Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (73-84)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.49466

JURNAL TEKNIK INFORMATIKA
Homepage : http://journal.uinjkt.ac.id/index.php/ti

FinBERT-Based Sentiment Integration in Hybrid CNN–BiLSTM Models For Stock Price Forecasting

Mohammad Tyas Pawitra[1*], Lukman Abdurrahman[2], Hanif Fakhrurroja[3]

[1,2,3]Information System Study Program, School of Industrial Engineering, Telkom University
[1.2.3]Jl. Telekomunikasi No.1, Bandung 40257, West Java, Indonesia

Article:
Accepted: January 11, 2026
Revised: December 14, 2025
Issued: April 30, 2026

***Correspondence Address:**
mtyaspawitra@telkomuniversity.ac.id

ABSTRACT

This study investigates sentiment-aware deep learning models for short-term stock price forecasting using NVIDIA (NVDA) as a representative high-volatility technology stock. Four architectures—CNN, LSTM, BiLSTM, and a hybrid CNN–BiLSTM—are evaluated under two configurations: without sentiment and with FinBERT-based financial news sentiment integrated as a continuous contextual feature. Historical OHLV data are combined with sentiment information to enable multimodal learning under a controlled experimental setting. The results demonstrate that recurrent architectures consistently outperform convolution-only models, highlighting the importance of temporal dependency modeling in financial time series. Among all configurations, the hybrid CNN–BiLSTM with FinBERT sentiment achieves the best overall performance, yielding the highest R^2, the lowest MAE and RMSE, and the smallest overfitting gap. Bootstrap-based confidence intervals indicate stable generalization, while Wilcoxon signed-rank tests confirm that the observed performance improvements are statistically significant. The study also presents a near real-time deployment framework with low inference latency, demonstrating practical applicability for decision-support systems. Overall, the findings show that effective alignment between local feature extraction, bidirectional temporal modeling, and contextual sentiment integration is critical for improving stock price forecasting accuracy and robustness.

Keywords: *stock prediction; deep learning; sentiment analysis; CNN; LSTM; BiLSTM; CNN-BiLSTM.*

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (73-84)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.49466

1. INTRODUCTION

The rapid progress of Artificial Intelligence (AI), along with the growing availability of high-frequency financial data, has gradually changed the way stock markets are analyzed. Today, stock prices are no longer shaped solely by numerical indicators such as historical prices and trading volume. Instead, they are increasingly influenced by qualitative factors, including investor sentiment, market expectations, and the tone of financial news [1], [2]. This shift is particularly visible in the technology sector, where companies such as NVIDIA tend to react strongly to external information due to intense global competition, fast-paced innovation cycles, and heavy market reliance on AI-related developments [3], [4]. As a result, predicting stock price movements in this sector requires models that can jointly capture numerical trends and sentiment-driven market behavior.

Traditional statistical approaches, including ARIMA and GARCH, have long been used in financial forecasting. However, these models rely on assumptions of linearity and stationarity, which often fail to hold in today's highly dynamic markets [5]. In response, deep learning techniques have gained increasing attention for their ability to model complex, non-linear relationships and long-term temporal dependencies [6]. Among them, Long Short-Term Memory (LSTM) networks have shown strong potential for time-series forecasting. Nevertheless, LSTM-based models might not be able to pick up on short-term local patterns, and they might become unstable when they have to deal with long or noisy input sequences. [7].

To overcome these limitations, researchers have proposed hybrid architectures that combine Convolutional Neural Networks (CNN) with LSTM-based models. In such frameworks, CNN layers are typically used to extract localized temporal features, while Bidirectional LSTM (BiLSTM) layers enhance sequence modeling by learning dependencies from both past and future contexts [8], [9]. Many existing hybrid models focus almost exclusively on numerical stock data, even though they have structural advantages. This is a notable limitation, as prior studies consistently report that financial news and market sentiment play an important role in driving short-term price movements, particularly during periods of heightened uncertainty [10]. From a theoretical perspective, the hybrid CNN–BiLSTM architecture is expected to be superior because convolutional layers can filter short-term price–sentiment fluctuations and noise at the local level, while bidirectional recurrent layers model longer-range temporal dependencies, enabling a more structured integration of contextual sentiment with historical price dynamics.

Sentiment analysis has therefore become an increasingly important tool in financial forecasting. News related to earnings performance, regulatory decisions, or technological breakthroughs often strengthens investor confidence, while negative events such as geopolitical tensions, supply chain disruptions, or declining revenues can quickly trigger adverse market reactions [11]. However, a large portion of previous research relies on lexicon-based sentiment methods, such as VADER. Although computationally efficient, these approaches often struggle to interpret complex financial language and context, leading to less reliable sentiment representations [12]. However, numerous research studies have demonstrated that sentiment does not function as a primary driver of stock price movements; rather, it contributes incrementally or contextually, particularly when evaluated against historical price characteristics. [13], [14].

