Available Online: 16 September 2026 DOI: 10.54254/2977-5701/2026.36910

Integrating FinBERT based news sentiment and technical indicators for LSTM stock price forecasting

Yuzheng Zhao

SWUFE-UD Institute of Data Science, Southwestern University of Finance and Economics, Chengdu, China

zyz051229@163.com

Abstract. Accurate stock price forecasting is important for investment and risk management but remains challenging due to complex interactions among market dynamics, firm-specific information, and investor sentiment. Although technical indicators capture historical price patterns, they may not fully reflect information conveyed through financial news. This study develops an LSTM-based framework that integrates technical indicators with FinBERT-derived news sentiment for next-day stock price forecasting. The model uses closing price, EMA26, MACD, RSI14, and daily sentiment scores extracted from 58,466 GDELT news headlines for Amazon, Apple, Google, Microsoft, and NVIDIA from 2019 to 2023. Company-level sentiment is temporally aligned with market features, and 15-day sequences are constructed to predict the normalized next-trading-day closing price. Out-of-sample evaluation from July to December 2023 reveals heterogeneous performance across firms: Apple achieves the lowest RMSE and MAE (0.0147 and 0.0110), while NVIDIA records the highest (0.0245 and 0.0201). Predictions generally capture temporal price dynamics but are less responsive to abrupt turning points. The findings demonstrate a systematic approach to integrating financial news sentiment and technical indicators for stock price forecasting while highlighting substantial cross-firm variation in predictive performance.

Keywords: stock price forecasting, LSTM, FinBERT, sentiment analysis, technical indicators, financial news

1. Introduction

Accurate stock price forecasting supports portfolio allocation, trading strategy formulation, and financial risk management. The task remains difficult because stock prices reflect interactions among firm performance, macroeconomic conditions, market liquidity, investor expectations, and newly released public information. These influences are nonlinear, time varying, and sensitive to changes in market regimes; consequently, historical closing prices alone provide an incomplete representation of the information available to market participants. Technical indicators derived from price histories can summarize trend and momentum, thereby making recurrent market patterns more accessible to forecasting models. Their informational scope nevertheless remains endogenous to past market behavior. Financial news offers a complementary, exogenous source of information concerning earnings, products, corporate management, regulation, competition, and industry conditions. Because such information may alter expectations before its implications are fully reflected in historical price patterns, a forecasting framework that combines structured market variables with finance-specific news sentiment has the potential to provide a more comprehensive representation of daily market conditions [1-7].

Stock price forecasting methods have developed from classical statistical specifications to machine-learning and deep-learning architectures. Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), and Vector Autoregression (VAR) models provide interpretable representations of autocorrelation, volatility, and multivariate dependence, but their fixed assumptions can limit their ability to accommodate nonlinear responses and abrupt information shocks. Machine-learning methods relax linearity assumptions and can model complex relationships among engineered predictors, although their performance remains sensitive to feature construction and temporal alignment [8]. Long Short-Term Memory (LSTM) networks use gated memory to preserve relevant sequential information [9-12]. Prior research has demonstrated their effectiveness in financial forecasting applications. In parallel, financial-text analysis initially relied on lexicon-based methods before progressing to contextual transformer models [13, 14]. FinBERT, a finance-domain adaptation of Bidirectional Encoder Representations from Transformers (BERT), enables context-sensitive sentiment extraction from financial language [13]. Recent studies have integrated financial-news sentiment with historical prices and technical indicators in LSTM-based stock price forecasting models [15-21]. Collectively, this literature supports the informational relevance of textual sentiment and structured market features. However, the evidence is produced under heterogeneous markets, companies, text sources, prediction targets, aggregation procedures, and evaluation metrics. Many studies further concentrate on an index, a single market, or a limited firm sample. Consequently, the company-level forecasting behavior of a consistent specification that combines daily FinBERT sentiment with identical technical and price features remains insufficiently characterized across multiple stocks.

