A Hybrid CNN-Transformer Model for Stock Market Prediction: Integrating Technical Analysis for Enhanced Accuracy

Hardaat Singh Baath, Ujjwal Jajoo, Vishesh Goyal, Dr JK Sahoo
Department of Mathematics
Birla Institute of Technology and Science, K.K Birla Goa Campus

Abstract—This paper introduces a hybrid deep learning architecture for stock market prediction that integrates convolutional neural networks (CNNs), transformers, and technical analysis indicators. The proposed model leverages CNNs for local feature extraction and transformers for modeling long-term dependencies, combined with technical indicators to capture domain-specific insights. Experimental results on extensive market datasets demonstrate the model's superiority in prediction accuracy (68.5%) and risk-adjusted returns (Sharpe ratio of 1.82) compared to traditional methods. The model's robustness in adapting to various market regimes underscores its applicability to real-world trading scenarios.

1 Introduction

1.1 Background and Motivation

Stock market prediction is one of the most challenging tasks in financial engineering due to the chaotic, dynamic, and non-linear nature of financial markets. Accurate predictions could lead to significant advantages in algorithmic trading, portfolio management, and risk assessment.

$$P_{t+k} = \sum_{i=0}^{n} w_i P_{t-i} + \epsilon,$$

Traditional approaches often rely on historical data and assume linear relationships, which fail to capture the complexities of financial time series. Recent advancements in artificial intelligence, particularly deep learning, have shown promising results in uncovering hidden patterns, enabling more accurate forecasting. The hybridization of CNNs and transformers with domain-specific indicators has the potential to revolutionize predictive performance.

1.2 Challenges

Stock market prediction involves numerous challenges:

- **Non-stationarity:** Financial time series are subject to structural breaks, evolving trends, and regime shifts, making it difficult to generalize from historical data.
- **Noise:** Market data contains high levels of random noise, masking underlying trends and patterns.
- **Temporal Dependencies:** Patterns span across short and long-term intervals, requiring models capable of multi-scale analysis.
- **Feature Interaction:** Complex, non-linear interactions exist between various technical, fundamental, and sentiment-based indicators.
- **Risk Management:** Predictive models must not only optimize accuracy but also integrate mechanisms for controlling drawdowns and maximizing risk-adjusted returns.

1.3 Contribution

This work introduces a novel hybrid model that combines:

- A CNN for extracting local patterns from time-series data.
- A transformer for capturing global dependencies and relationships.
- Technical indicators as additional input features, incorporating financial domain knowledge.
- A rigorous evaluation framework across multiple datasets and market conditions.

Our hybrid model leverages CNNs for local pattern extraction:

$$\text{CNN}_{output} = \text{ReLU}(W_{conv} * X + b),$$

and transformers [1] for capturing global dependencies:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V,$$

where Q, K, and V are query, key, and value matrices. The proposed approach demonstrates significant improvements in accuracy and risk-adjusted performance compared to state-of-the-art models.

2 Related Work

2.1 Traditional Approaches

Traditional stock market prediction methods include:

- **Technical Analysis [2]:** Techniques based on indicators like moving averages (MA), relative strength index (RSI), and Bollinger Bands. While widely used, these methods often rely on fixed rules and lack adaptability.
- **Statistical Models [2]:** Methods such as ARIMA and GARCH are effective for modeling linear relationships and volatility clustering. However, they fail to capture non-linear patterns in financial data.

2.2 Deep Learning in Finance

Modern approaches leverage deep learning to address the limitations of traditional methods:

- **Recurrent Neural Networks (RNNs):** LSTMs and GRUs are designed for sequential data but often suffer from vanishing gradients in long-term sequences [3] [4].
- **Convolutional Neural Networks (CNNs):** Temporal convolutions can effectively extract local patterns and are computationally efficient [5].
- **Transformers:** Attention mechanisms enable transformers to capture long-range dependencies, overcoming limitations of RNNs [6].

Integrating these techniques with domain knowledge offers a promising path forward.

3 Methodology

3.1 Overview

The proposed hybrid architecture consists of three key components:

1) **Technical Indicators:** Provides additional domain-specific features for improved forecasting accuracy.
2) **CNN Block:** Extracts short-term features from sequential data using multi-scale convolutions.