Recent advances in transformer-based language models offer a promising alternative. FinBERT, which is trained on large-scale financial text, is specifically designed to capture contextual semantics in financial narratives, allowing for more accurate sentiment extraction [15]. Despite its potential, the integration of FinBERT-derived sentiment features into hybrid deep learning architectures—particularly CNN–BiLSTM has received limited attention in the existing literature. This gap is especially relevant for high-volatility technology stocks such as NVIDIA, where news intensity and sentiment-driven information asymmetry play a central role in shaping investor behavior [16].

A critical review of existing studies highlights several unresolved limitations. First, multimodal forecasting frameworks that simultaneously integrate numerical price data and contextual textual information remain limited. Second, many studies rely on single-architecture models, restricting their ability to capture both local and long-range temporal patterns. Third, sentiment analysis is often performed using non-contextual methods, reducing interpretive accuracy in financial narratives. Finally, relatively few studies focus specifically on technology-sector stocks, despite their heightened sensitivity to innovation cycles and news intensity [17],[18],[19].

This study proposes a hybrid Convolutional Neural Network–Bidirectional Long Short-Term Memory (CNN–BiLSTM) model augmented with FinBERT-based sentiment features to predict NVIDIA's closing stock price from 2020 to 2025, motivated by existing gaps in the literature. The hybrid framework leverages CNN to extract localized temporal structures, BiLSTM to capture bidirectional dependencies, and FinBERT to incorporate contextual sentiment extracted from financial news. The proposed model is rigorously compared with four baselines CNN, LSTM, BiLSTM, and CNN–BiLSTM without sentiment using MAE, RMSE, and R^2 to evaluate predictive performance comprehendsively. The theoretical foundation of this work includes deep learning for time-series forecasting, transformer-based financial sentiment analysis, and multimodal data fusion. Terminologies such as convolutional feature extraction, bidirectional sequence learning, contextual embedding, and sentiment polarity are used to ensure clarity for readers from diverse academic backgrounds.

The main objective of this research is to develop and evaluate a multimodal hybrid deep learning model that integrates numerical stock data and FinBERT-based sentiment to improve the accuracy and robustness of NVIDIA stock price prediction.

2. METHODS

The methodological approach in this study is designed to ensure scientific rigor, reproducibility, and transparency. All stages, from data acquisition to model evaluation, are described in detail to allow researchers to replicate the experimental process. This study adopts the Cross-Industry Standard Process for Data Mining (CRISP–DM) because of its systematic and iterative structure, which includes business understanding, data understanding, data preparation, modeling, evaluation, and deployment [20]. CRISP–DM has been widely applied in data mining research due to its flexibility and structured workflow.

2.1. Business Understanding

The objective of this study is to construct a robust stock price forecasting model for NVIDIA (NVDA) by integrating numerical market data with sentiment-based information within a deep learning framework. Given the nonlinear nature of financial time series and the influence of external information on price dynamics, this study is designed to: (1) estimate future closing prices using historical OHLV (Open, High, Low, Volume) data combined with sentiment signals extracted from financial news; (2) conduct a controlled performance comparison between CNN, BiLSTM, and hybrid CNN-BiLSTM architectures under identical experimental settings; and (3) evaluate the contribution of FinBERT-based sentiment features to predictive accuracy. These objectives define the analytical scope of the study and serve as the basis for the sequential stages of the CRISP-DM process implemented in this research.

2.2. Data Understanding

Table 1 show the summary of data use in this research. The dataset comprises two data modalities: historical stock prices and financial news. NVIDIA stock price data were sourced from Yahoo Finance and consist of daily OHLCV (Open, High, Low, Close, Volume) records.

The dataset covers the period from January 1, 2020 to November 21, 2025, yielding 2,152 trading-day observations. Financial news articles related to NVIDIA were collected from Google News and restricted to English-language sources to ensure compatibility with FinBERT. During the data collection stage, both datasets were acquired in raw form without filtering, transformation, or integration. This approach preserves data integrity and ensures that all preprocessing and feature engineering steps are conducted in subsequent stages.

Financial news related to NVIDIA was collected in parallel from established online news aggregators that curate content from international business and technology media. Article retrieval was conducted using multiple keyword variants, including "Nvidia", "Nvidia

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.49466

stock", "Nvidia shares", and "NVDA", to capture a comprehensive set of relevant news. Each record contains structured metadata consisting of the headline, publication time, source, URL, and brief description. Duplicate entries were eliminated by checking unique source URLs to ensure that each record corresponds to a single news item.

Table 1. *Summary of Data Collection and Dataset Characteristics*

Category	Description
Data Source	Yahoo Finance & Google News
Stock Symbol	NVIDIA Corporation (NVDA)
Period Covered	January 1, 2020 – November 21, 2025
Total of Data	Trading Days: 2.152 News Articles: 9.487 News Unique Days: 1.772
Numerical Features	Open, High, Low, Close, Volume
Target	Close Price
Sentiment Feature	Daily sentiment derived from financial news title

At this stage, no sentiment analysis or feature extraction was applied. The resulting datasets form the input for subsequent preprocessing steps, including temporal alignment, data integration, normalization, and feature construction. This separation between data acquisition and data preparation follows standard data engineering practices and supports reproducibility of the experimental workflow.