To address this gap, this study constructs a unified daily forecasting framework for Amazon, Apple, Google, Microsoft, and NVIDIA. Company-related headlines from the Global Database of Events, Language, and Tone (GDELT) are processed with FinBERT to extract sentiment scores, which are aggregated by company and date. The resulting average daily sentiment score is aligned with the closing price obtained from Yahoo Finance and three technical indicators: the 26-day Exponential Moving Average (EMA26), Moving Average Convergence Divergence (MACD), and the 14-day Relative Strength Index (RSI14). These five features form 15-day input sequences for an LSTM model that directly predicts the min-max-normalized closing price of the next trading day. Forecasting performance is evaluated consistently for each company using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and predicted-versus-actual test-set curves.

To enable a consistent comparison, this study applies the same five-feature input specification and forecasting procedure to Amazon, Apple, Google, Microsoft, and NVIDIA, which differ in price dynamics and news coverage. The study contributes to the literature in three respects. First, the study establishes a systematic company–date alignment procedure that links daily FinBERT sentiment scores derived from company-related news with the corresponding market observations. Second, the study implements a feature-level integration strategy in which FinBERT-derived sentiment, closing price, EMA26, MACD, and RSI14 form the inputs to an LSTM model, while sentiment extraction and closing-price forecasting remain methodologically distinct stages. Third, the study applies a consistent preprocessing, sequence-construction, hyperparameter-selection, and evaluation protocol across all five stocks, thereby permitting a comparable assessment of forecasting performance and an analysis of cross-stock differences in predictive error.

2. Related work

2.1. Existing methods for time-series analysis

Stock price forecasting has been studied using classical statistical models, conventional machine learning, and deep learning. Classical approaches such as ARIMA, GARCH, and VAR remain useful because their assumptions and estimated parameters are relatively transparent. ARIMA models represent autocorrelation and linear temporal structure, GARCH models describe changing volatility, and VAR models examine interactions among several time series. A broad survey of financial time-series forecasting summarizes these traditional approaches and positions them as interpretable benchmarks for more flexible learning models [8]. Their interpretability makes them suitable for hypothesis testing and stable financial processes. However, stock markets contain nonlinear relationships, abrupt regime changes, and responses to external information that are difficult to express through fixed linear specifications. These limitations become more important when the objective is to combine prices with unstructured news information.

Machine learning methods, including Random Forest, support vector regression, and XGBoost, relax linear assumptions and can learn complex relationships among engineered predictors. A comprehensive survey reviews the theoretical foundations and empirical applications of these models in financial time-series forecasting, providing general support for their use in this domain [8]. They are often effective when technical indicators, lagged returns, and sentiment scores have already been transformed into structured features. A BERT-based investigation uses sentiment estimates together with historical market information and applies a Random Forest classifier to predict the direction of the S&P 500 [5]. An MLP-based approach combines historical prices with sentiment scores generated by VADER, TextBlob, and Flair, showing that sentiment-enhanced features can support both next-day and next-week trend prediction [2]. Although these models capture nonlinear effects, their performance depends on feature selection and they do not inherently preserve long sequential dependencies.

Deep learning reduces this dependence on manual feature construction by learning temporal representations directly from data. Recurrent neural networks process observations sequentially. The LSTM architecture uses gates to retain useful information and reduce the vanishing-gradient problem [12]. Related GRU models apply comparable gating principles. Price-only evidence from S&P 500 constituents further demonstrates the value of LSTM for financial sequences [15]. Temporal CNNs capture local patterns efficiently through convolution, whereas Transformers model distant relationships through self-attention. A comparative analysis reports that both LSTM and GRU benefit from the inclusion of financial news sentiment, with neither architecture showing a universal advantage [10]. LSTM is especially relevant to the present research because it can process ordered market observations while preserving information from earlier trading days. Its limitations include sensitivity to hyperparameters, substantial data requirements, possible overfitting, and limited interpretability. Recent research consequently favors hybrid models, attention mechanisms, transfer learning, and multimodal inputs. This transition is illustrated by models that combine sequential forecasting with finance-specific language models, attention or decomposition mechanisms, technical indicators, and multiple textual sources [1, 6, 11].

