, 19 August 2025, 8 September 2025, 17 September 2025, 22 September 2025, and 27 September 2025 | Accepted: 28 September 2025*

 *DOI: https://doi.org/10.48084/etasr.13327*

## ABSTRACT

**Stock is a financial product described by flexible trading, high risk, and high return, which is favored by many investors. Investors may obtain sufficient returns by precisely approximating stock price developments. However, a stock price is affected by several factors, such as market conditions, macroeconomic situation, major economic and social events, and managerial decisions of companies. As a result, Stock Price Prediction (SPP) has consistently attracted attention and is an important field of investigation. Currently, Machine Learning (ML) is extensively utilized in SPP, but considerably more**

**appropriate techniques are suggested for the prediction of stock prices, as numerous investigations have proved that Deep Learning (DL) has better efficacy than other methods. The paper proposes a Financial Time Series Forecasting Using Quasi Recurrent Neural Network and Crown Porcupine Optimizer (FTSF-QRNNCPO) method, offering an intelligent framework for financial stock market price prediction to accurately assess market volatility and forecast associated investment risks. The FTSF-QRNNCPO method begins with data preprocessing, comprising missing value handling, data cleaning, and normalization to prepare the input data effectively. The Secretary Bird Optimization Algorithm (SBOA) is employed for optimal Feature Selection (FS). Then, the Quasi-Recurrent Neural Network (QRNN) approach is employed for prediction. The Crown Porcupine Optimizer (CPO) approach is employed in the parameter tuning process. The FTSF-QRNNCPO technique was experimentally evaluated on a Tesla stock price dataset, demonstrating superior accuracy, achieving a significantly lower MAPE of 0.415% and outperforming previous models.**

***Keywords-financial time series forecasting; quasi recurrent neural network; crown porcupine optimizer; stock market; risk prediction***

## I. INTRODUCTION

The stock market is a vital part of a country's financial system, serving as a global hub of capital exchange [1]. It plays an essential role in the channeling and guiding of distributed funds and savings toward the most efficient avenues [2]. Thus, with careful planning, the limited financial resources are spread across numerous profitable ventures and projects. Speculators and investors in stock markets focus on making increased profits from market information analysis [3]. Stock Price Prediction (SPP) is crucial in finance, but due to market volatility, conventional methods, such as statistical, technical, and fundamental analyses, often lack the depth required for accurate forecasting [4]. The dynamic nature of the stock market and the intricate temporal patterns make it a difficult time series to analyze [5]. Dynamic trading causes fluctuations in stock prices as supply and demand forces interact [6]. In addition, stock prices are subject to several unpredictable factors, causing their time series to be inherently unstable and lacking a constant statistical trend [7]. Thus, SPP is considered the most difficult issue in all types of prediction tasks [8]. Recent studies on SPP focus on improving prediction models and Feature Selection (FS) [9]. Conventional econometric methods struggle with external factors, resulting in the use of Machine Learning (ML) and Deep Learning (DL) methods, with the latter showing superior performance [10].

The study in [11] examined the ability of hybrid quantum-classical methods to enhance feature learning. Dual hybrid optimizer models were used. Initially, conventional recurrent methods, such as LSTM and RNN, captured temporal relations before quantum processing, followed by a joint learning model. In [12], a DL-based model fused a CNN with LSTM. NLP was utilized for feature extraction. CNN handles local feature extraction, while LSTM models time-series data for sentiment prediction. In [13], ML models were used to improve SPP for retail investors. In [14], a dung beetle optimizer with DL was proposed to solve inverse problems in the prediction of financial futures (DBODL-SIPPFF). In addition, the DC optimizer was used to overcome inverse problems, and a ConvLSTM was utilized for SPP. In [15], a hybrid ML approach was used for Long-Term Stock Market price trend prediction (LT-SMF). Polarization and scaling were utilized to identify useful qualities, and a Brown Planthopper Optimizer (BPO) method decreased data dimensionality for optimum FS. In [16], an innovative hybrid approach, SA-DLSTM, employed a Denoising Autoencoder (DAE), ECNN, and LSTM. Primarily, user comments on the Internet were employed, and ECNNs were used to capture sentimental representations.

