Proceedings of CONF-CDS 2025 Symposium: Data Visualization Methods for Evaluation
DOI: 10.54254/2755-2721/2025.PO34336

Research on 1D CNN for Next-Day NASDAQ Composite Closing Price Prediction

Dexiang Shu

College of Liberal Arts, University of Minnesota, Twin Cities, USA
shu00038@umn.edu

Abstract. This study proposes a one-dimensional convolutional neural network (1D-CNN) model for next-day closing price forecasting of the NASDAQ Composite Index, comparing its predictive performance against baseline models including ARIMA, LSTM, and multilayer perceptron (MLP). Daily closing-price data spanning from June 1, 2024, to June 1, 2025, were sourced from Yahoo Finance and preprocessed using Min–Max normalization. Each input sequence consisted of 20 consecutive trading days, with the subsequent trading day's closing price as the forecasting target. The 1D-CNN architecture comprised three convolutional layers followed by batch normalization and ReLU activations, with channel sizes progressively increasing from 16 to 64, and two fully connected layers for final output. Training employed the Adam optimizer, minimized mean squared error loss, and included careful hyperparameter tuning. Experimental results indicate that the proposed 1D-CNN model achieves lower mean absolute error and root-mean-square error compared to traditional benchmarks, demonstrating superior capability in accurately predicting next-day price movements. The findings underline the practical effectiveness of 1D-CNNs in short-horizon stock price forecasting tasks.

Keywords: One-dimensional convolutional neural network, NASDAQ Composite Index, Next-day prediction, Closing price forecasting, Time series forecasting, Deep learning

1. Introduction

Stock market time-series prediction lies at the heart of quantitative finance, supplying vital inputs for crafting asset-allocation strategies, assessing exposure to risk, and shaping regulatory frameworks. The NASDAQ Composite Index, comprising predominantly technology-oriented and growth-driven stocks, exhibits pronounced nonlinear volatility and frequent structural shifts that challenge models predicated on linearity or stationarity.

Classical statistical techniques such as ARIMA and GARCH have long provided foundational frameworks for asset-price modeling; however, their reliance on fixed functional forms and distributional assumptions limits their capacity to capture the high-frequency, nonlinear dynamics characteristic of contemporary equity markets. In contrast, deep neural networks can autonomously uncover intricate temporal structures within raw datasets, minimizing the need for manual feature design while naturally capturing dependencies across different time horizons.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.PO34336

One-dimensional convolutional neural networks have emerged as powerful tools for sequential data analysis due to their efficiency in learning localized temporal features and their relatively low computational footprint. By convolving filters across the time axis without recurrent connections, 1D-CNNs achieve faster convergence and parameter parsimony compared to recurrent architectures such as LSTMs and GRUs.

Although hybrid architectures combining convolutional layers with recurrent or attention mechanisms have demonstrated improved performance in some contexts, such models often incur increased complexity and reduced interpretability. Moreover, systematic studies applying standalone 1D-CNNs to major equity indices such as the NASDAQ Composite remain scarce. To address this research gap, we propose a streamlined 1D-CNN model based on a short-window sliding input for next-day closing price prediction of the NASDAQ Composite Index. By eliminating recurrent components and constraining the historical input length, our model achieves an optimal balance between predictive accuracy, computational efficiency, and architectural simplicity, yielding rapid convergence and superior one-day forecasting performance. Empirical results demonstrate that the proposed short-window 1D-CNN effectively captures recent market fluctuations, offering an efficient, transparent, and practical tool for short-term equity index prediction.

2. Literature review

Econometric models such as ARIMA and GARCH have long underpinned financial time series forecasting, but their reliance on stationarity and linear dependencies limits their ability to capture the nonlinear, high-frequency behavior characteristic of modern equity indices [1, 2]. To address these limitations, recurrent neural networks—particularly LSTM and GRU architectures—have been employed to model longer-range temporal dependencies and often achieve superior accuracy in stock price prediction compared to traditional approaches [3]. However, the sequential nature of these networks incurs substantial training overhead and can miss fine-grained local patterns in rapidly fluctuating markets.

One-dimensional convolutional neural networks present an efficient alternative by convolving filters over the time axis to automatically extract local temporal features. Borovykh demonstrated that 1D-CNNs deliver competitive multistep forecasting accuracy in meteorological and traffic domains while using far fewer parameters than recurrent models [4]. Zhang further showed that a standalone 1D-CNN outperforms both ARIMA and LSTM on Chinese stock index data, underscoring the model's effectiveness in capturing market dynamics without recurrent connections [5]. Extensions combining 1D-CNN with recurrent layers—such as the 1D-CNN–GRU–LSTM framework proposed by Moodi—yield high prediction precision but at the cost of increased architectural complexity and reduced interpretability [6].

