

# Enhancing Stock Price Forecasting Accuracy Using Compositional Rnn Learning Framework With Grey Wolf Optimization

Bharani K<sup>1</sup>, KrishnaKumar R<sup>2</sup>, Surendharan R<sup>3</sup>, Dalphin Mary F<sup>4</sup>

<sup>1, 2, 3, 4</sup>Dept of AI & Data Science

<sup>4</sup> Assistant Professor, Jeppiaar University

<sup>1, 2, 3, 4</sup> Jeppiaar University, Chennai, India

**Abstract-** Predicting stock prices accurately is essential for making well-informed decisions in erratic financial markets. This paper introduces a compositional deep learning framework for multivariate time-series forecasting that integrates three RNN variants: LSTM, GRU, and SRU. Grey Wolf Optimizer (GWO) and Random Search (RS) were used to develop and optimize a total of 54 model architectures. The best results were obtained by LSTM-GWO (1-1-0-1), with  $R^2 = 99.2427\%$ ,  $MAPE = 1.1721\%$ ,  $RMSE = 339.3902$ ,  $WI = 0.9981$ ,  $NSE = 0.9924$ , and minimal bias ( $PBIAS = 0.0523$ ). Additionally, GRU-GWO and SRU-GWO performed better than RS-based models, demonstrating the efficacy of metaheuristic optimization. The results show that GWO and systematic architectural design greatly improve forecasting accuracy and model stability for reliable financial prediction systems.

**Keywords:** Stock Price Forecasting; Recurrent Neural Networks; LSTM; GRU; SRU; Grey Wolf Optimizer; Compositional Deep Learning; Metaheuristic Optimization; Time-Series Prediction.

## I. INTRODUCTION

Stock markets are inherently volatile, with prices fluctuating daily due to economic, political, and behavioral factors. This volatility creates both investment opportunities and significant risks, making accurate stock price forecasting essential for informed and risk-aware decision-making. Classical investment theory emphasizes the importance of predictive modeling for portfolio management and risk mitigation [1, 2].

Traditional statistical approaches, particularly Autoregressive Integrated Moving Average (ARIMA) and ARIMA-GARCH models, have been widely used to model short-term linear and non-stationary patterns in financial time series [7, 9, 10]. Similarly, conventional machine learning methods such as Support Vector Regression (SVR), Random

Forest (RF), and K-Nearest Neighbors (KNN) have demonstrated effectiveness in stock prediction tasks [13–15].

However, the inherently stochastic and nonlinear nature of financial markets limits the ability of these models to capture complex multivariate dependencies and long-term temporal relationships. Deep learning techniques, especially Recurrent Neural Network (RNN) variants such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Simple Recurrent Unit (SRU), have shown superior capability in modeling sequential financial data [4, 20]. These architectures effectively capture long-term dependencies and nonlinear patterns, leading to improved predictive performance. Despite these advancements, most existing studies rely on single, monolithic RNN architectures that process multivariate inputs directly, often overlooking feature-specific temporal characteristics and structured architectural design.

To address these limitations, this study proposes a compositional RNN learning framework that systematically integrates multiple RNN variants within a unified forecasting architecture. The proposed approach introduces feature-wise univariate encoding for Open, High, Low, Close, and Volume (OHLCV) data, followed by structured concatenation-based fusion and dual-dropout regularization to enhance generalization. Furthermore, 54 stacked architectural configurations are systematically explored and optimized using Random Search (RS) and the Grey Wolf Optimizer (GWO).

The framework is empirically validated using Hang Seng Index (HSI) data, demonstrating significant improvements across multiple performance metrics. By combining compositional architecture design with metaheuristic optimization, this study presents a modular, scalable, and high-performance alternative to traditional monolithic RNN models for robust stock price forecasting.

## II. RELATED WORK

Stock price forecasting has evolved significantly, beginning with statistical models such as ARIMA and ARIMA-GARCH, which effectively capture short-term linear and non-stationary patterns in financial time-series data [7–10].

Several studies demonstrated ARIMA's reliability in predicting stock trends across various markets. However, its limitation in modeling complex nonlinear dependencies led researchers to explore machine learning approaches.