This study presents a Financial Time Series Forecasting Using Quasi-Recurrent Neural Network and Crown Porcupine Optimizer (FTSF-QRNNCPO) method. The key contributions of the proposed technique are as follows:

- Performs effective data preprocessing, including missing value handling, data cleaning, and normalization. This ensures high-quality input data and mitigates noise, contributing to more accurate and stable stock market predictions.
- Employs the SBOA technique for choosing the most relevant features. This mitigates dimensionality and removes redundant data, improving model accuracy and computational efficiency in SPP.
- Utilizes the QRNN to effectively model and predict stock market risk, capturing temporal dependencies with a mitigated computational cost. This improves prediction speed and reliability in dynamic financial environments.
- Employs the CPO for fine-tuning the hyperparameters of the QRNN model for optimal performance, balancing exploration and exploitation during tuning. This results in improved convergence speed and overall prediction accuracy.
- The integration of SBOA-based FS and CPO-based tuning within the QRNN framework introduces a novel hybrid approach for stock risk prediction, improving both feature relevance and optimization. This is a unique synergy that addresses key challenges in financial forecasting with improved accuracy.

## II. PROPOSED METHODOLOGY

The aim of this study was to develop an intelligent framework for financial stock market price prediction. The proposed method comprises four stages: data processing, feature reduction, prediction, and optimization. Figure 1 represents the flow of the FTSF-QRNNCPO technique.

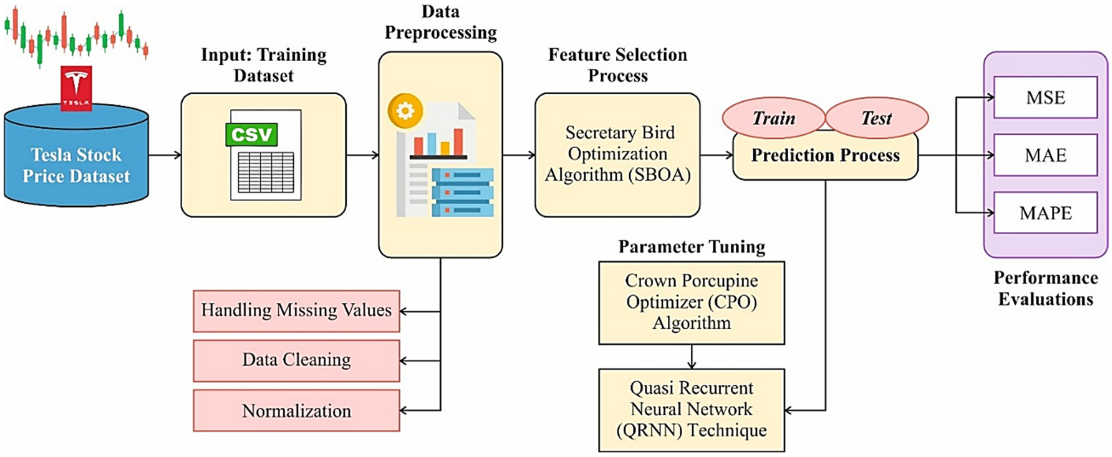


Fig. 1. Workflow of the FTSF-QRNNCPO approach.

## *A. Stage I: Input Data Processing*

Initially, the data preprocessing stage consists of various levels, such as handling missing values, data cleaning, and normalization, for transforming the input data into a beneficial format [17]. The preprocessing stage prepares raw data for analysis through key steps: handling missing values via imputation to maintain data integrity, cleaning to remove outliers and errors, and normalization to scale features for balanced contribution during training. Min-max scaling is used to normalize the data. Each stock price $x$ was converted to a normalized value $x' \in [0,1]$ using:

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

This normalization is crucial for neural network convergence and to prevent high-priced stocks from disproportionately influencing learning.

## *B. Stage II: SBOA-Based Feature Reduction*

Afterward, SBOA is used to choose the optimal features [18]. This model was chosen for its robust balance between exploration and exploitation, assisting in avoiding local optima. Its adaptive search and efficient convergence make it appropriate for handling high-dimensional financial data. This enhances feature relevance while reducing computational cost. SBOA is derived from the Secretary Bird's (SB) behaviors and applies two major strategies. The initial tactic focuses on foraging and includes three stages: consuming, searching for, and attacking prey. The second tactic focuses on evading hunters through evasion and camouflage stages.

