Optimizing Convolutional Neural Networks Architectures with Genetic Algorithms for Enhanced Time Series Prediction

Chia-Yang Fang, Jui-Chung Hung
Department of Computer Science University of Taipei, Taipei, Taiwan
z034598656@gmail.com, juichung@gmail.com
Corresponding Author: Jui-Chung Hung Email: juichung@gmail.com

***Abstract*—This study explores the application of convolutional neural networks (CNNs) in financial time series forecasting, specifically within the stock market domain. Given the dynamic and irregular nature of stock market data, this research proposes CNN architectures tailored to effectively capture these patterns. The simultaneous optimization of CNN parameters and architectures presents a highly nonlinear and complex problem. To address this challenge, we introduce an iterative optimization framework leveraging genetic algorithms (GAs) for parameter estimation. The GA-based approach is designed to efficiently navigate the solution space, achieving a globally optimal configuration with accelerated convergence. Experimental results demonstrate that incorporating both architectural and parametric optimization significantly enhances the model's performance, yielding superior in-sample estimation accuracy and improved out-of-sample next-day stock price predictions.**

Keywords—Convolutional Neural Network; Genetic Algorithm; Time Series; Deep Learning.

I. INTRODUCTION

In recent years, the global economy has faced numerous challenges, with severe issues such as inflation and soaring housing prices significantly impacting economic stability. These factors have led to a continuous increase in the cost of living, making homeownership increasingly difficult for the general population. Against this backdrop, investing in the stock market has emerged as a crucial means of generating passive income, serving not only as a hedge against the decline in real purchasing power but also as a strategy to enhance overall financial well-being. However, stock prices are influenced by multiple factors, including corporate financial performance, global economic conditions, and market sentiment, exhibiting highly nonlinear and irregular fluctuations. These complexities make accurate stock price prediction an exceptionally challenging task.

II. RELATED WORKS

A. Characteristics of the Stock Market

Before the widespread adoption of deep learning, many researchers relied on traditional statistical methods and time series models for stock price prediction, such as the Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models. While these models handle linear data well, they struggle with nonlinear and irregular fluctuations [1], [2]. Reference [3] applied Rough Set Theory (RST) to extract temporal rules from stock market data. Their findings confirmed identifiable patterns between market indicators and stock price movements, showing that historical market conditions correlate with future trends.

To address the limitations of traditional time series models, researchers have shifted toward deep learning approaches. With powerful feature extraction and pattern recognition capabilities, deep learning techniques can uncover latent relationships in large historical datasets and effectively capture stock market dynamics. Deep learning has been widely applied to stock price prediction and investment analysis, showing superior potential over traditional models.

B. Applications of Deep Learning in the Stock Market

Convolutional Neural Networks (CNNs) are a powerful deep learning model capable of identifying feature patterns in data[4]. With their hierarchical architecture, CNNs learn local features and long-term dependencies in market data, uncovering latent patterns in price fluctuations to improve time series prediction.

However, model parameters directly affect performance. Reference [5] showed that deeper convolutional layers can improve model performance but may cause overfitting when training data is limited. Additionally, kernel size and the number of filters influence feature extraction [6], [7]. Selecting optimal parameters requires extensive experimentation, making the process time-consuming.

C. Optimization Methods for Parameters and Model Structure

To improve parameter tuning efficiency and reduce manual effort, researchers have developed optimization techniques to find optimal model architectures. In this study, Genetic Algorithm (GA) is employed due to its flexible chromosome encoding using bit arrays, enabling it to handle both discrete and continuous parameter spaces. Moreover, GA uses crossover and mutation to avoid local optima and improve global search efficiency [8], [9]. Thus, GA is well-suited for optimizing model structure and parameters in this study.

Several studies have applied GA to optimize time series analysis. Reference [10] used GA to optimize CNN kernel size and filter count. Experiments showed that GA-optimized CNNs outperformed non-optimized models in stock market prediction, confirming GA's effectiveness. Reference [11] proposed a GA-based approach to tune CNN training parameters. Their study aimed to adjust CNN hyperparameters (learning rate, batch size, dropout rate) to improve convergence speed and accuracy. However, no study has integrated GA with deep learning to optimize both structural and training parameters, highlighting its potential for further research.