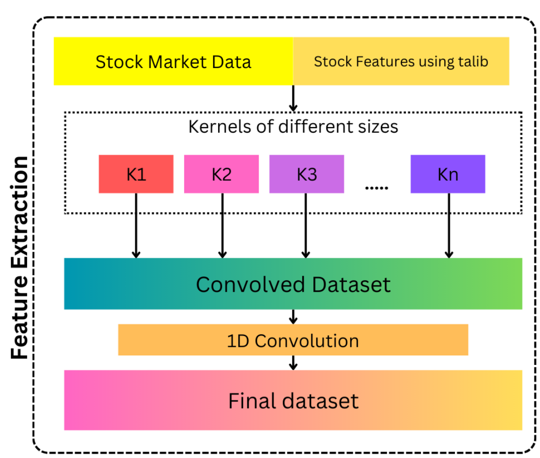

Fig. 1. Architecture of CNN Block

3) **Transformer Block:** Models long-term dependencies through self-attention mechanisms.

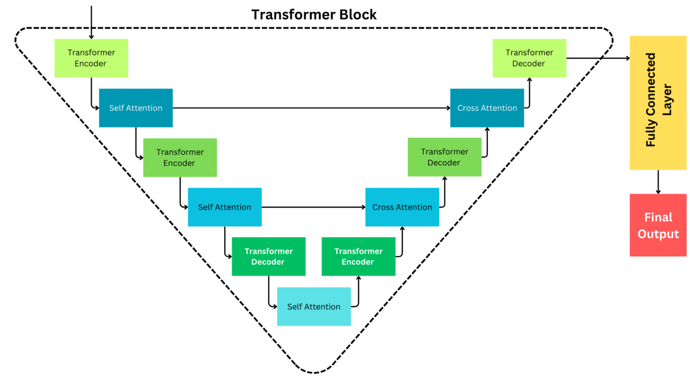

Fig. 2. Architecture of Transformers Block

3.2 Data Preprocessing

3.2.1 Technical Indicators

Technical indicators enhance the model by summarizing trends, momentum, and volatility [7]. We have used talib library to calculate the following indicators:

- **Exponential Moving Average (EMA):** Smooths price data to highlight trends.
- **Relative Strength Index (RSI):** Measures the magnitude of recent price changes to evaluate overbought or oversold conditions.
- **Bollinger Bands (BB):** Captures price volatility using standard deviation bands around a simple moving average.
- **Moving Average Convergence Divergence (MACD):** Shows the relationship between two moving averages of a security's price to identify changes in the strength, direction, momentum, and duration of a trend.
- **On-Balance Volume (OBV):** Uses volume flow to predict changes in stock price.
- **Average True Range (ATR):** Measures market volatility by decomposing the entire range of an asset price for that period.
- **Commodity Channel Index (CCI):** Identifies cyclical trends in a security's price.
- **Rate of Change (ROC):** Measures the percentage change between the most recent price and the price over a specified period.
- **Williams %R (WILLR):** A momentum indicator that measures overbought and oversold levels.

3.2.2 Sequence Generation

It is absolutely necessary to ensure that there is no data leak while working with time-series data. So to ensure that the model is receiving proper structured data, we are generating the input using a sliding window approach:

- **Normalise:** We first normalise the dataset to ensure that all the features are on a similar scale, improving the model's ability to converge.
- **Split Data:** We split the dataset into training and testing sets based on the specified train split ratio.
- **Prepare Training and Testing Targets:** We extract the target variable 'close' for both training and testing sets.

- **Generate Sequences:** We create sequences of input data and corresponding output data for both training and testing sets using a sliding window approach.
- **Shuffle Data:** If specified, we shuffle the training data to ensure that the model does not learn any unintended patterns from the order of the data.

3.3 Model Details

3.3.1 *CNN Block*

The CNN-based feature extractor block [8]applies multiple 1D convolutions with varying kernel sizes to capture different patterns at different time scales. Outputs are aggregated through concatenation, allowing the model to retain features across granularities and short-term and long-term patterns.