Traditional machine learning techniques, including Support Vector Machines (SVM), Random Forest (RF), Gradient Boosting Machines (GBM), and ensemble methods, have been widely applied to stock prediction [12–15]. These models improved classification and regression performance compared to purely statistical approaches. Nevertheless, their ability to model long-term temporal dependencies remains limited, particularly when handling high-dimensional and highly volatile financial data.

The emergence of deep learning, particularly Recurrent Neural Networks (RNNs), marked a significant advancement in time-series forecasting [4, 5]. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures demonstrated superior capability in capturing long-term dependencies and complex temporal dynamics [20]. Subsequent research introduced regularization strategies, such as dropout, and optimization improvements to enhance generalization and reduce overfitting.

Recent developments have focused on hybrid and attention-based architectures, combining CNNs, LSTMs, GRUs, transformers, and decomposition techniques to improve prediction accuracy [3, 6]. These models effectively capture nonlinear patterns, seasonal trends, and feature interactions. Additionally, metaheuristic and advanced hyperparameter optimization methods—including Bayesian optimization, Hyperband, and nature-inspired algorithms—have been employed to further enhance performance and training efficiency.

Despite these advancements, most studies rely on monolithic architectures and direct multivariate inputs, with limited exploration of structured compositional designs and feature-wise encoding strategies. This gap highlights the need for a modular and systematically optimized RNN framework capable of leveraging complementary temporal dynamics while improving stability and forecasting accuracy.

## III. MATERIALS AND METHODS

### 1. Research Framework

This study adopts a structured and modular experimental framework designed to ensure methodological rigor and reproducibility. The workflow consists of five main stages: data acquisition, data preprocessing, compositional RNN architecture design, hyperparameter optimization, and performance evaluation. Each stage is independently defined yet systematically integrated to maintain consistency between theoretical modeling and empirical validation.

The framework emphasizes feature-wise sequence modeling, structured architectural composition, and metaheuristic-based hyperparameter tuning. By separating encoding, fusion, and optimization components, the proposed design allows controlled experimentation across multiple RNN variants (LSTM, GRU, and SRU) and architectural configurations.

### 2. Data Collection

The dataset used in this study consists of daily historical price data obtained from Yahoo Finance using the YFinance Python library.

Five primary market features—Open, High, Low, Close, and Volume (OHLCV)—were selected as input variables. These features represent essential financial indicators that capture price dynamics and market trading activity. Unlike studies that rely on technical indicators or engineered features, this research intentionally utilizes raw OHLCV data to preserve the inherent temporal structure of financial sequences and allow the RNN models to learn patterns directly from primary signals.

Descriptive statistical analysis was conducted to examine the distribution, scale, and variability of each feature, ensuring appropriate normalization and stable training behavior.

### 3. Data Preprocessing

To improve numerical stability and accelerate convergence during training, Min-Max normalization was applied to scale all input features into the range [0, 0.95]. This scaling prevents feature dominance due to magnitude differences (e.g., volume versus price) and reduces boundary saturation issues in activation functions.

Sequential data segmentation was performed to transform the dataset into supervised learning format suitable for RNN models. Two window sizes—20 trading days (short-term dependency) and 40 trading days (medium-term dependency)—were used to construct input sequences. Each sequence was formatted into a three-dimensional tensor structure (samples, time steps, features).

To preserve temporal integrity and avoid data leakage, the dataset was split chronologically into 80% training data (4,678 samples) and 20% testing data (1,170 samples). This ensures that future information does not influence past predictions.

#### 4. Optimization Strategy

Hyperparameter optimization plays a crucial role in enhancing deep learning model performance. In this study, two optimization techniques were employed: Random Search (RS) and Grey Wolf Optimizer (GWO). The optimized parameters included the number of RNN units, learning rate, batch size, number of epochs, and dropout rates.

Random Search explores the hyperparameter space stochastically, while GWO, a nature-inspired metaheuristic algorithm, balances exploration and exploitation to identify near-optimal parameter combinations efficiently.

#### 5. Performance Evaluation

Model performance was evaluated using multiple statistical metrics, including  $R^2$ , Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Root Mean Square Percentage Error (RMSPE), Percent Bias (PBIAS), Willmott Index (WI), and Nash–Sutcliffe Efficiency (NSE). These metrics collectively assess prediction accuracy, error magnitude, bias, agreement, and robustness.