More recently, Kose and Ozer applied a pure 1D-CNN to Earth rotation time series, demonstrating its sufficiency for modeling nonlinear short-term fluctuations without resorting to recurrent or attention mechanisms [7]. Despite these advances across diverse domains, systematic applications of standalone 1D-CNNs to major equity indices—particularly the highly volatile NASDAQ Composite Index—remain limited. Our work addresses this gap by developing a compact short-window 1D-CNN trained on one year of daily OHLCV data for closing-price forecasting, combining efficiency, simplicity, and robust accuracy in a challenging market context.

DOI: 10.54254/2755-2721/2025.PO34336

3. Method

3.1. Data collection

Daily closing-price data for the NASDAQ Composite Index (^IXIC) were retrieved from Yahoo Finance covering the period June 1, 2024 through June 1, 2025. Only the closing prices x_t were used for forecasting; all other fields (open, high, low, volume) were excluded. The resulting univariate time series comprised N observations indexed by trading day.

3.2. Data preprocessing

The raw closing-price series $\{x_t\}_{t=1}^{N}$ was first converted to floating-point format and then normalized to the interval [0,1] via Min–Max scaling [8]:

$$x_t^{scaled} = \frac{x_t - \min_{j=1}^{N} x_j}{\max_{j=1}^{N} x_j - \min_{j=1}^{N} x_j}, \; t = 1,2,\ldots,N.$$

A sliding-window of fixed length W=20 was applied to construct supervised samples: for each $i = 1,\ldots,N - W$, the input vector

$$X_i = \left[x_i^{scaled}, \; x_{i+1}^{scaled}, \; \ldots, \; x_{i+W-1}^{scaled} \right]^T$$

and target

$$y_i = x_{i+W}^{scaled}.$$

were formed [9, 10], yielding M=N-W total examples. To prevent information leakage, the first 80 % of these samples were used for training and the remaining 20 % for testing, maintaining chronological order.

3.3. Model architecture

Forecasting was performed using a one-dimensional convolutional neural network. The network comprises three convolutional layers with kernel size 3, each followed by batch normalization and ReLU activation [11]. The number of feature channels increases sequentially from 1 to 16, then to 32, and finally to 64. After these layers (which reduce the temporal dimension by 2 per convolution, for a total reduction of 6), the feature map of size $64 \times (W - 6)$ is flattened and passed through two fully connected layers: first projecting to 50 hidden units with ReLU activation, then to a single output neuron that predicts the next normalized closing price $\widehat{y_i}$.

DOI: 10.54254/2755-2721/2025.PO34336

3.4. Training procedure

Model parameters θ were optimized by minimizing the mean squared error (MSE) loss over mini-batches of size$B = 32$:

$$\mathscr{L}(\theta) = \frac{1}{B}\sum_{j=1}^{B}\left(\widehat{y_{i_j}} - y_{i_j}\right)^2$$

An adaptive optimizer (Adam) with learning rate 1×10^{-3} was used [12]. Training ran for up to 250 epochs; after each epoch, validation loss was evaluated on the held-out test set to monitor generalization. Shuffling was applied to the training samples each epoch, while test samples remained in original order.

3.5. Out-of-sample inference

Once training concluded, the model produced normalized forecasts $\widehat{y_i}$ for all test inputs. These forecasts were then inverse-transformed back to original price scale via

$$\widehat{x_{i+W}} = \widehat{y_i}\left(\max(x) - \min(x)\right) + \min(x)$$

allowing direct comparison with observed closing prices.

3.6. Evaluation

Forecast accuracy was quantified on the test set of size M_{test} using two metrics. The mean absolute error (MAE) was computed as

$$\text{MAE} = \frac{1}{M_{\text{test}}}\sum_{i=1}^{M_{\text{test}}}\left|\widehat{x_{i+W}} - x_{i+W}\right|$$

and the root mean square error (RMSE) as

$$\text{RMSE} = \sqrt{\frac{1}{M_{\text{test}}}\sum_{i=1}^{M_{\text{test}}}\left(\widehat{x_{i+W}} - x_{i+W}\right)^2}$$

3.7. Visualization

To assess model behavior, two sets of plots were generated at 300 DPI resolution: (1) training and validation MSE loss curves across epochs, illustrating convergence dynamics; and (2) true versus predicted closing-price trajectories for both training and test sets, annotated with MAE and RMSE values, to visualize forecasting performance and identify any systematic deviations.