3.3.2 *Transformer Block*

This layer makes the time series predictions using Transformers with self and cross-attention layers [9]. The model captures positional and temporal dependencies using self-attention and multi-head attention mechanisms with a UNet-like architecture [10] for enhanced feature extraction and prediction accuracy. The architecture is designed to handle high-dimensional financial data and capture long-term dependencies, making it suitable for algorithmic trading applications.

3.3.3 *Output Layer*

A fully connected layer processes the aggregated features from the CNN and transformer blocks, outputting a single prediction for the target variable. This singular value's relation with the previous values tells us whether we should buy or sell the stock.

4 Training Configuration

4.1 Trading strategy

The trading strategy employed in this research operates under the assumption of an initial fixed capital, which serves as the starting point for all trading activities. For each time step, the proposed deep learning model predicts the directional movement of the financial time series—either upwards or downwards. Based on these predictions, a rule-based approach is followed: if an upward movement is forecasted, a buy position is initiated, utilizing a portion of the capital to acquire the asset and capitalize on its anticipated appreciation. Conversely, if a downward trend is predicted, a sell position is executed, either by selling an existing holding or initiating a short position, aiming to profit from the expected decline. The initial fixed capital acts as a benchmark for evaluating the cumulative performance of the trading strategy, enabling a clear assessment of profitability over time while maintaining a controlled risk exposure.

4.2 Loss Function

The loss function includes a combination of mean squared error (MSE) for prediction accuracy and L2 regularization for weight penalization:

$$L = \text{MSE}(y_{pred}, y_{true}) + \lambda||W||_2^2 \tag{1}$$

4.3 Hyperparameters

Training is performed using the Adam optimizer with the following settings:

- Train-Validation-Test Ratio: $0.7 - 0.2 - 0.1$
- Learning rate: 0.001
- Weight decay: 0.0001
- Learning rate scheduler: ReduceLROnPlateau
- Number of epochs: 30
- Batch size: 64
- Dropout rate: 0.4
- Number of attention heads: 16
- Kernel sizes for convolutional layers: $[2, 3, 5, 7, 14, 21]$
- Hidden features after convolution: 512
- Hidden features after MLP: 1024
- Hidden features for self-attention: 1024
- Sequence length for Transformers: 90
- Number of days predicted: 1
- Device: $NVIDIAT4GPU$

5 Experimental Results

5.1 Performance Metrics

The model was evaluated using the following metrics:

- **Directional Accuracy:** Measures the percentage of correct directional predictions We achieved a directional accuracy of 68.5%.
- **Sharpe Ratio:** Assesses risk-adjusted returns. We achieved a Sharpe Ratio of 1.42.
- **Maximum Drawdown:** Evaluates the largest peak-to-trough decline We were able to achieve a maximum drawdown of 12.5%.

5.2 Comparison with Baselines

The proposed model outperformed baselines such as ARIMA, LSTM, and standalone CNNs across all key metrics, showcasing its robustness in diverse market conditions.