### *1) Exploration Strategy*

The complete predation process is divided into three equivalent time intervals, $< \frac{1}{3}T$, $\frac{1}{3}T \le t < \frac{2}{3}T$, and $\frac{2}{3}T \le t \le T$, where $t$ and $T$ denote the present and complete iteration count. The method starts with differential evolution for global search, uses historic top locations to guide optimization, and applies Levy Flight (LF) to improve global exploration.

$$x_{i,j}^{new} = \begin{cases} x_{i,j} + \left(x_{random_1} - x_{random_2}\right) \times R_1, \quad t < \frac{1}{3}T \\ x_{best} + \exp\left((t/T) \wedge 4\right) \times (RB - 0.5) \times (x_{best} - x_{i,j})^1, \frac{1}{3}T \le t < \frac{2}{3}T \\ x_{best} + \left(\left(1 - \frac{t}{T}\right) \wedge \left(2 \times \frac{t}{T}\right)\right) \times x_{i,j} \times RL, \quad \frac{2}{3}T \le t \le T \end{cases}$$

where $t$ is the current iteration, $T$ is the maximum iteration, $x_{random_1}$ and $x_{random_2}$ are random candidate solutions, $R_1$ is a $1 \times Dim$ random array in (0,1), $Dim$ is the solution dimension, $x_{i,j}^{new}$ is the $j$-th dimension value, $RB$ is a $1 \times Dim$ normal distribution array (mean 0, std 1), $x_{best}$ is the current best solution, and $RL$ is the LF function times 0.5.

### *2) Exploitation Strategy*

When SBs detect a predator, they seek a safe hiding spot; if none is available, they flee by flying or running. A dynamic perturbation factor $(1 - \frac{t}{T})^2$ improves global search early and local search later. The two evasion tactics of SBs are then applied accordingly.

$$x_{i,j}^{new} = \begin{cases} x_{best} + (2 \times RB - 1) \times (1 - \frac{t}{T})^2 \times x_{i,j}, if\ rand < r \\ x_{i,j} + R_2 \times \left(x_{random} - K \times x_{i,j}\right), else \end{cases} \quad (3)$$

where $r = 0.5$, $R_2$ signifies a random array of dimensions $1 \times Dim$ from the typical distribution, $x_{random}$ signifies random candidate solutions for the present iteration, and $K$ symbolizes an arbitrary selected number of 1 or 2.

The Fitness Function (FF) balances minimizing feature count with maximizing classification accuracy. Equation (4) depicts the FF to estimate solutions.

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|C|} \quad (4)$$

where $\gamma_R(D)$ epitomizes the classifier error rate. $|R|$ signifies cardinality of the designated subset, $|C|$ denotes complete feature counts in the dataset, and $\alpha$ and $\beta$ are dual parameters corresponding to the implication of subset length and classification quality, belonging to $[1, 0]$ and $\beta = 1 - \alpha$.

### C. Stage III: Prediction Using the QRNN Model

The proposed FTSF-QRNNCPO model employs the QRNN method for the prediction process [19]. This model is chosen for its fast training and efficient capture of temporal patterns in stock data. It integrates convolutional and recurrent layers, presenting better speed and accuracy than conventional RNNs or LSTMs. QRNN also handles long-term dependencies well. Due to the benefits of RNN in sequential data processing, several investigators employed its general versions, GRU, and LSTM, to the residual task of time prediction, and attained improved prediction precision than conventional techniques. Table I summarizes the layer-wise architecture of the FTSF-QRNNCPO model using QRNN.

TABLE I. LAYER-WISE CONFIGURATION OF THE FTSF-QRNNCPO MODEL USING QRNN

| Layer | Type | Details |
|---|---|---|
| Input | - | TIME_STEPS = 30, CHOSEN_FEATURE via SBOA |
| Layer 1 | Conv1D | FILTERS = 64, KERNEL_SIZE = 5, ACTIVATION = ReLU |
| Layer 2 | QRNN | UNITS = 128, WINDOW_SIZE = 2, ACTIVATION = Tanh |
| Layer 3 | Max Pooling 1D | POOL_SIZE = 2 |
| Layer 4 | Dropout | RATE = 0.3 |
| Layer 5 | Dense | UNITS = 64, ACTIVATION = ReLU |
| Layer 6 | Dropout | RATE = 0.2 |
| Output layer | Dense | UNITS = 1, ACTIVATION = Sigmoid |

LSTM can resolve RNN gradient explosion and gradient disappearance problems. The LSTM neuron outputs:

$$(h_t, C_t) = LSTM(x_t, h_{t-1}, C_{t-1}) \tag{5}$$

where $x_t$ denotes the input at time $t$, and $C_{t-1}$, $C_t$, $h_{t-1}$, $h_t$ characterize the state of the cell and output at time $t$ and $t-1$.