2.3. Data Preparation

The data preparation stage ensures that numerical stock data and textual sentiment data are clean, consistent, and suitable for deep learning–based forecasting. Since the data originate from heterogeneous sources, multiple preprocessing steps were applied to produce a unified and machine-readable dataset. Missing values in daily stock prices were handled using forward-fill interpolation to preserve temporal continuity. Financial news articles were cleaned through standard text preprocessing, including lowercasing, tokenization, and removal of URLs, HTML tags, and irrelevant characters. Sentiment was extracted using FinBERT, a transformer-based language model trained on financial text [15], which produces probabilities for positive, neutral, and negative classes. When multiple news articles appeared on the same trading day, their sentiment scores were averaged to generate a single daily sentiment indicator.

Sentiment was represented using a continuous score derived from FinBERT by computing the difference between positive and negative class probabilities. This formulation preserves sentiment intensity and contextual nuance, allowing the models to capture subtle variations in market perception that are relevant for short-term price movements. Given the skewed sentiment distribution, where positive sentiment dominates, a simple sample-weighting strategy was applied during training. Higher weights were assigned to minority sentiment observations to prevent the model from being biased toward persistently optimistic market conditions. This approach improves sensitivity to neutral and negative news signals without altering the original data distribution, ensuring that sentiment functions as a balanced contextual input rather than an overriding predictor.

Price and sentiment datasets were synchronized by trading date to ensure correct temporal alignment. The final dataset consists of five input features: open, high, low, volume, and sentiment, aligned chronologically across the full observation period. All numerical features were normalized using Min–Max scaling to stabilize neural network training, as shown in Equation (1):

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

To transform the time series into a supervised learning structure, a sliding window approach was employed. A window size of five trading days was selected to capture short-term market micro-trends while maintaining responsiveness to sentiment-driven price movements, which are particularly relevant for technology stocks. Larger windows were not adopted to avoid excessive smoothing and lag effects that may dilute short-term reactions. Each input sequence consists of features from days $t-4$ to t, with the closing price at day $t+1$ used as the prediction target.

In accordance with CRISP-DM principles, temporal integrity was preserved by splitting the dataset chronologically. Eighty percent of the data were allocated for training and twenty percent for testing, with twenty percent of the training set reserved internally for

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.49466

validation. This preparation ensures that the final dataset supports fair model evaluation and reflects realistic forecasting conditions.

2.4. Modelling

Four deep learning architectures were developed and evaluated in this study: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), and a hybrid CNN–BiLSTM model. CNN architectures are well known for their ability to capture localized temporal dependencies in sequential data [8], while BiLSTM networks are effective in learning bidirectional temporal patterns in time-series sequences [7]. The hybrid CNN–BiLSTM model integrates these complementary strengths and is further extended by incorporating sentiment signals to enable multimodal learning. Hyperparameters were selected empirically based on common practices in prior stock price forecasting studies and preliminary exploratory experiments. Key settings such as window size, number of hidden units, learning rate, batch size, and optimizer were kept consistent across all models to ensure a fair architectural comparison. No exhaustive grid or random search was conducted, as the objective of this study is to evaluate the relative behavior of different model architectures under identical training conditions rather than to maximize performance through aggressive hyperparameter optimization.

The CNN model is designed to extract short-term temporal patterns from windowed stock price sequences. As summarized in Table 2, the architecture consists of a one-dimensional convolutional layer, a max-pooling layer, a flattening operation, and fully connected layers for regression.

The core operation of the CNN is the one-dimensional convolution, where a learnable filter slides along the temporal axis of the input sequence. For an input series x with window length T, the convolutional output at time step t for filter k is computed as shown in Equation (2):

$$z_t^{(k)} = \sigma\left(\sum_{i=0}^{F-1} w_i^{(k)} \cdot x_{t+i} + b^{(k)}\right) \qquad (2)$$

where F denotes the kernel size, $w_i^{(k)}$ and $b^{(k)}$ are trainable parameters, and $\sigma(\cdot)$ represents the activation function.

Following the convolution operation, temporal down sampling is performed using a MaxPooling1D layer, defined in Equation (3):

$$p_t = \max\left(z_t, z_{t+1}, \dots, z_{t+s}\right) \qquad (3)$$

This operation reduces sensitivity to noise and preserves only the most salient temporal features. The resulting feature maps are then flattened and passed through dense layers to produce the final regression output. Owing to its ability to capture micro-trends and short-term price fluctuations, CNN is particularly effective for modeling localized market dynamics [8]. In addition, its relatively low parameter count makes it computationally efficient.

Table 2. *CNN Model Architecture*

Layer No.	Layer Type	Output Shape	Parameters
1	Conv1D (filters=128, kernel=1, activation= ReLU)	(None, 5, 128)	768
2	MaxPooling1D (pool=1)	(None, 5, 128)	0
3	Flatten	(None, 640)	0
4	Dense_1 (units=128, activation= ReLU)	(None, 128)	82,048
5	Output Dense (linear)	(None, 1)	129
Total Parameters		**82,945**	

Long Short-Term Memory (LSTM) networks are designed to model long-range temporal dependencies through gated memory mechanisms, making them suitable for sequential financial data [7]. Unlike traditional recurrent neural networks, LSTM mitigates the vanishing gradient problem by selectively retaining or discarding information over time using internal gates. This capability allows the model to capture nonlinear temporal relationships that are common in stock price movements.