2.2. Impact of financial news on stock prices

Financial news affects prices because it changes expectations about firms, industries, and the wider economy. Sentiment analysis converts this qualitative information into variables that can be incorporated into forecasting models. Early approaches rely on sentiment lexicons, which assign polarity scores to words or phrases. Domain-specific dictionaries, such as the Loughran-McDonald dictionary, improve on general lexicons because words that appear negative in ordinary language may carry different meanings in financial reports [14]. A Hong Kong market application combines technical indicators with news sentiment vectors in a layered LSTM framework and shows that a finance-specific dictionary produces larger performance improvements than general dictionaries. This finding indicates that the quality of sentiment representation matters as much as the forecasting architecture [4].

Classical machine learning expands sentiment analysis beyond fixed dictionary counts. One investigation assembles five years of daily prices for S&P 500 companies and more than 265,000 related financial news articles, then compares support vector machines, Naive Bayes methods, and deep learning. The results indicate that historical prices or text alone provide an incomplete representation of market behavior [9]. Another approach compares sentiment scores from VADER, TextBlob, and Flair before supplying them to an MLP regressor, reporting support for both next-day and next-week trend prediction [2]. These studies support the value of combining textual and numerical features, but general sentiment tools may miss context, negation, and finance-specific meanings. Their forecasting models also depend on how news is aggregated and aligned with trading dates. Earlier multimodal research also combined distributed textual representations with numerical market variables in an LSTM-based stock-prediction framework [16].

Deep learning methods address part of this problem by learning representations from text and price sequences. A sentiment-aware LSTM combines investor sentiment with empirical mode decomposition and attention [6]. Sentiment contributes external information, decomposition separates price movements at different scales, and attention helps the model focus on useful temporal features. A BERT-based approach identifies sentiment in financial news and PTT forum discussions, then integrates these signals with historical transactions in an LSTM forecasting model. This design connects fundamental, social, and technical information, although its evidence is specific to the Taiwanese data environment [7]. Another multi-source framework combines historical prices, technical indicators, financial news, and stock posts. A CNN-based sentiment method produces an investor sentiment index that is supplied with structured market variables to an LSTM model for five companies in the Chinese A-share market [11]. These results show that multimodal models can outperform single-source prediction, but their conclusions remain tied to particular markets and textual platforms. Related event-driven work encodes structured news events with neural models, applies attention to distinguish influential reports from noisy information, and combines social-media text with historical prices [17-19].

Transformer-based language models provide a stronger representation of contextual meaning. One investigation fine-tunes BERT on labeled financial text and compares sentiment derived from news titles, article content, and their combination. The generated scores improve S&P 500 direction prediction when used in a trading rule or Random Forest classifier [5]. FinBERT further adapts transformer representations to financial language. A FinBERT-LSTM framework combines sentiment extracted from financial news with recent NASDAQ-100 price information and weighted market, industry, and company news categories, reporting better forecasting performance than LSTM and DNN models based only on prices [1]. Another approach combines VADER sentiment with latent Dirichlet allocation to examine FTSE 100 returns and volatility. It finds that different textual sources carry different predictive signals and achieves 63% accuracy for volatility direction through sentiment and topic features [3]. This result indicates that information value depends on source, timing, and target variable. Across the ten reviewed studies, the common finding is that financial text adds information that is absent from historical prices, while domain adaptation and temporal alignment determine how effectively that information supports forecasting. Complementary evidence integrates technical analysis with sentiment embeddings, while a FinBERT-LSTM investigation of the S&P 500 reports lower MSE, RMSE, and MAE than an otherwise comparable no-sentiment model [20, 21].