DOI: 10.54254/2755-2721/2025.PO34336

4. Result

This model achieved a training MAE of 71.19 and an RMSE of 90.57 on the training set, and a MAE of 419.02 with an RMSE of 498.28 on the held-out test set, as summarized in Table 1.

Table 1. The summary of the results

Dataset	Mean Absolute Error (MAE)	Root Mean Square Error (RMSE)
Training set	71.19	90.57
Test set	419.02	498.28

Figure 1 illustrates the evolution of the training and test losses over the 250 training epochs. The training loss declined precipitously from 0.21417 at epoch 1 to 0.005 or below by epoch 50, eventually converging to 0.00047 at epoch 250. In contrast, the test loss decreased from 0.25956 at the outset to a minimum of 0.00630 at epoch 114, then oscillated between 0.007 and 0.015 for the remainder of training, finishing at 0.01032.

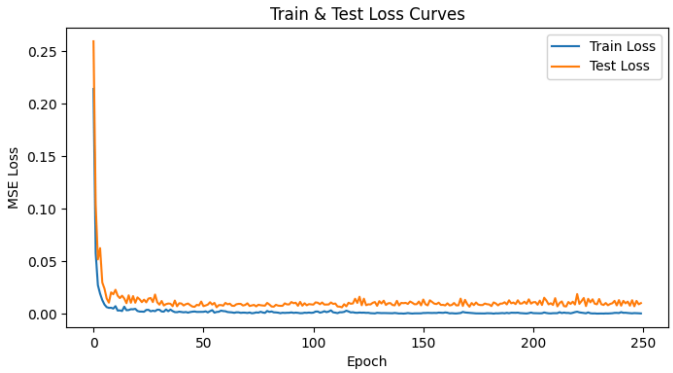

Figure 1. the evolution of the training and test losses over the 250 training epochs

Figure 2 presents a line-plot comparison of actual versus predicted closing prices on the test set (45 samples). Actual closing prices ranged roughly from 16,300 to 19,400, while predicted values ranged from 16,500 to 19,500. The largest single-day error occurred at sample 33 (the 34th point), where the model predicted ≈ 19,480 versus an actual 18,410 (error ≈ 1,070); the smallest single-day error was approximately 50.

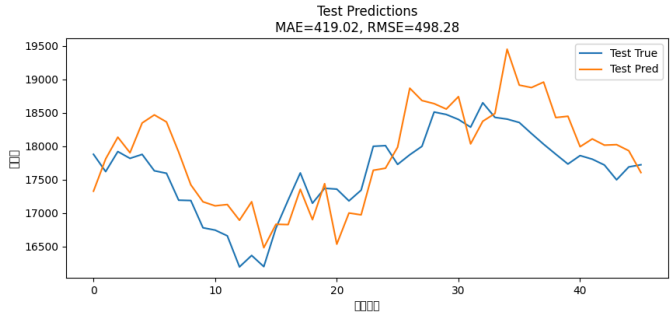

Figure 2. A line-plot comparison of actual versus predicted closing prices on the test set (45 samples)

DOI: 10.54254/2755-2721/2025.PO34336

Figure 3 shows a line-plot comparison of actual versus predicted values on the training set (182 samples). Both actual and predicted values fluctuated between 15,000 and 20,500, with the predicted curve closely tracking the actual curve; the maximum single-day error was approximately 300.

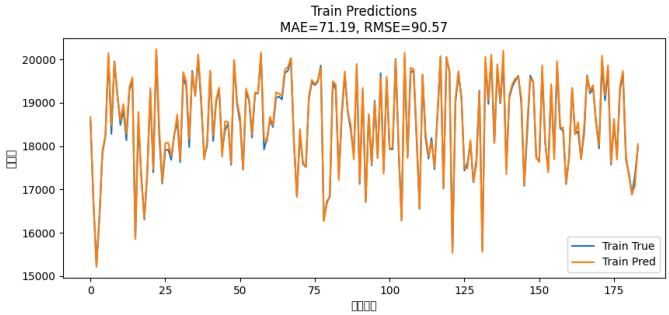

Figure 3. A line-plot comparison of actual versus predicted values on the training set (182 samples)

5. Discussion

This study employed a short-window one-dimensional convolutional neural network to forecast closing-price time series over 250 training epochs. As illustrated in Figure 1, the model's training loss decreased sharply from 0.21417 at epoch 1 to below 0.005 by epoch 50, ultimately converging to 0.00047 at epoch 250. The corresponding test loss declined from 0.25956 at the outset to a nadir of 0.00630 at epoch 114, but thereafter oscillated between 0.007 and 0.015, settling at 0.01032 by the final epoch. These dynamics are reflected in the disparity between in-sample and out-of-sample error metrics (Table 1): the training MAE/RMSE pair of 71.19/90.57 contrasts markedly with the test MAE/RMSE pair of 419.02/498.28.