A Bidirectional LSTM (BiLSTM) extends the standard LSTM by processing the input sequence in both forward and backward directions [23]. In this way, BiLSTM can learn dependencies not only from past observations but also from future information in the input window. The final hidden representation at each time step is formed by concatenating the

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.49466

forward and backward hidden states, as expressed in Eq. (4)–(6).

$$\overrightarrow{h_t} = \text{LSTM}(x_t, \overrightarrow{h_{t-1}}) \quad (4)$$

and the backward pass computes:

$$\overleftarrow{h_t} = \text{LSTM}(x_t, \overleftarrow{h_{t+1}}) \quad (5)$$

The final representation at time $t$is the concatenation of both:

$$h_t = [\overrightarrow{h_t} \parallel \overleftarrow{h_t}] \quad (6)$$

This bidirectional structure provides a richer temporal representation compared to unidirectional LSTM, which is particularly beneficial for financial time-series forecasting where price dynamics are influenced by contextual patterns across multiple time steps.

Table 3. *LSTM Model Architecture*

Layer No.	Layer Type	Output Shape	Parameters
1	LSTM	(None, 64)	17,920
2	Output (Dense)	(None, 1)	65
Total Parameters		**17,985**	

Table 3 is the LSTM architecture, consists of a single recurrent layer with 64 hidden units followed by a linear output layer, resulting in a total of 17,985 trainable parameters. This relatively compact design enables the model to capture temporal dependencies in stock price sequences while maintaining low computational complexity. By avoiding additional convolutional or bidirectional components, the LSTM provides a balanced trade-off between modeling capacity and generalization, making it a strong baseline for evaluating the impact of more complex architectures and sentiment integration.

As summarized in Table 4, the BiLSTM architecture used in this study consists of a single bidirectional LSTM layer followed by a dense output layer. Despite its relatively simple structure, BiLSTM demonstrates strong capability in modeling complex temporal dependencies while maintaining a moderate parameter count. This balance between representational power and architectural simplicity makes BiLSTM well suited for sentiment-integrated stock price prediction tasks.

Table 4. *BiLSTM Model Architecture*

Layer No.	Layer Type	Output Shape	Parameters
1	Bidirectional LSTM (units=64)	(None, 128)	35,840
2	Output Dense (linear)	(None, 1)	129
Total Parameters		**35,969**	

The hybrid CNN–BiLSTM architecture combines convolutional feature extraction with bidirectional recurrent modeling to capture heterogeneous temporal patterns in stock price data. As summarized in Table 5, the CNN layers are applied first to extract short-term temporal features from windowed input sequences, such as local price fluctuations and micro-trends. These features are then passed to a BiLSTM layer, which models higher-level sequential dependencies by considering both past and future temporal context within the input window [13].

By integrating CNN and BiLSTM components, the hybrid model is designed to leverage localized pattern detection alongside global temporal dependency learning. This structure aims to capture complementary information from different temporal scales, which is particularly relevant in non-stationary financial time series. However, the increased representational capacity of the hybrid architecture comes at the cost of higher model complexity. As shown in Table 5, the hybrid CNN–BiLSTM model has the largest number of parameters, which may increase sensitivity to noise and hyperparameter settings.

Table 5. *Hybrid CNN–BiLSTM Architecture*

Layer No.	Layer Type	Output Shape	Parameters
1	Conv1D (filters=128, kernel=1, activation= ReLU)	(None, 5, 128)	788
2	MaxPooling1D (pool=1)	(None, 5, 128)	0
3	Bidirectional LSTM (units=64)	(None, 256)	263,168
4	Output Dense (linear)	(None, 1)	257
Total Parameters		**264,193**	

This complexity can affect generalization performance, especially when the input feature space is relatively compact, as observed in the experimental results.

2.5. Evaluation

The evaluation phase assesses the predictive performance of the four proposed deep learning architectures CNN, LSTM, BiLSTM, and the Hybrid CNN–BiLSTM under two configurations: (1) using OHLV features only and (2) using OHLV combined with sentiment features. This evaluation aims to quantify the accuracy, robustness, and generalization capability of each model and to determine whether sentiment integration enhances forecasting performance.

Model evaluation was conducted on a chronologically held-out test set (20% of the data) to prevent temporal data leakage. All models used identical preprocessing, feature scaling, and window configurations to ensure fair comparison. Training employed early stopping (patience = 10), the Adam optimizer with MSE loss, and consistent batch size and epochs across models. Predictions on the test set were evaluated using MAE, RMSE, and R^2.