2.3. Current limitations and research gap

Despite broad evidence supporting sentiment-enhanced forecasting, several limitations remain. First, reducing each article to a positive, negative, or neutral label can oversimplify complex financial information. Identical polarity may reflect different events, levels of uncertainty, or implications for future cash flows. General lexicons and conventional classifiers are particularly vulnerable to this problem, while BERT and FinBERT improve contextual interpretation without eliminating information loss when their outputs are compressed into daily sentiment scores. Second, textual models and temporal forecasting models are often developed as separate stages. Sentiment may be calculated without careful alignment to trading days, publication timing, or company-level price sequences. Evidence shows that headlines, full articles, and social media can exhibit different relationships with returns and volatility [3]. A consistent alignment procedure is therefore necessary before textual signals are used as time-series inputs.

Third, the reviewed evidence is fragmented across markets and prediction targets. Existing evidence spans the United Kingdom and Hong Kong markets, Taiwan, and Chinese A-share companies [3, 4, 7, 11]. Some studies target S&P 500 direction, whereas others examine different price or trend forecasting tasks [1, 2, 5-7, 9-11]. Differences in data periods, sentiment sources, aggregation rules, and evaluation metrics make direct comparison difficult. Several studies also examine a single index, a small set of firms, or one textual platform. Consequently, evidence from one setting may not transfer directly to companies with different price scales and news coverage (The comparison could be seen in Table 1).

Table 1. Comparative study

Ref.	Study and market/data	Sentiment and forecasting method	Main finding and principal limitation
[1]	Gu et al.; NASDAQ-100 and financial news	FinBERT with weighted market-, industry-, and company-level news; FinBERT–LSTM	Finance-specific sentiment plus price sequences outperformed price-only LSTM and DNN models. Limitation: one index and a limited news environment.
[2]	Maqbool et al.; four companies from different sectors	VADER, TextBlob, and Flair; MLP regressor	Sentiment-enhanced inputs improved next-day and next-week trend prediction, with some configurations approaching 90%. Limitation: small company sample and manually aggregated signals.
[3]	Deveikyte et al.; FTSE 100 with Reuters, RavenPack, and Twitter	VADER and latent Dirichlet allocation; topic and volatility classification	Different sources contained different predictive signals; sentiment and topics achieved 63% volatility-direction accuracy. Limitation: volatility direction rather than company closing-price regression.
[4]	Li et al.; Hong Kong market	Four sentiment lexicons, including Loughran–McDonald; layered LSTM and fully connected network	Technical indicators plus sentiment performed best, and the finance-specific lexicon produced the largest gain. Limitation: lexicon methods have restricted contextual understanding.

Table 1. Continued

[5]	Fazlija and Harder; S&P 500 with Bloomberg and Reuters news	Fine-tuned BERT; Random Forest and trading strategy	Article-content sentiment improved index-direction prediction. Limitation: index direction rather than company-level price regression.
[6]	Jin et al.; stock prices and investor sentiment	Sentiment index; empirical mode decomposition, attention, and LSTM	The hybrid design improved closing-price prediction. Limitation: sentiment was not produced by a finance-specific transformer.
[7]	Ko and Chang; Taiwan market with news and PTT discussions	BERT; LSTM	Integrating textual sentiment with historical transactions reduced forecasting error. Limitation: evidence is tied to the Taiwanese market and PTT platform.
[9]	Mohan et al.; S&P 500 and more than 265,000 news articles	Machine-learning sentiment; SVM, Naive Bayes, and deep learning	Joint text and price information was more informative than either source alone. Limitation: limited finance-specific contextual language modeling.
[10]	Shahi et al.; stock features and financial news	News sentiment; LSTM and GRU	Both recurrent architectures benefited from sentiment, with no universal winner. Limitation: no finance-specific transformer representation.
[11]	Wu et al.; five Chinese A-share companies	CNN-derived sentiment index; S_I_LSTM and attention LSTM	Multiple textual and structured sources improved prediction. Limitation: one market and a small company sample.