#### 6. Compositional RNN Architecture

The proposed architecture follows a feature-wise compositional design. Each OHLCV feature is independently encoded using stacked RNN layers (LSTM, GRU, or SRU) to preserve unique temporal characteristics. Dropout regularization is applied before concatenation to enhance generalization.

Encoded feature representations are then concatenated to form a unified multivariate representation. Optional post-concatenation RNN layers further refine the temporal fusion. Finally, dense layers generate the predicted stock price output.

A total of 18 architectural configurations per RNN variant were systematically constructed and evaluated, enabling comprehensive exploration of compositional structures and their impact on forecasting performance.

### IV. RESULTS AND DISCUSSION

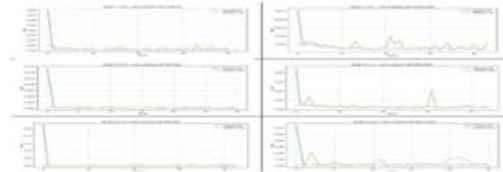


Figure 1: Proposed multi-variant compositional RNN architecture for stock price forecasting.

#### 1. Hyperparameter Optimization Performance

The tuning results show that GWO consistently achieved lower validation loss compared to RS across all RNN variants. This confirms the effectiveness of GWO in exploring the hyperparameter space more efficiently. The optimized parameters included units, dropout rate, learning rate, batch size, and epochs.

Among all configurations, the LSTM-GWO (1-1-0-1) architecture emerged as the best-performing model, achieving:

- $R^2 = 99.2427\%$
- $RMSE = 339.3902$
- $MAPE = 1.1721\%$
- $RMSPE = 1.6221\%$
- $WI = 0.9981$
- $NSE = 0.9924$
- $PBIAS = 0.0523$

These metrics indicate extremely high explanatory power, minimal prediction error, strong agreement between observed and predicted values, and negligible bias.

Algorithm	Optimizer	Model	R-square	RMSE	MAPE	RMSPE	WI	NSE	PBIAS
LSTM	RS	1-1-0-0	99.2329	341.5754	1.1841	1.6307	0.9981	0.9923	0.0349
LSTM	GWO	1-1-0-1	99.2427	339.3902	1.1721	1.6221	0.9981	0.9924	0.0523
GRU	RS	2-2-0-0	99.2156	345.4072	1.1809	1.6473	0.9980	0.9922	-0.2154
GRU	GWO	2-2-1-1	99.2322	341.7225	1.1821	1.6197	0.9981	0.9923	-0.1357
SRU	RS	2-2-1-0	99.0165	386.7643	1.365	1.8447	0.9976	0.9902	0.3276
SRU	GWO	2-2-0-0	99.2009	348.6384	1.208	1.6405	0.998	0.992	-0.1794

Figure 2: Comparative performance metrics of LSTM, GRU, and SRU models under RS and GWO optimization strategies.

#### 2. Comparative Performance of RNN Variants

**LSTM Models:** LSTM models achieved the highest stability and accuracy. Both RS and GWO produced strong results, but GWO slightly improved  $R^2$  and reduced RMSE and MAPE. The narrow distribution of errors and minimal bias

confirm that LSTM effectively captures long-term dependencies in financial time-series data. The LSTM loss curves also showed smooth convergence with minimal overfitting, demonstrating strong generalization ability.

**GRU Models:** GRU models performed competitively, with GRU-GWO (2-1-1-1) achieving:

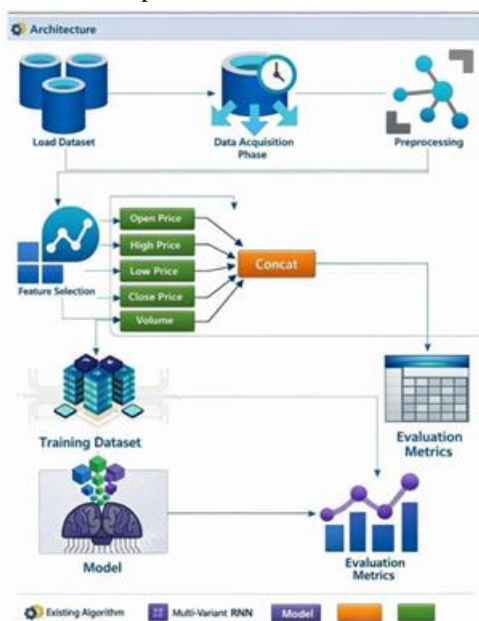
- $R^2 = 99.2322\%$
- $RMSE = 341.7225$
- $MAPE = 1.1821\%$

Although slightly behind LSTM, GRU demonstrated strong predictive capability with marginal underestimation bias ( $PBIAS = -0.1357$ ). GWO again improved stability and reduced variability compared to RS. GRU’s simpler gating structure allowed efficient computation while maintaining high forecasting accuracy.