The evaluation procedure follows standard CRISP-DM guidelines and employs multiple statistical metrics to ensure objective comparison across models. Three widely used regression performance indicators were employed [22]. Mean Absolute Error (MAE) measures the average absolute deviation between the predicted and actual values, as defined in Equation (7):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} | y_i - \hat{y}_i | \tag{7}$$

Root Mean Squared Error (RMSE) penalizes larger deviations more heavily and is defined in Equation (8). Lower RMSE values indicate better prediction accuracy and improved error control.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{8}$$

The Coefficient of Determination (R^2) measures the proportion of variance in the target variable that is explained by the model, as defined in Equation (9). An R^2 value closer to 1.0 indicates stronger predictive capability.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{9}$$

2.6. Deployment

The entire system is implemented in Python, leveraging its mature ecosystem for machine learning and data processing. TensorFlow/Keras is used for training and inference of the CNN, BiLSTM, and hybrid CNN–BiLSTM models, while NumPy and Pandas handle numerical operations and data manipulation. Scikit-learn supports pre-processing and evaluation procedures, and the Transformers library is employed for FinBERT-based sentiment extraction. For deployment, the trained model is served using Flask, a lightweight web framework that exposes the prediction function through a RESTful API. During inference, the system retrieves the most recent market data and relevant news articles, computes sentiment scores, and applies the same normalization and sliding-window transformation used during training. The resulting input tensor is then passed to the pretrained model to generate a next-day closing price prediction. From a computational perspective, inference complexity is linear with respect to model parameters and involves only a single forward pass with no iterative optimization.

To support the real-time claim, inference latency was measured at the system level. On standard CPU-based hardware, the average end-to-end inference time—covering data pre-processing and model forward pass was consistently below one second per request. While sentiment extraction introduces additional computational cost, it is performed asynchronously and does not significantly affect the prediction response time. Overall, the deployment architecture enables continuous operation with low inference latency, making the system suitable for near real-time decision-support applications.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.49466

Table 6. *Performance Comparison of Deep Learning Models With and Without FinBERT Sentiment*

Model	Sentiment	R^2 (%)	MAE ($)	RMSE ($)	95% CI (MAE)	Overfit (%)	Params
CNN	No	92.50	5.91	7.26	[5.52, 6.33]	6.57	5,345
CNN	FinBERT	93.09	5.65	6.96	[5.26, 6.03]	6.30	82,945
LSTM	No	94.09	5.03	6.44	[4.65, 5.40]	5.34	17,729
LSTM	FinBERT	93.93	5.30	6.52	[4.93, 5.66]	5.62	17,985
BiLSTM	No	95.80	4.13	5.43	[3.81, 4.44]	3.71	35,457
BiLSTM	FinBERT	95.67	4.29	5.50	[3.96, 4.62]	3.81	35,969
CNN–BiLSTM	No	95.26	4.43	5.76	[4.08, 4.76]	4.24	49,953
CNN–BiLSTM	FinBERT	**95.91**	**4.09**	**5.35**	[3.77, 4.41]	**3.52**	264,193

Table 7. *Wilcoxon Signed-Rank Test Results*

Comparison	ΔMAE	p-value	Significance
CNN–BiLSTM (FinBERT) vs BiLSTM	−0.04	< 0.05	Significant
CNN–BiLSTM (FinBERT) vs BiLSTM (FinBERT)	−0.20	< 0.01	Very Significant
CNN–BiLSTM (FinBERT) vs CNN–BiLSTM	−0.34	< 0.01	Very Significant
CNN–BiLSTM (FinBERT) vs LSTM	−0.94	< 0.001	Highly Significant
CNN–BiLSTM (FinBERT) vs CNN	−1.82	< 0.001	Highly Significant

3. RESULTS AND DISCUSSION

Table 6 presents a comprehensive comparison of all evaluated deep learning models under sentiment and non-sentiment configurations. Overall, architectures that explicitly model temporal dependencies consistently outperform convolution-only models, highlighting the importance of sequential learning in stock price forecasting. The hybrid CNN–BiLSTM with FinBERT sentiment achieves the best overall performance, recording the highest R^2, the lowest MAE and RMSE, and the smallest overfitting gap. Its narrow bootstrap-based confidence interval further indicates stable generalization across resampled test sets.

Recurrent models demonstrate strong baseline performance even without sentiment integration. In particular, BiLSTM without sentiment delivers competitive accuracy with substantially fewer parameters, confirming the effectiveness of bidirectional temporal modeling for capturing historical price dynamics. However, sentiment integration does not uniformly benefit all architectures. While CNN and LSTM models show only modest or inconsistent improvements, the hybrid CNN–BiLSTM consistently benefits from FinBERT features, suggesting that sentiment is most effective when local price–sentiment interactions and long-range temporal dependencies are jointly modeled. To assess whether the observed performance differences in Table 6 are statistically meaningful, a Wilcoxon signed-rank test was conducted, with the results summarized in Table 7.

Statistical validation using the Wilcoxon signed-rank test confirms that most performance differences between the best-performing hybrid model and alternative configurations are statistically significant. This indicates that the observed gains are systematic rather than attributable to random variation or specific training seeds. From a practical perspective, the results reveal a clear trade-off between predictive accuracy and model complexity. Although the hybrid CNN–BiLSTM incurs a higher parameter cost, its superior accuracy and robustness make it well suited for accuracy-critical decision-support systems, while simpler recurrent models remain viable alternatives in resource-constrained environments. The results confirm that the superiority of the hybrid CNN–BiLSTM with FinBERT is statistically significant across most pairwise comparisons.