The present study addresses these gaps through a unified daily framework. FinBERT supplies finance-specific contextual sentiment classification, while the average daily FinBERT sentiment score represents the news environment. This feature is aligned at the company-date level with closing price, EMA26, MACD, and RSI14 before sequential modeling. LSTM then learns temporal relationships from the resulting five-feature representation across Amazon, Apple, Google, Microsoft, and NVIDIA. This framework does not claim to retain every semantic detail of a news article; instead, it uses contextual language modeling to produce a more reliable sentiment feature and combines it with selected structured market indicators. The resulting design examines the research opportunity identified in prior work: integrating finance-aware sentiment, technical conditions, and temporal forecasting within one reproducible stock price forecasting framework.

3. Methodology

3.1. Data description

3.1.1. Data sources

The study uses two data sources for Amazon, Apple, Google, Microsoft, and NVIDIA: daily market records from Yahoo Finance and company-related news headlines from GDELT. For each company, it is collected 1,258 daily closing-price observations from Yahoo Finance for the period from 2019-01-02 to 2023-12-29. Each company's closing-price series are caculated to calculate three technical indicators: the 26-day

Exponential Moving Average (EMA26), Moving Average Convergence Divergence (MACD), and the 14-day relative strength index (RSI14).

The GDELT dataset comprises 58,466 company-related headlines published from 2019-01-01 to 2023-12-31, including 19,010 for Amazon, 16,441 for Google, 16,259 for Apple, 6,174 for Microsoft, and 582 for NVIDIA. Each headline are processed by using FinBERT to derive an article-level sentiment score, as detailed in Section 3.3, and average the scores for each company–date pair according to the calendar publication date. Then it align each daily average sentiment score with the market observation for the same company and trading date. When no headline is matched to a trading date, it is assigned a value of zero to the average daily FinBERT sentiment feature.

3.1.2. Statistical description

Table 2 summarizes the raw closing-price series before normalization. Each company contributes 1,258 observations, so the comparisons are based on equal-length time series. Mean closing prices range from 17.7464 for NVIDIA to 236.2598 for Microsoft, confirming that the five stocks operate on markedly different price scales. Microsoft also exhibits the largest absolute standard deviation (72.3885). However, absolute dispersion is partly influenced by the underlying price level. When variability is expressed relative to the mean, NVIDIA has the highest coefficient of variation at approximately 0.70, whereas the other four companies range from approximately 0.24 to 0.38. The contrast between absolute and relative dispersion supports training-only min-max normalization of the closing-price target, which maps values to a common numerical range while preserving direct next-trading-day closing-price prediction.

Table 3 describes the daily FinBERT aggregates after company-date alignment. The average daily FinBERT sentiment score has a mean of −0.1110, indicating a modest negative tendency across the aligned observations. Neutral headlines form the largest component of the mean daily article count: the mean neutral count is 4.7967, compared with 2.2034 negative and 0.6263 positive headlines. Of the mean 7.6264 headlines per company–day, approximately 62.9% are neutral, 28.9% are negative, and 8.2% are positive. The mean number of negative headlines exceeds that of positive headlines, which is consistent with the negative mean sentiment score. Daily article counts range from 0 to 92, and their standard deviation (8.2490) is slightly higher than their mean, showing that the number of aligned articles changes across company-days. The count variables are retained for descriptive interpretation, whereas the average daily FinBERT sentiment score is the news-derived feature supplied to the forecasting model.

Table 2. Descriptive statistics of daily closing prices

Company	Count	Minimum	Maximum	Mean	Std. dev.
Amazon	1,258	75.0140	186.5705	127.6497	31.2095
Apple	1,258	35.5475	198.1100	123.0311	46.5274
Google	1,258	51.2735	149.8385	98.1930	29.6923
Microsoft	1,258	97.4000	382.7000	236.2598	72.3885
NVIDIA	1,258	3.1997	50.4090	17.7464	12.4676

Table 3. Descriptive statistics of daily sentiment variables

Variable	Count	Minimum	Maximum	Mean	Std. dev.
Average daily FinBERT sentiment score	6,125	−0.9737	0.9535	−0.1110	0.2341
Article count	6,125	0	92	7.6264	8.2490

Table 3. Continued

Positive count	6,125	0	17	0.6263	1.1059
Negative count	6,125	0	60	2.2034	3.6139
Neutral count	6,125	0	58	4.7967	5.2193