**SRU Models:** SRU models showed acceptable but comparatively lower performance. The best configuration, SRU-GWO (2-2-0-0), achieved:

- $R^2 = 99.2009\%$
- $RMSE = 348.6384$
- $MAPE = 1.2080\%$

Although still highly accurate, SRU exhibited greater variability and sensitivity to data fluctuations. The wider distribution of errors and higher standard deviation indicate lower robustness compared to LSTM and GRU.



**Figure 3:** Training and validation loss curves for optimized LSTM, GRU, SRU models.

### 3. Statistical and Visual Analysis

The statistical summary (mean, standard deviation, skewness, kurtosis) reinforces that:

- LSTM-GWO has the most stable and tightly clustered performance distribution.
- GRU-GWO performs competitively with moderate variability.
- SRU models show higher dispersion and occasional extreme deviations.

Visualization techniques such as box plots, violin plots, radar plots (PBIAS), Taylor diagrams (WI and NSE), and loss curves consistently confirm that GWO optimization improves accuracy, reduces bias, and enhances generalization.

### 4. Overall Findings

The results confirm three major conclusions:

1. Compositional architecture design improves forecasting accuracy by modeling each feature independently before fusion.
2. Grey Wolf Optimizer outperforms Random Search, consistently reducing prediction error and bias.
3. LSTM-GWO (1-1-0-1) is the most accurate and stable model, making it the best candidate for stock price forecasting in this study.

In summary, the results section strongly validates the effectiveness of the proposed multi-variant compositional RNN framework. The integration of feature-wise encoding, structured concatenation, dual dropout regularization, and metaheuristic optimization significantly enhances stock forecasting performance in nonlinear and volatile financial markets.

## V. CONCLUSION

This study proposed a compositional deep learning framework for enhancing stock price forecasting accuracy using multi-variant Recurrent Neural Network (RNN) architectures. By systematically constructing and evaluating stacked configurations across LSTM, GRU, and SRU models, the research demonstrated that architectural composition combined with metaheuristic optimization significantly improves predictive performance in multivariate financial time-series data.

The proposed framework introduced feature-wise uni-variate RNN encoding, where each OHLCV feature was modeled independently before structured concatenation. This design preserves individual temporal dynamics while enabling effective feature fusion. Additionally, the implementation of dual dropout regularization before and after concatenation enhanced generalization capability and reduced overfitting. The systematic exploration of architectural combinations further allowed the identification of optimal layer compositions for each RNN variant.

Experimental results using Hang Seng Index (HSI) data confirmed the effectiveness of the proposed approach. Among all evaluated configurations, the LSTM-GWO (1-1-0-1) model achieved the best performance, with an  $R^2$  value of 99.2427%, RMSE of 339.3902, MAPE of 1.1721%, RMSE of 1.6221%, WI of 0.9981, NSE of 0.9924, and minimal bias ( $PBIAS = 0.0523$ ). These results indicate high explanatory power, low prediction error, strong agreement with observed values, and negligible systematic bias. The GRU-GWO model demonstrated competitive performance with slightly lower accuracy but improved computational efficiency. Although SRU models achieved acceptable forecasting results, they exhibited higher variability and lower robustness compared to LSTM and GRU.

Furthermore, the Grey Wolf Optimizer consistently outperformed Random Search in hyperparameter tuning across all RNN variants. GWO achieved lower validation loss, reduced error dispersion, and improved predictive stability, highlighting the importance of advanced metaheuristic optimization in deep learning model development for financial forecasting tasks.

Overall, the findings confirm that a structured compositional RNN design combined with metaheuristic optimization provides a robust, scalable, and high-accuracy solution for stock price prediction in dynamic and nonlinear markets. The proposed framework offers a practical foundation for developing intelligent decision-support systems to assist investors and financial analysts.