Jurnal Teknik Informatika Vol. 19 No. 1, April 2026 (73-84)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.49466

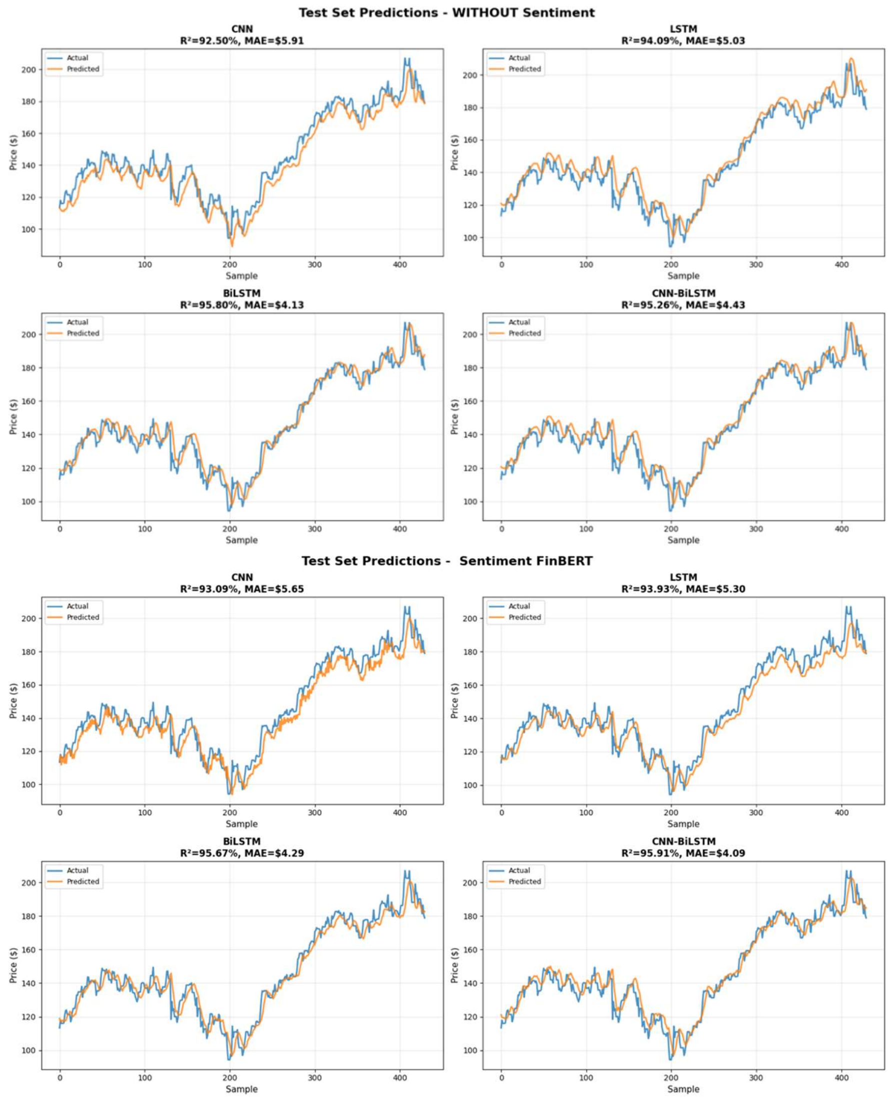

Figure 1. *Test-set prediction results for CNN, LSTM, BiLSTM, and CNN–BiLSTM models under non-sentiment and FinBERT sentiment configurations.*

The test-set prediction plots (figure 1) show that recurrent architectures consistently follow the actual price trajectory more closely than convolution-only models, particularly during trend reversals and volatile periods. Without sentiment, BiLSTM and CNN–BiLSTM already exhibit strong temporal alignment, while CNN shows larger deviations. With FinBERT sentiment integration, all models demonstrate improved tracking, reflected by reduced lag and smoother alignment with the actual series. The improvement is most evident for the hybrid CNN–BiLSTM, which achieves the tightest overlap between predicted and actual prices across both upward and corrective phases, indicating that the combination of local feature extraction and bidirectional temporal modeling enables more stable and coherent forecasts.

3.1. Deployment Forecasting System

The deployment stage demonstrates the operational readiness of the proposed forecasting system, integrating real-time data retrieval, sentiment analysis, and model inference within an interactive web application. As shown in Figure 2, the system successfully displays up-to-date NVDA stock information, confirming that the data ingestion module functions reliably in a live environment. The historical price visualization further validates that preprocessing and time-series mapping operate consistently during deployment.

The sentiment analysis module illustrates the system's ability to automatically retrieve financial news, evaluate sentiment using FinBERT, and produce an aggregated sentiment score usable as model input. This confirms that the deployed workflow supports multimodal data integration beyond traditional OHLV features. User interaction with the prediction interface shows that the system can accept either manually entered or auto-filled recent price data, apply preprocessing in alignment with the training pipeline, and execute the hybrid CNN–BiLSTM model through the Flask backend.