3.2. Long Short-Term Memory

Long Short-Term Memory is a recurrent neural network architecture designed to model temporal dependencies while reducing the vanishing-gradient problem associated with conventional recurrent networks [12]. The LSTM memory mechanism contains a cell state and three control gates that determine which information is retained, added, or exposed at each position in an input sequence. For input x_t, previous hidden state h_{t-1}, and previous cell state c_{t-1}, the operations are defined in Equations (1)-(6):

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{c}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

where x_t is the normalized five-feature input vector at time step t; h_{t-1} and c_{t-1} are the preceding hidden state and cell state, respectively; f_t, i_t, and o_t denote the forget, input, and output gates; $\tilde{c}_t$ is the candidate cell state; c_t is the updated cell state; and h_t is the updated hidden state. The matrices W_f, W_i, W_c, and W_o, together with the corresponding bias vectors b_f, b_i, b_c, and b_o, are trainable parameters. The notation $[h_{t-1}, x_t]$ denotes vector concatenation, σ is the sigmoid activation function, $tanh$ is the hyperbolic tangent function, and $\odot$ denotes element-wise multiplication. The final hidden state of the 15-day sequence is mapped to the predicted normalized closing price through the linear output layer defined in Equation (7):

$$\hat{y}_{t+1} = W_y h_t + b_y \tag{7}$$

where $\hat{y}_{t+1}$ is the predicted min–max-normalized closing price for the next trading day, and W_y and by are the trainable weight matrix and bias of the output layer. The model therefore predicts the next-trading-day closing price directly rather than predicting a return.

During training, the model minimizes the Mean Squared Error (MSE) in the normalized closing-price space as defined in Equation (8):

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

where N is the number of training samples, y_i is the actual min–max-normalized closing price for sample i, and $\hat{y}_i$ is the corresponding predicted value.

3.3. FinBERT sentiment analysis

FinBERT is a transformer language model adapted to financial text [13]. For headline j, FinBERT produces logits $z_{j,k}$ for the positive, negative, and neutral classes. Softmax converts the logits to probabilities as shown in Equation (9):

$$p_{j,k} = \frac{exp(z_{j,k})}{\sum_m exp(z_{j,m})} \tag{9}$$

The predicted label is the class with the largest probability and q_j is that maximum probability. A signed, confidence-weighted article score is then defined in Equation (10):

$$s_j = \{ \begin{matrix} q_j & ifpositive \\ -q_j & ifnegative \\ 0 & ifneutral \end{matrix} \tag{10}$$

For company c on date d, the average daily FinBERT sentiment score is the arithmetic mean of the article-level scores observed on that company-date using Equation (11):

$$\overline{S}_{c,d} = \frac{1}{N_{c,d}} \sum_{j=1}^{N_{c,d}} s_j \tag{11}$$

It is retained the daily class-specific headline counts—positive, negative, and neutral—solely for descriptive characterization and exclude them from the LSTM input vector. For company–trading-day observations without matched headlines, it is imputed the daily FinBERT sentiment feature with zero. Consequently, the LSTM receives a single scalar sentiment measure at the company–date level rather than token-level contextual representations. FinBERT functions exclusively as a pretrained sentiment feature extractor, and its parameters are not jointly optimized with those of the downstream LSTM forecasting model.

3.4. Financial technical indicators

Closing price is the observed market variable used by the forecasting model. EMA12 and EMA26 summarize price trends over different horizons. EMA26 is retained as an input feature, whereas EMA12 is calculated only as an intermediate quantity for MACD. For period n, the exponential moving average is calculated recursively with smoothing factor as shown in Equation (12):

$$EMA_{n,t} = \alpha_n P_t + (1 - \alpha_n) EMA_{n,t-1} \tag{12}$$

MACD is the difference between the short- and long-period exponential moving averages as defined in Equation (13):