Building upon previous studies, this paper proposes a GA-driven binary encoding approach aimed at simultaneously optimizing both the structural and training parameters of deep learning models through a global search mechanism. Given the uncertainty and volatility of the stock market, this method seeks to develop a highly adaptive prediction model, enhancing both forecasting accuracy and model stability.

III. Materials and Methods

Section A explains stock market data structure and preprocessing to prepare it for deep learning. Section B introduces CNN fundamentals, highlighting their applications and advantages in time series forecasting. Section C examines how Genetic Algorithms (GA) use binary encoding for model structures and parameters. It also analyzes how GA optimizes configurations via selection, crossover, and mutation to find the best predictive model.

A. Data Acquisition and Preprocessing

This study uses the yfinance API to retrieve key stock market data, including Date, Open, High, Low, Close, and Volume. These indicators reflect price fluctuations and trading dynamics, providing essential information for stock market prediction. To ensure data quality and model reliability, this study performs data cleaning to address missing values, outliers, and other factors affecting analysis, ensuring input data completeness and accuracy.

Given the varying numerical ranges of dataset features, standardization ensures the model processes feature relationships effectively while preventing data leakage. Thus, both training and test sets are normalized before model training. Table 1 shows the raw training data of Apple Inc. (AAPL) stock before normalization. The trading volume (Volume) range differs significantly from other features, potentially causing imbalanced weight distribution in training. This may affect model predictive performance.

TABLE I. APPL Stock Data Before Normalization

Date	Open	High	Low	Close	Volume
2014-01-02	19.85	19.89	19.72	19.75	234684800
2014-01-03	19.75	19.77	19.30	19.32	392467600
…	…	…	…	…	…
2023-12-27	192.49	193.50	191.09	193.15	48087700
2023-12-28	194.14	194.66	193.17	193.58	34049900

This study applies Min-Max Normalization from scikit-learn to scale data within [0,1], preserving distribution characteristics and relative feature relationships. The computation formula is given in (1). Table 2 shows the normalized training data, confirming that feature values are scaled appropriately for effective model learning.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

TABLE II. APPL Stock Data After Normalization

Date	Open	High	Low	Close	Volume
2014-01-02	0.0120	0.0108	0.0116	0.0105	0.2022
2014-01-03	0.0114	0.0102	0.0093	0.0081	0.3537
…	…	…	…	…	…
2023-12-27	0.9693	0.9663	0.9670	0.9724	0.0230
2023-12-28	0.9784	0.9727	0.9786	0.9748	0.0096

To help the model learn timestep dependencies, this study applies the Sliding Window technique. It segments time series data into fixed-length windows, preserving local timestep dependencies. This allows the model to better capture dynamic patterns, improving learning efficiency and prediction accuracy.

B. Design of Structured Deep Learning Models

1) Convolutional Neural Network

CNNs are widely used in image recognition and financial market prediction [4]. In stock market prediction, CNNs extract features from historical price movements and technical indicators, adapting to market fluctuations. The architecture in this study consists of convolutional, flatten, and fully connected layers. Convolutional layers capture local structures and trends, improving pattern recognition.

To optimize performance, this study examines key parameters, including convolutional layers, filters, and kernel size, which affect feature extraction and learning capacity. A GA optimizes parameters adaptively, dynamically adjusting CNN configurations for effective market trend learning (Fig. 1).

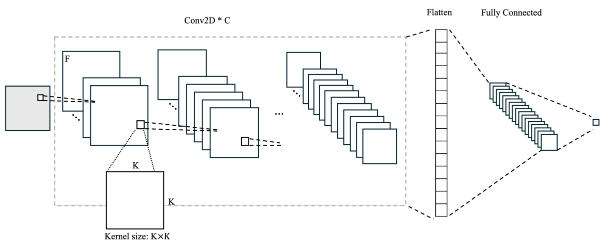

Fig. 1. Convolutional Neural Network architecture. C represents the number of convolutional layers, K denotes the kernel size, and F indicates the number of filters.

2) *Design of Modeling Parameters*

In addition to model parameters, hyperparameters significantly impact performance. For instance, a large learning rate may hinder convergence, while a small one slows training. A large batch size consumes excessive memory, whereas a small batch size causes unstable gradients. Similarly, too many training epochs may lead to overfitting, while too few result in underfitting. The dropout rate also plays a key role in preventing overfitting by randomly deactivating neurons during training.