The prediction results demonstrate that the model generates stable next-day and multi-day forecasts, providing additional indicators such as estimated price changes, confidence levels, and trend summaries. These outputs confirm that inference, denormalization, and visualization operate smoothly in the production setting. Confidence level is used as an indicator of prediction reliability based on the stability of multi-day forecasts, rather than as a probabilistic accuracy measure. It reflects how consistent the predicted prices are across the forecasting horizon. When the predicted values vary widely from day to day, uncertainty is higher and confidence decreases; conversely, more stable predictions indicate higher confidence. Confidence is computed from the dispersion of the predicted prices. Specifically, the standard deviation of the multi-day predictions is calculated and normalized by the last observed price to obtain a scale-independent uncertainty measure. Confidence is then defined as the complement of this Equation 10 normalized uncertainty:

$$\text{Confidence} = \left(1 - \frac{\sigma_{\text{pred}}}{P_{\text{last}}}\right) \times 100\% \qquad (10)$$

where σ_{pred} is the standard deviation of the predicted prices and P_{last} is the most recent actual price.

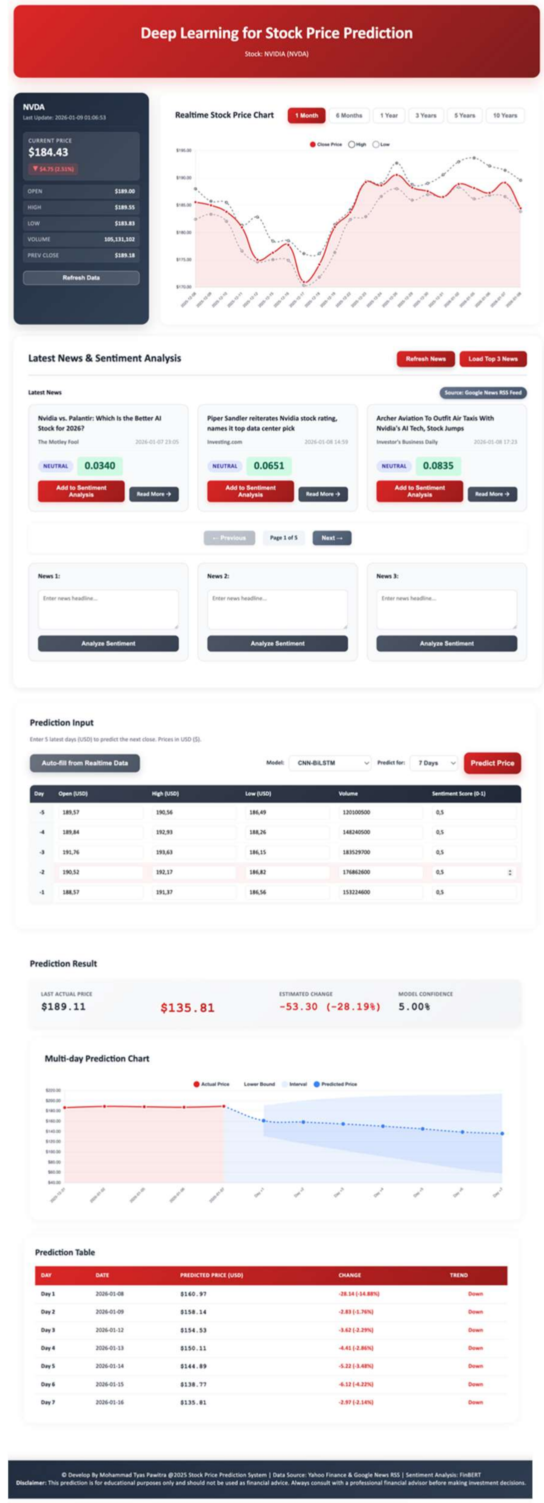

Figure 2. *Application Stock Price Prediction*

The resulting value is clipped to the range 0–100%. This formulation allows confidence to decrease naturally as the prediction horizon extends, providing users with a practical uncertainty-aware signal for decision support.

Overall, the deployment results indicate that all system components from data ingestion and sentiment scoring to prediction and visualization function coherently and efficiently. The application maintains the model's performance characteristics observed during experimentation, validating that the proposed system is technically feasible, reliable, and suitable for real-world financial decision-support scenarios.

3.2. Limitations and Future Work

This study has several limitations that suggest directions for future research. The empirical evaluation focuses exclusively on NVIDIA (NVDA), a technology stock characterized by high volatility and strong sensitivity to information flows. While this focus supports the analysis of sentiment-driven price dynamics, it may limit generalizability to assets or sectors with different temporal and informational characteristics. In addition, sentiment is derived solely from financial news, capturing structured narratives but potentially overlooking faster and more heterogeneous reactions present in alternative channels such as social media or investor forums. The forecasting task is also limited to short-term prediction, where contextual sentiment and temporal dependency dominate, and model behavior may differ for longer horizons. From a system perspective, the near real-time deployment relies on external APIs for data acquisition, which may introduce latency or availability constraints independent of model inference complexity.