GRU simplifies LSTM by combining forget, output, and input gates into update and reset gates, retaining LSTM features while speeding up training. A GRU neuron output is computed as:

$$h_t = GRU(x_t, h_{t-1}) \tag{6}$$

LSTM and GRU rely on previous outputs, causing longer computation and issues with long sequences, using convolution and pooling layers. A sequence data $X = (x_1, x_2, \dots, x_T)$ applied as the input to the layer of convolution is transformed into a sequence $Z = (z_1, z_2, \dots, z_T)$. Once the filter width $d$ in the layer of convolution is transformed to the variable parameter $h$, the width of $z_t$ gained after providing the sequence data into the layer of convolutional is $x_{t-h+1} \sim x_t$. The RNN can be described as:

$$(h_t, C_t) = QRNN(X, C_{t-1}) \tag{7}$$

### D. Stage IV: Parameter Tuning Process

The CPO method [20] is used for parameter tuning. This method is chosen for its robust balance between exploration and exploitation, helping avoid local optima during parameter tuning. Its adaptive search strategy ensures efficient and thorough optimization. This improves performance and speeds convergence, making CPO ideal for tuning the QRNN. Inspired by the CP's four defense tactics, it models escalating aggression in four zones, starting with a collection of primary individuals and gradually reducing their number over iterations as part of the search process.

$$\overrightarrow{X_i} = \vec{L} + \rightarrow r \times (\vec{U} - \vec{L}) | i = 1,2,3, \ldots, N \tag{8}$$

where $N$ denotes the individual counts, for example, the size of the population, $\overrightarrow{x_i}$ denotes the $i$-th candidate solution in the searching region, $\vec{U}$ and $\vec{L}$ denote the upper and lower limits of the searching range, and $\vec{r}$ denotes an arbitrarily initialized vector in [0, 1]. Particular CPs are found from the population in the optimizer procedure for accelerating the convergence speed, and then reintroduced into the population.

$$N = N_{\min} + (N' - N_{\min}) * \left(1 - \left(\frac{t\% \frac{T_{\max}}{T}}{\frac{T_{\max}}{T}}\right)\right) \tag{9}$$

where $N'$ and $N$ refer to the first and present population size, $N_{\min}$ denotes the minimal number of individuals in the recently generated population, $t$ and $T_{\max}$ means present and maximal quantity of function evaluations, $T$ signifies iteration counts, and % stands for the modulo operator.

The CPO models four defensive tactics: (i) midpoint between present and best positions are used for randomly approaching or retreating; (ii) noise is generated to scare predators, updating its position using other random individuals with weighted influence; (iii) foul-smelling gases are emitted to create diffusion zones, adjusting its position through odor diffusion, direction, and defense coefficients; and (iv) attacks using short feathers if a predator approaches, updating position based on the top solution and average impact force. These tactics integrate randomness, adaptive parameters, and population dynamics to guide the optimization process.

The CPO is applied to define the intricate parameter in the QRNN technique. The MSE is used for the objective function, calculated as:

$$MSE = \frac{1}{T}\sum_{j=1}^{L}\sum_{i=1}^{M}\left(y_j^i - d_j^i\right)^2 \tag{18}$$

where $L$ and $M$ denote the resulting data value and layer, correspondingly, and $y_j^i$ and $d_j^i$ indicate the appropriate and attained sizes for the $j$-th element from the resulting layer of the system in time $t$, correspondingly.