Given the complex interdependencies among hyperparameters, this study employs GA for automated search and optimization. Using a fitness function to evaluate parameter combinations, GA applies selection, crossover, and mutation to find the optimal configuration, ultimately improving model predictive accuracy.

C. Design of Genetic Algorithm

GA is a heuristic search method that simulates biological evolution through selection, crossover, and mutation. Using a fitness function to assess candidate solutions, GA enables the population to evolve toward the optimal solution [8], [9].

Since both model architecture and training parameters are crucial to deep learning performance, this study employs GA to optimize them simultaneously. To enhance search efficiency, a binary encoding scheme maps each parameter into a sequence of 0s and 1s, forming a binary array that represents parameter combinations, simulating genetic evolution (Fig. 2). Through fitness evaluation and genetic evolution, GA adaptively searches for the optimal CNN architecture and training parameters, ultimately improving prediction accuracy and generalization.

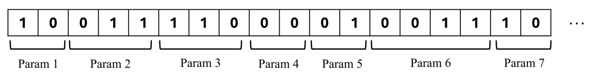

Fig. 2. Schematic representation of the binary array in the Genetic Algorithm

To ensure diversity in the initial population, this study uses a Bernoulli distribution to generate initial genes, ensuring a uniform search space. An elitism strategy preserves the fittest individuals to accelerate convergence. Single-point crossover enhances genetic diversity, while mutation introduces random perturbations to avoid local optima. Additionally, Mean Squared Error (MSE) is used as the fitness function to evaluate predictive performance, as formulated in (2).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2}$$

To maintain a fixed-length genetic sequence, this study enforces a uniform parameter rule, ensuring consistent kernel size and filter count across CNN layers. This prevents parameter mismatches, improving stability and computational efficiency in GA-based optimization. While simplifying the parameter space, this design ensures consistent genetic encoding, enhancing search reliability.

IV. Experiments and Results

This study selects two U.S. stock market assets, Apple Inc. (AAPL) and Meta Platforms Inc. (META), to evaluate GA's effectiveness in optimizing CNN structures. Historical data are retrieved via the yfinance API, with the training set spanning January 1, 2014, to December 31, 2023, and 20% allocated for validation. The test set, covering January 1, 2024, to December 31, 2024, assesses the model's generalization ability.

For model evaluation, this study uses Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) as performance metrics. RMSE measures the absolute deviation between predicted and actual values (3), while MAPE assesses percentage error relative to actual values, providing an intuitive measure of predictive accuracy (4). Using these metrics, this study evaluates the impact of GA-optimized CNNs on AAPL and META stock forecasting and explores their adaptability across different market conditions.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{3}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{4}$$

The GA parameter design in this study is based on the settings proposed by [12] as an initial reference. To improve adaptability and efficiency in optimizing CNN structures and hyperparameters, this study incorporates parameter settings from multiple related studies, expanding the search space for comprehensive optimization [12]–[15]. The final GA optimization range is determined based on these references, with detailed configurations in Table 3.

TABLE III. TUNABLE PARAMETERS IN THE GA-CNN MODEL

Parameters	**Value**
Convolution Layers	[1-3]
Filters	[32, 64, 128, 256]
Kernel size	[1-4]
Dropout	[0-0.3]
Timesteps	[1-16]
Batch size	[16, 32, 64, 128]
Learning rate	[0.001, 0.01]
Epochs	[20, 30, 40, 50]

Table 4 compares the baseline CNN and GA-optimized CNN models using RMSE and MAPE on training and testing datasets.

The results show that the GA-optimized CNN consistently outperforms the baseline CNN, achieving lower RMSE and MAPE across both datasets. For AAPL, the GA-CNN reduces training RMSE from 2.97 to 2.43 and testing RMSE from 4.08 to 3.05. Training MAPE decreases from 2.36% to 1.88%, while testing MAPE drops from 1.63% to 1.10%, demonstrating GA's effectiveness in improving in-sample and out-of-sample accuracy. For META, similar improvements are observed. The GA-CNN lowers training RMSE from 7.67 to 6.43 and testing RMSE from 15.02 to 11.86. Training MAPE decreases from 2.78% to 2.30%, while testing MAPE drops from 2.19% to 1.59%. These results confirm that the GA-optimized CNN enhances predictive performance across stocks with different volatility characteristics.