For future research, the framework may be extended by incorporating additional technical indicators, attention mechanisms, hybrid deep learning architectures, or transformer-based models to further enhance forecasting adaptability and robustness across different financial markets.

## REFERENCES

- [1] Bodie, A. Kane, and A. J. Marcus, *Investments*, 11th ed., New York, NY, USA: McGraw-Hill, 2018.
- [2] F. K. Reilly, K. C. Brown, and S. J. Leeds, *Investment Analysis & Portfolio Management*, 11th ed., Singapore: Cengage Learning Asia PTE Limited, 2019.
- [3] P. Lin, S. Ma, and R. Fildes, "The extra value of online investor sentiment measures on forecasting stock return volatility: A large-scale longitudinal evaluation based on Chinese stock market," *Expert Systems with Applications*, vol. 238, 121927, 2024.
- [4] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: Machine learning techniques applied to financial market prediction," *Expert Syst. Appl.*, vol. 124, pp. 226–251, Jun. 2019.
- [5] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: A review and taxonomy of prediction techniques," *Int. J. Financial Stud.*, vol. 7, no. 2, p. 26, May 2019.
- [6] Y. Zhou, H. Ren, Z. Li, and W. Pedrycz, "An anomaly detection framework for time series data: An interval-based approach," *Knowledge-Based Systems*, vol. 228, 107153, 2021.
- [7] J. Peng, C. Deng, and Y.-C. Chen, "Application of ARIMA time series model in stock data prediction," in *Proc. Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, 2023, pp. 223–232.
- [8] K. Prakhar, S. Sountharajan, E. Suganya, M. Karthiga, and S. Kumar, "Effective stock price prediction using time series forecasting," in *Proc. 6th Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2022, pp. 1636–1640.
- [9] B. Xiao, "Prediction of U.S. stocks based on ARIMA model," in *Proc. 8th Int. Conf. Financial Innov. Econ. Develop. (ICFIED)*, 2023, pp. 312–322.
- [10] A. Ziková and J. Veselá, "Forecasting analysis of stock prices on European markets using the ARIMA-GARCH model," *Statistika: Statist. Economy J.*, vol. 103, no. 3, pp. 342–354, Sep. 2023.
- [11] S. V. Kedar, "Stock market increase and decrease using Twitter sentiment analysis and ARIMA model," *Turkish J. Comput. Math. Educ. (TURCOMAT)*, vol. 12, no. 1S, pp. 146–161, Apr. 2021.
- [12] S. Singh, K. S. Parmar, and J. Kumar, "Soft computing model coupled with statistical models to estimate future of stock market," *Neural Comput. Appl.*, vol. 33, no. 13, pp. 7629–7647, Jul. 2021.
- [13] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," *Expert Syst. Appl.*, vol. 197, Art. no. 116659, Jul. 2022.

- [14] Y. Liu, A. Ayitelieke, and J. Yu, "Short-term stock price prediction algorithm construction based on integrated learning of SVR and RF with bagging," *Highlights Sci., Eng. Technol.*, vol. 22, pp. 8–15, Dec. 2022.
- [15] M. Bansal, A. Goyal, and A. Choudhary, "Stock market prediction with high accuracy using machine learning techniques," *Proc. Comput. Sci.*, vol. 215, pp. 247–265, Jan. 2022.
- [16] A. L. Gavin, P. K. Prasanna, C. V. Vedha, and A. Sinduja, "Data analysis and price prediction of stock market using machine learning regression algorithms," *Adv. Sci. Technol.*, vol. 124, pp. 409–417, Mar. 2023.
- [17] B. T. Khoa and T. T. Huynh, "Forecasting stock price movement direction by machine learning algorithm," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 12, no. 6, pp. 6625–6634, Dec. 2022.
- [18] Y. Huang, L. F. Capretz, and D. Ho, "Machine learning for stock prediction based on fundamental analysis," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–10.
- [19] A. Quadir, S. Kapoor, A. V. C. Junni, A. K. Sivaraman, K. F. Tee, H. Sabireen, and N. Janakiraman, "Novel optimization approach for stock price forecasting using multi-layered sequential LSTM," *Appl. Soft Comput.*, vol. 134, Art. no. 109830, Feb. 2022.
- [20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.