## III. EXPERIMENTAL RESULTS AND ANALYSIS

The FTSF-QRNNCPO model was experimentally tested on the Tesla stock price dataset [21]. This dataset spans from January 29, 2010, to March 24, 2022, comprising 2,957 daily records with seven features: Date, Open, High, Low, Close, Adjusted Close, and Volume, with all units in USD. The dataset's frequency is daily, and after preprocessing, it includes 2,957 rows and 7 columns. The proposed method was developed using Python 3.6.5 on an i5-8600k CPU, 4GB GPU, 16GB RAM, 250GB SSD, and 1TB HDD, using a 0.01 learning rate, ReLU, 50 epochs, 0.5 dropout, and a batch size of 5.

Table II illustrates the results of the FTSF-QRNNCPO model with different metrics in terms of the training (TRAST) and testing (TESST) sets [22-24]. On TRAST, the FTSF-QRNNCPO method scored MSE, MAE, MAPE, and $R^2$ of 918.289, 26.398, 0.411, and 0.8511, respectively. On TESST, the MSE, MAE, MAPE, and $R^2$ scores were 909.036, 26.256, 0.415, and 0.8945, respectively. Figure 2 presents a collection of distribution plots (density plots with histograms) for five stock market-relevant features, such as open, high, low, close, and volume. These plots are normally applied in financial data preprocessing and investigative data analysis to determine the basic distribution of the data variables.

TABLE II. TRAST AND TESST OUTCOME OF THE FTSF-QRNNCPO TECHNIQUE WITH DISTINCT METRICS

| Metrics | TRAST | TESST |
|---|---|---|
| MSE | 918.289 | 909.036 |
| MAE | 26.398 | 26.256 |
| MAPE | 0.411 | 0.415 |
| $R^2$ | 0.8511 | 0.8945 |

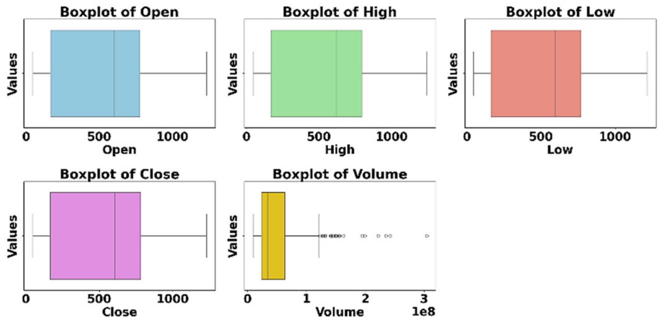


Fig. 2. Distribution plots.

Table III compares FTSF-QRNNCPO with existing techniques. In MSE, FTSF-QRNNCPO achieved a lower MSE of 909.036, while existing methods such as GRU, LSTM, RNN, AM-LSTM, CNN-BiLSTM, XGBoost, and LSTM+GRU had MSEs of 913.606, 913.046, 912.406, 911.616, 911.016, 910.296, and 909.766, respectively. In MAE, FTSF-QRNNCPO scored the least MAE and MAPE of 26.256 and 0.415, with GRU, LSTM, RNN, AM-LSTM, CNN-BiLSTM, XGBoost, and LSTM+GRU obtaining higher MAE at 30.876, 30.206, 29.636, 28.836, 28.276, 27.526, and 27.006, and MAPE at 0.865, 0.803, 0.729, 0.664, 0.592, 0.516, and 0.465, respectively. Finally, in $R^2$, FTSF-QRNNCPO attained a moderate value of 0.8945.

TABLE III. COMPARATIVE ANALYSIS OF FTSF-QRNNCPO WITH EXISTING TECHNIQUES

| Model | MSE | MAE | MAPE | $R^2$ |
|---|---|---|---|---|
| GRU [22] | 913.606 | 30.876 | 0.865 | 0.5998 |
| LSTM [22] | 913.046 | 30.206 | 0.803 | 0.7482 |
| RNN [23] | 912.406 | 29.636 | 0.729 | 0.5244 |
| AM-LSTM [23] | 911.616 | 28.836 | 0.664 | 0.5052 |
| CNN-BiLSTM [23] | 911.016 | 28.276 | 0.592 | 0.5185 |
| XGBoost [24] | 910.296 | 27.526 | 0.516 | 0.7232 |
| LSTM+GRU [24] | 909.766 | 27.006 | 0.465 | 0.5381 |
| FTSF-QRNNCPO (Proposed) | 909.036 | 26.256 | 0.415 | 0.8945 |

Table IV shows the results of an ablation study of FTSF-QRNNCPO. Starting with the base QRNN, performance improves, and integrating the SBOA with QRNN mitigates MSE, MAE, MAPE, and $R^2$ scores, indicating the efficiency of optimal FS. Further improvement is observed with the inclusion of the CPO, which tunes the model parameters to achieve better accuracy. The FTSF-QRNNCPO model achieves the lowest error values across all metrics, highlighting the synergistic benefit of incorporating FS and parameter optimization.