Fig. 3 compares actual stock prices of AAPL and META with predictions from the baseline CNN and GA-optimized CNN. The GA-optimized model (red X) aligns more closely with actual prices (black line) than the baseline CNN (blue square), demonstrating superior trend-capturing ability. These findings confirm the effectiveness of GA-driven optimization in improving predictive accuracy and model adaptability.

The results indicate that GA identifies different optimal CNN structures for AAPL and META, reflecting variations in price patterns and volatility. This underscores the importance of adaptive model design, where CNN architecture and hyperparameters are tailored to each stock's characteristics. Such flexibility enables GA-CNN to maintain high predictive accuracy across diverse market conditions.

TABLE IV. PERFORMANCE COMPARISON BETWEEN CNN AND GA-CNN MODELS FOR APPLE INC. (AAPL) AND META PLATFORMS INC. (META)

Model	**Training RMSE**	**Testing RMSE**	**Training MAPE**	**Testing MAPE**
Apple Inc. (AAPL)				
CNN	2.97	4.08	2.36%	1.63%
GA-CNN	2.43	3.05	1.88%	1.10%
Meta Platforms Inc. (META)				
CNN	7.67	15.02	2.78%	2.19%
GA-CNN	6.43	11.86	2.30%	1.59%

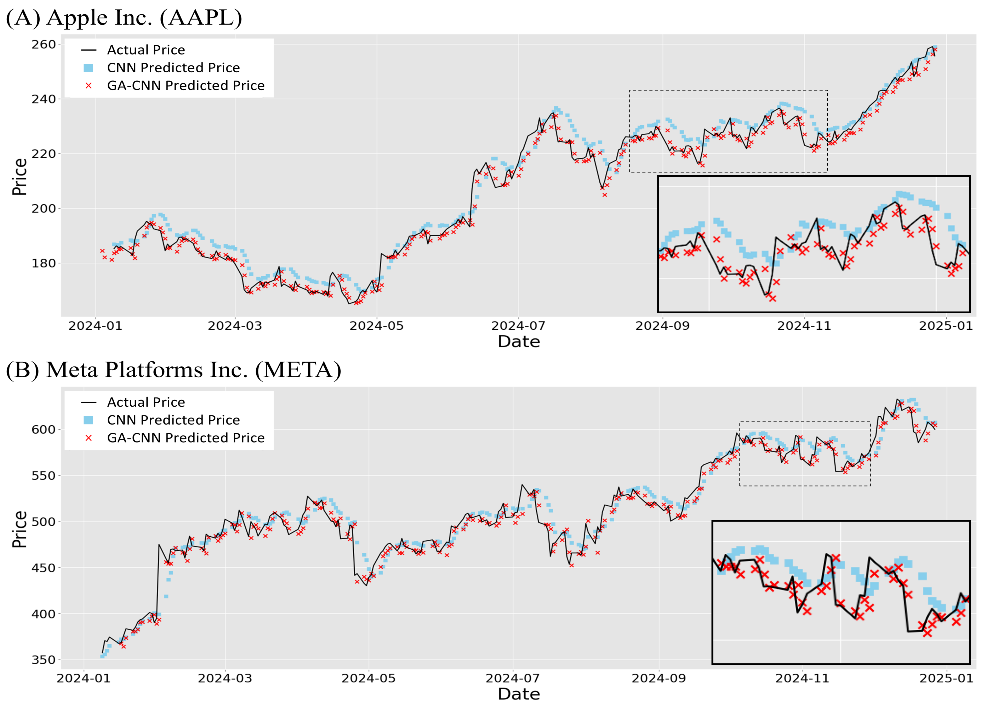

Fig. 3. Comparison of predicted and actual stock prices using CNN and GA-CNN models.

V. Conclusions

This study investigates the application of Genetic Algorithm (GA) in optimizing the architecture of Convolutional Neural Networks (CNNs) and evaluates its effectiveness in predicting stock prices of Apple Inc. (AAPL) and Meta Platforms Inc. (META). By adaptively adjusting CNN hyperparameters, including the number of layers, kernel size, and the number of filters, GA successfully constructs a model capable of capturing historical market patterns and improving forecasting accuracy.