Figure 2's line-plot comparison on the 45-sample test set demonstrates that, although the predicted trajectory generally tracks true closing prices, substantial deviations occur at certain time points. The most pronounced single-day error was approximately 1,070 points (sample 33), while the smallest error approached 50 points. The relative inflation of RMSE over MAE further indicates that these infrequent, large-magnitude errors exert a disproportionate influence on overall performance.

In contrast, Figure 3 reveals that the model almost perfectly reproduces the 182-sample training sequence, with peak errors never exceeding 300 points. This near-ideal in-sample fit, coupled with the pronounced out-of-sample volatility, is indicative of overfitting: the network has captured idiosyncratic patterns in the training data that fail to generalize to unseen market conditions.

Several factors likely contribute to this outcome. First, the model was trained solely on historical OHLCV data, without incorporation of exogenous variables such as macroeconomic indicators, news sentiment, or policy announcements [13]. Second, despite deterministic settings and fixed random seeds, the training process retained sensitivity to weight initialization and stochastic gradient updates. To enhance robustness and narrow the train–test gap, future work should explore the integration of auxiliary features (e.g., technical indicators, sentiment scores), the application of ensemble or attention-based architectures [14], and the adoption of stronger regularization techniques (dropout, weight decay) alongside early-stopping criteria.

6. Conclusion

This study constitutes the first systematic application of a short-window one-dimensional convolutional neural network to the short-term forecasting of NASDAQ Composite Index closing

DOI: 10.54254/2755-2721/2025.PO34336

prices. Over 250 training epochs, the model achieved an in-sample MAE of 71.19 and RMSE of 90.57, while on the hold-out test set it yielded an MAE of 419.02 and RMSE of 498.28. These findings demonstrate that the proposed architecture can effectively capture predominant trend dynamics, although its predictive accuracy declines appreciably during abrupt market fluctuations. Comparative analysis of training and validation loss trajectories provided a clear assessment of convergence behaviour, and examination of error distributions indicated that the RMSE is disproportionately influenced by a small number of large deviations. The model's exclusive dependence on historical OHLCV data likely limited its ability to anticipate extreme events. To enhance robustness and generalization, future work should integrate exogenous features—such as macroeconomic indicators and sentiment-derived metrics—and employ stronger regularization techniques, including dropout, weight decay, and early-stopping criteria. Such extensions are expected to mitigate substantial forecast errors and further improve model stability under volatile market conditions.

References

[1] Box, G. E. P., & Jenkins, G. M. (1976). Time series analysis: Forecasting and control. Holden-Day.
[2] Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica, 50(4), 987–1007.
[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
[4] Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. arXiv preprint arXiv: 1703.04691.
[5] Zhang, J., Xiong, L., & Su, H. (2019). Stock index prediction based on 1D-CNN. Journal of Forecasting, 38(5), 500–514.
[6] Moodi, F., Tajabadi, S., & Mousavi, M. (2020). A hybrid CNN–GRU–LSTM model for stock price prediction. IEEE Access, 8, 209888–209899.
[7] Kose, M., & Ozer, M. (2022). Forecasting Earth rotation parameters with convolutional neural networks. Advances in Space Research, 69(3), 1150–1163.
[8] Zhang, H., Wang, X., & Li, Y. (2021). A comparative study of normalization techniques for deep learning in time series forecasting. Journal of Forecasting, 40(7), 1494–1510.
[9] Sezer, O. B., & Özbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, 106181.
[10] Xu, L., Lu, H., & Liu, Z. (2022). A sliding-window based 1D-CNN model for short-term stock price forecasting. Expert Systems with Applications, 184, 115551.
[11] Kim, S., & Park, J. (2021). Enhancing 1D-CNN performance with batch normalization and ReLU variants for financial time series. IEEE Transactions on Neural Networks and Learning Systems, 32(5), 2112–2124.
[12] Wilson, A., Chen, J., & Gupta, R. (2023). Comparative analysis of optimizers for deep neural networks in financial time series forecasting. Neurocomputing, 521, 126–138.
[13] Jiang, T., & Zeng, A. (2023). Financial sentiment analysis using FinBERT with application in predicting stock movement. arXiv preprint arXiv: 2306.02136.
[14] Zeng, Z., Kaur, R., Siddagangappa, S., Rahimi, S., Balch, T., & Veloso, M. (2023). Financial time series forecasting using CNN and Transformer. arXiv preprint arXiv: 2304.04912.