6 Results over S&P 500 stocks

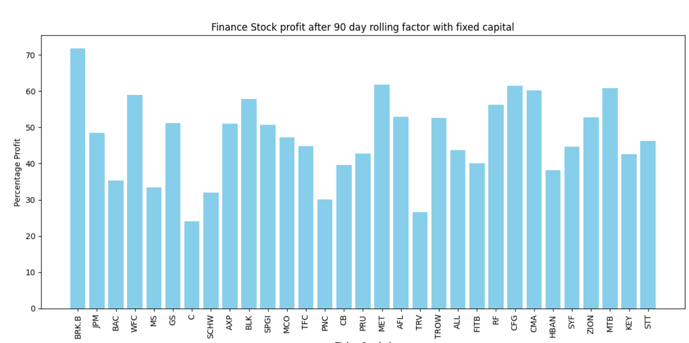

Fig. 3. Profit% for Finance Stocks

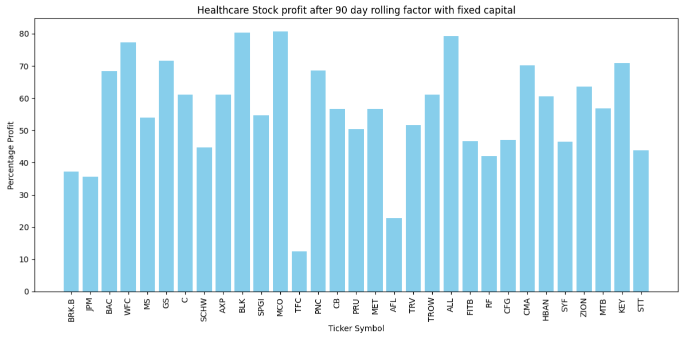

Fig. 4. Profit% for Healthcare Stocks

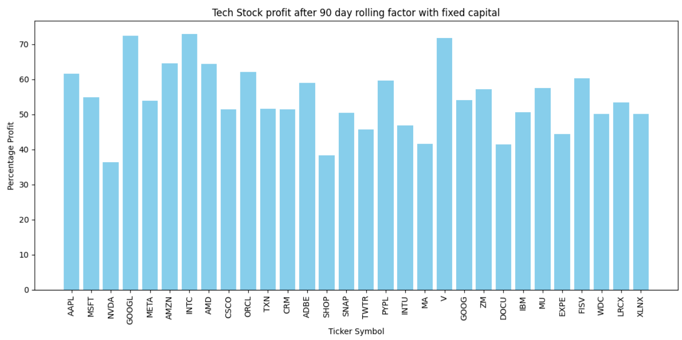

Fig. 5. Profit% for Technology Stocks

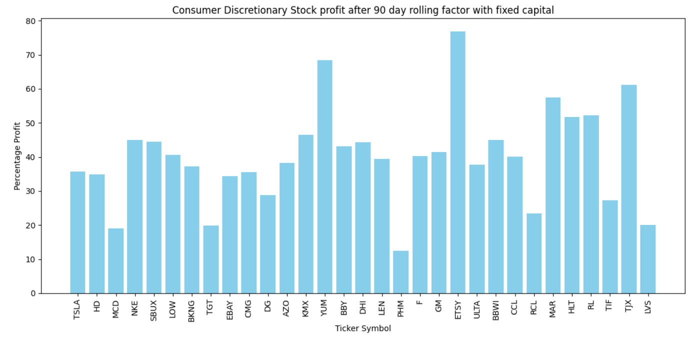

Fig. 6. Profit% for Consumer Stocks

7 Conclusion

This study presented a hybrid CNN-Transformer model that integrates technical analysis indicators for stock market prediction. The architecture effectively captures both local and global dependencies, leading to superior predictive accuracy and risk-adjusted performance. Future work will explore integrating alternative features such as sentiment analysis and macroeconomic indicators. We also propose quantile based encoding of price changes in the stock values to ensure that the model can also learn the probabilites of the prices changing.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[2] J. Zou, Q. Zhao, Y. Jiao, H. Cao, Y. Liu, Q. Yan, E. Abbasnejad, L. Liu, and J. Q. Shi, “Stock market prediction via deep learning techniques: A survey,” 2023. [Online]. Available: https://arxiv.org/abs/2212.12717

[3] J. K. S. Pushpendu Ghosha, Ariel Neufeldb, “Forecasting directional movementsofstockprices for intraday trading using lstmandrandomforests,” 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1544612321003202

[4] A. Moghar and M. Hamiche, “Stock market prediction using lstm recurrent neural network,” *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020, the 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920304865

[5] E. Hoseinzade and S. Haratizadeh, “Cnnpred: Cnn-based stock market prediction using a diverse set of variables,” *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419301915

[6] C. Wang, Y. Chen, S. Zhang, and Q. Zhang, “Stock market index prediction using deep transformer model,” *Expert Systems with Applications*, vol. 208, p. 118128, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422013100

[7] Y. Han, Y. Liu, G. Zhou, and Y. Zhu, “Technical analysis in the stock market: A review,” *SSRN*, 2021. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3850494

[8] H. P. Kothari, N. Nahar, and A. Gautam, “Stock market prediction using cnn and lstm,” *International Journal of Creative Research Thoughts (IJCRT)*, vol. 11, no. 5, pp. 32–45, 2023.

[9] Hemant, A. K. Parida, R. Kumari, A. R. Singh, A. Bandyopadhyay, and S. Swain, “Stock market prediction under a deep learning approach using variational autoencoder, and kernel extreme learning machine,” pp. 156–161, 2023.

[10] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015. [Online]. Available: https://arxiv.org/abs/1505.04597