TABLE IV. ABLATION STUDY RESULTS OF THE FTSF-QRNNCPO METHOD

| Method | MSE | MAE | MAPE | $R^2$ |
|---|---|---|---|---|
| QRNN | 929.136 | 28.076 | 0.688 | 0.8763 |
| QRNN+SBOA (With FS without parameter tuning) | 922.436 | 27.476 | 0.593 | 0.8813 |
| QRNN+CPO (Without FS with parameter tuning) | 916.436 | 26.936 | 0.498 | 0.8872 |
| FTSF-QRNNCPO (With FS and parameter tuning) | 909.036 | 26.256 | 0.415 | 0.8945 |

## IV. CONCLUSION

This study presented the FTSF-QRNNCPO method. Initially, the data preprocessing stage involved various tasks, such as handling missing values, data cleaning, and normalization, to transform the input data into a beneficial pattern. SBOA was utilized to choose the optimal features. In addition, the QRNN method was employed for prediction. Finally, the CPO method was implemented to improve the performance of the prediction. The FTSF-QRNNCPO approach, validated on Tesla stock data, achieved a lowest MAPE of 0.415%, outperforming previous models. The limitations of this study comprise its reliance on a single stock dataset, which may restrict generalizability across broader markets. Additionally, external factors such as news sentiment or macroeconomic indicators were not incorporated. Future work can explore multi-stock datasets and integrate external data sources to enhance prediction robustness.

## REFERENCES

[1] M. R. Islam and N. Nguyen, "Comparison of Financial Models for Stock Price Prediction," *Journal of Risk and Financial Management*, vol. 13, no. 8, Aug. 2020, Art. no. 181, https://doi.org/10.3390/jrfm13080181.

[2] Z. Hu, Y. Zhao, and M. Khushi, "A Survey of Forex and Stock Price Prediction Using Deep Learning," *Applied System Innovation*, vol. 4, no. 1, Mar. 2021, Art. no. 9, https://doi.org/10.3390/asi4010009.

[3] P. Gao, R. Zhang, and X. Yang, "The Application of Stock Index Price Prediction with Neural Network," *Mathematical and Computational Applications*, vol. 25, no. 3, Sep. 2020, Art. no. 53, https://doi.org/10.3390/mca25030053.

[4] X. Ji, J. Wang, and Z. Yan, "A stock price prediction method based on deep learning technology," *International Journal of Crowd Science*, vol. 5, no. 1, pp. 55–72, Apr. 2021, https://doi.org/10.1108/IJCS-05-2020-0012.

[5] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *The Journal of Supercomputing*, vol. 76, no. 3, pp. 2098–2118, Mar. 2020, https://doi.org/10.1007/s11227-017-2228-y.

[6] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," *Procedia*

*Computer Science*, vol. 167, pp. 599–606, Jan. 2020, https://doi.org/10.1016/j.procs.2020.03.326.

[7] K. Zhang, G. Zhong, J. Dong, S. Wang, and Y. Wang, "Stock Market Prediction Based on Generative Adversarial Network," *Procedia Computer Science*, vol. 147, pp. 400–406, Jan. 2019, https://doi.org/10.1016/j.procs.2019.01.256.

[8] T. S. Mian, "Evaluation of Stock Closing Prices using Transformer Learning," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11635–11642, Oct. 2023, https://doi.org/10.48084/etasr.6017.

[9] U. P. Gurav and S. Kotrappa, "Sentiment Aware Stock Price Forecasting using an SA-RNN-LBL Learning Model," *Engineering, Technology & Applied Science Research*, vol. 10, no. 5, pp. 6356–6361, Oct. 2020, https://doi.org/10.48084/etasr.3805.