Experimental results indicate that GA-optimized CNNs demonstrate significant improvements in both RMSE and MAPE metrics compared to fixed-architecture CNNs, suggesting that GA can effectively search for the optimal CNN structure and enhance the model's adaptability to market trends. Moreover, in varying market conditions—such as the high volatility of META and the relative stability of AAPL—the CNN structures optimized by GA differ, indicating the adaptability and flexibility of this approach.

However, this study has certain limitations. For instance, GA is constrained by the search space and computational cost. Future research could explore Bayesian Optimization or Reinforcement Learning to further enhance optimization efficiency. Additionally, integrating LSTM or Transformer-based time-series models may improve the model's ability to capture long-term dependencies in stock market predictions.

Acknowledgment

The author would like to express sincere gratitude to Professor Jui-Chung Hung for his invaluable guidance and encouragement throughout this research. His insightful feedback and support have been instrumental in refining this work. Additionally, the author appreciates the assistance of PhD student Yi Chuang from National Taiwan University for helping with formatting, manuscript drafting, and revision. Yi Chuang's contributions have greatly improved the clarity and presentation of this paper.

References

[1] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," *J Appl Math*, vol. 2014, pp. 1–7, 2014, doi: 10.1155/2014/614342.

[2] R. Gençay, "The predictability of security returns with simple technical trading rules," *J Empir Finance*, vol. 5, pp. 347–359, 1998.

[3] R. Golan and D. Edwards, "Temporal rules discovery using datalogic/R+ with stock market data," 1994, pp. 74–81. doi: 10.1007/978-1-4471-3238-7_9.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[5] M. Kabkab, E. Hand, and R. Chellappa, "On the size of convolutional neural networks and generalization performance," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2016, pp. 3572–3577. doi: 10.1109/ICPR.2016.7900188.

[6] G. Jin and O. Kwon, "Impact of chart image characteristics on stock price prediction with a convolutional neural network," *PLoS One*, vol. 16, no. 6, p. e0253121, 2021, doi: 10.1371/journal.pone.0253121.

[7] S. Khurana, "Deep learning hybrid CNN-LSTM framework for multi-stock price prediction: An advanced comparative analysis of Tesla, S&P 500, Amazon, and Apple," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 22, no. 6, 2024, [Online]. Available: https://google.academia.edu/JournalofComputerScience

[8] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[9] J. H. Holland, *Adaptation in natural and artificial systems*. The University of Michigan Press, 1975.

[10] H. Chung and K. Shin, "Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction," *Neural Comput Appl*, vol. 32, no. 12, pp. 7897–7914, 2020, doi: 10.1007/s00521-019-04236-3.

[11] A. Zafar *et al.*, "An optimization approach for convolutional neural network using non-dominated sorted genetic algorithm-II," *Computers, Materials & Continua*, vol. 74, no. 3, pp. 5641–5661, 2023, doi: 10.32604/cmc.2023.033733.

[12] P. Singh, M. Jha, M. Sharaf, M. A. El-Meligy, and T. R. Gadekallu, "Harnessing a hybrid CNN-LSTM model for portfolio performance: A case study on stock selection and optimization," *IEEE Access*, vol. 11, pp. 104000–104015, 2023, doi: 10.1109/ACCESS.2023.3317953.

[13] A. S. Girsang and D. Tanjung, "Fast genetic algorithm for long short-term memory optimization," *Engineering Letters*, vol. 30, no. 2, 2022.

[14] E. Urbinate, F. Itano, and E. Del-Moral-Hernandez, "CNN-LSTM optimized by genetic algorithm in time series forecasting: An automatic method to use deep learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14125 LNAI, Springer Science and Business Media Deutschland GmbH, 2023, pp. 286–295. doi: 10.1007/978-3-031-42505-9_25.

[15] X. Zeng, J. Cai, C. Liang, and C. Yuan, "A hybrid model integrating long short-term memory with adaptive genetic algorithm based on individual ranking for stock index prediction," *PLoS One*, vol. 17, no. 8, p. e0272637, 2022, doi: 10.1371/journal.pone.0272637.