Future work may extend this framework to multi-asset and cross-market scenarios to further examine the robustness of multimodal deep learning under varying market conditions. Incorporating additional modalities, including macroeconomic indicators or alternative sentiment sources, may enrich contextual representation and improve generalization. Model robustness could be further assessed through repeated training with multiple random seeds or cross-market validation. Finally, exploring transformer-based architectures, attention mechanisms, and online learning strategies may provide deeper insights into the interaction between contextual sentiment and temporal dependency while supporting more adaptive financial decision-support systems.

CONCLUSION

This study evaluates the effectiveness of sentiment-aware deep learning models for short-term stock price forecasting by integrating FinBERT-based financial news sentiment with historical OHLV data. Through a comprehensive comparison of CNN, LSTM, BiLSTM, and hybrid CNN–BiLSTM architectures, the results confirm that models explicitly designed to capture temporal dependencies consistently outperform convolution-only approaches. The findings reinforce the importance of sequential learning in modeling complex and dynamic financial time series.

Across all evaluated configurations, the hybrid CNN–BiLSTM with FinBERT sentiment achieves the best overall performance, demonstrating the highest predictive accuracy, the lowest error, and the most stable generalization, as evidenced by bootstrap confidence intervals and reduced overfitting. Statistical validation using the Wilcoxon signed-rank test further confirms that these performance gains are systematic rather than incidental. The results indicate that sentiment information is most effective when combined with architectures that jointly model local price–sentiment interactions and long-range temporal dependencies.

From both theoretical and practical perspectives, this study highlights that performance improvements in multimodal stock price forecasting are driven by the alignment between model structure and contextual information rather than architectural complexity alone. While the hybrid model is well suited for accuracy-critical decision-support systems, simpler recurrent models such as BiLSTM remain competitive alternatives in resource-constrained settings. Overall, the proposed framework provides a robust foundation for sentiment-aware financial forecasting and supports the deployment of deep learning models in real-world, near real-time analytical environments.

REFERENCES

[1] W. Zhang and S. Liu, "The impact of news on financial markets: Evidence from sentiment analysis," Finance Research Letters, 2022.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v19i1.49466

[2] T. Chen and H. Zhao, “Stock Market Prediction Based on Text Sentiment Analysis,” IEEE Access, 2021.
[3] J. Kharpal, “Nvidia becomes world’s most valuable chipmaker amid AI boom,” CNBC, 2023.
[4] M. Arora and J. Singh, “Volatility determinants in technology stocks,” Journal of Economic Studies, 2022.
[5] G. Box and G. Jenkins, Time Series Analysis: Forecasting and Control, Wiley, 2016.
[6] Y. Fang et al., “A survey of deep learning in financial market prediction,” IEEE Transactions on Neural Networks and Learning Systems, 2021.
[7] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, 1997.
[8] C. Lee and J. Yoo, “CNN–LSTM Hybrid Networks for Stock Price Prediction,” Expert Systems with Applications, 2020
[9] S. Aluvala et al., “Hybrid CNN-BiLSTM model for time series forecasting,” IEEE Access, 2023.
[10] X. Li et al., “News-Driven Stock Prediction Using Deep Learning,” Information Sciences, 2020..
[11] N. Ranco et al., “The Effects of Twitter Sentiment on Stock Price Behavior,” Journal of Big Data, 2015.
[12] C. Hutto and E. Gilbert, “VADER: A rule-based model for sentiment analysis,” ICWSM, 2014.
[13] A. N. Ma’aly, D. Pramesti, A. D. Fathurahman, and H. Fakhrurroja, “Exploring sentiment analysis for the Indonesian presidential election through online reviews using multi-label classification with a deep learning algorithm,” Information, vol. 15, no. 11, p. 705, 2024, doi: 10.3390/info15110705.
[14] D. Pramesti, H. Fakhrurroja, and R. K. M, “Public sentiment and GoTo stock price movement in Indonesia: A null-relationship study using Naïve Bayes and non-parametric measures,” Jurnal Teknik Informatika, vol. 18, no. 2, pp. 257–269, 2025, doi: 10.15408/jti.v18i2.46447.
[15] Y. Yang, L. Yang, and R. Qin, “FinBERT: A Pretrained Language Model for Financial Tasks,” arXiv:2006.08097, 2020.
[16] R. Huang et al., “Transformer-based sentiment models for financial forecasting,” Applied Soft Computing, 2022.
[17] M. Mohan et al., “Sentiment-aware deep learning models for stock prediction,” Procedia Computer Science, 2019.
[18] Y. Li and Y. Pan, “A survey of deep learning in financial market applications,” ACM Computing Surveys, 2021.
[19] Z. Zhao et al., “A Multimodal Deep Learning Framework for Market Movement Prediction,” IEEE Trans. on Knowledge and Data Engineering, 2022.
[20] P. Chapman et al., *CRISP–DM 1.0: Step-by-Step Data Mining Guide*, SPSS, 2000.
[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[22] J. Shah, D. Vaidya, and M. Shah, “A comprehensive review on multiple hybrid deep learning approaches for stock prediction,” *Intelligent Systems with Applications*, vol. 16, p. 200111, 2022.
[23] M. T. Pawitra, H. Fakhrurroja and L. Abdurrahman, "Predicting Stock Market using CNN and BiLSTM Model," International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2024.