[10] B. Y. Dwiandiyanta, R. Hartanto, and R. Ferdiana, "Optimization of Stock Predictions on Indonesia Stock Exchange: A New Hybrid Deep Learning Method," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19370–19379, Feb. 2025, https://doi.org/10.48084/etasr.9363.

[11] P. K. Choudhary, N. Innan, M. Shafique, and R. Singh, "HQNN-FSP: A Hybrid Classical-Quantum Neural Network for Regression-Based Financial Stock Market Prediction." arXiv, Mar. 19, 2025, https://doi.org/10.48550/arXiv.2503.15403.

[12] W. Yang, Y. Lin, H. Xue, and J. Wang, "Research on stock market sentiment analysis and prediction method based on convolutional neural network," in *Proceedings of the 2025 International Conference on Machine Learning and Neural Networks*, May 2025, pp. 91–96, https://doi.org/10.1145/3747227.3747241.

[13] M. Sui, C. Zhang, L. Zhou, S. Liao, and C. Wei, "An Ensemble Approach to Stock Price Prediction Using Deep Learning and Time Series Models," in *2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Shenyang, China, Jul. 2024, pp. 793–797, https://doi.org/10.1109/ICPICS62053.2024.10796661.

[14] H. Alnafisah, H. Abdulrahim, A. A. Hassaballa, A. Alsulami, and Adil. O. Y. Mohamed, "Dung beetle optimization with deep learning approach for solving inverse problems in predicting financial futures," *Alexandria Engineering Journal*, vol. 109, pp. 71–82, Dec. 2024, https://doi.org/10.1016/j.aej.2024.08.082.

[15] K. Venkateswararao and B. V. R. Reddy, "LT-SMF: long term stock market price trend prediction using optimal hybrid machine learning technique," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5365–5402, Jun. 2023, https://doi.org/10.1007/s10462-022-10291-5.

[16] Y. Zhao and G. Yang, "Deep Learning-based Integrated Framework for stock price movement prediction," *Applied Soft Computing*, vol. 133, Jan. 2023, Art. no. 109921, https://doi.org/10.1016/j.asoc.2022.109921.

[17] I. E. Fattoh, M. El Maghawry Ibrahim, and F. A. Mousa, "Unveiling market dynamics: a machine and deep learning approach to Egyptian stock prediction," *Future Business Journal*, vol. 11, no. 1, Feb. 2025, Art. no. 18, https://doi.org/10.1186/s43093-025-00421-0.

[18] R. Tang *et al.*, "Three-Dimensional Path Planning for AUVs Based on Interval Multi-Objective Secretary Bird Optimization Algorithm," *Symmetry*, vol. 17, no. 7, Jul. 2025, Art. no. 993, https://doi.org/10.3390/sym17070993.

[19] C. Liu *et al.*, "Comparative evaluation of encoding techniques for workflow process remaining time prediction for cloud applications," *Journal of Cloud Computing*, vol. 14, no. 1, Jul. 2025, Art. no. 36, https://doi.org/10.1186/s13677-025-00763-8.

[20] Z. Zhou, H. Li, K. Wu, J. Chen, T. Yao, and Y. Wu, "Development of Time Series Models and Algorithms: Creep Prediction for Low-Carbon Concrete Materials," *Materials*, vol. 18, no. 13, Jan. 2025, Art. no. 3152, https://doi.org/10.3390/ma18133152.

[21] "Tesla Stock Price." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/jillanisofttech/tesla-stock-price.

[22] V. Gkonis and I. Tsakalos, "A hybrid optimized deep learning model via the Golden Jackal Optimizer for accurate stock price forecasting," *Expert Systems with Applications*, vol. 278, Jun. 2025, Art. no. 127287, https://doi.org/10.1016/j.eswa.2025.127287.

[23] J. L. Li and W. K. Shi, "Hybrid preprocessing for neural network-based stock price prediction," *Heliyon*, vol. 10, no. 24, Dec. 2024, https://doi.org/10.1016/j.heliyon.2024.e40819.

[24] D. M. Teixeira and R. S. Barbosa, "Stock Price Prediction in the Financial Market Using Machine Learning Models," *Computation*, vol. 13, no. 1, Jan. 2025, Art. no. 3, https://doi.org/10.3390/computation13010003.