$$MACD_t = EMA_{12,t} - EMA_{26,t} \tag{13}$$

RSI14 represents the balance between recent upward and downward price movements. If $\Delta \mathrm{P}_t = P_t - P_{t-1}$, the gain and loss components are $G_t = \max(\Delta \mathrm{P}_t, 0)$ and $L_t = \max(-\Delta \mathrm{P}_t, 0)$. With 14-period average gain and loss, relative strength and RSI14 are calculated using Equations (14) and (15):

$$RS_t = \frac{AverageGain_{14,t}}{AverageLoss_{14,t}} \tag{14}$$

$$RSI14_t = 100 - \frac{100}{1 + RS_t} \tag{15}$$

3.5. Input features

The feature vector for company c on trading day t is defined in Equation (16):

$$x_{c,t} = [P, EMA26, MACD, RSI14, \overline{S}]^T \in R^5 \tag{16}$$

For each company, feature-wise min–max normalization using the minimum and maximum values estimated exclusively from the training partition is performed using Equation (17):

$$x'_{c,t} = \frac{x_{c,t} - x_{min,train}}{x_{max,train} - x_{min,train}} \tag{17}$$

For a lookback of 15 trading days, the input sequence is defined in Equation (18):

$$X_{c,t} = [x'_{c,t-14}, \ldots, x'_{c,t}] \in R^{15\times 5} \tag{18}$$

The 15-day lookback window corresponds to approximately three trading weeks. It was selected as a parsimonious design choice that provides sufficient recent observations for the LSTM to represent short-term price and indicator dynamics while limiting the inclusion of older information that may be less relevant to next-trading-day prediction. The prediction target is the min–max-normalized closing price on the subsequent trading day. Each LSTM input sequence comprises five features: closing price, EMA26, MACD, RSI14, and the average daily FinBERT sentiment score. FinBERT and the LSTM are implemented as sequential but independently parameterized components. FinBERT produces the scalar sentiment feature incorporated into the LSTM input vector, whereas the LSTM neither processes FinBERT embeddings nor updates FinBERT parameters during training.

3.6. Experiment setup

The forecasting model uses historical closing prices, EMA26, MACD, RSI14, and the average daily FinBERT sentiment score as the five LSTM input features. Firstly the FinBERT are applied to company-related news headlines to derive article-level sentiment classifications and confidence-weighted sentiment scores. These scores are aggregated by company and date to obtain the average daily FinBERT sentiment score, which is subsequently aligned with closing price, EMA26, MACD, and RSI14. It is normalized the five features using min–max parameters estimated from the training set and construct the LSTM inputs as 15-day sequences. The LSTM directly estimates the min–max-normalized next-trading-day closing price, after which the prediction can be inverse-normalized to the original closing-price scale.

The experiments were conducted in Google Colaboratory (Colab). The software environment used Python 3.10 and PyTorch 2.6.0 for model construction, training, validation, and evaluation.

Hyperparameter selection was performed over predefined candidate sets. The number of hidden units was tuned over {10, 20, 30}, the number of LSTM layers was tuned over {1, 2}, and the initial learning rate was selected from {0.001, 0.005, 0.01}. For all configurations, the batch size was fixed at 32. The models were trained using the Adam optimizer with a weight decay of 5×10^{-4} and a dropout rate of 0.2. The maximum number of training epochs was set to 1,500. Early stopping was applied when the validation loss did not decrease for 200 consecutive epochs.

4. Results

4.1. Evaluation metrics

Root Mean Squared Error (RMSE) measures the square root of the average squared difference between the actual and predicted normalized closing prices. Because the errors are squared before averaging, RMSE assigns greater weight to comparatively large forecasting deviations. It is defined in Equation (19):

$$RMSE = \sqrt{\tfrac{1}{N} \textstyle\sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{19}$$

Mean Absolute Error (MAE) measures the average absolute difference between the actual and predicted normalized closing prices without considering whether an individual prediction is above or below the actual value. It is defined in Equation (20):

$$MAE = \tfrac{1}{N} \textstyle\sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{20}$$

where N is the number of observations in the test set, y_i is the actual normalized closing price for observation i, and $\hat{y}_i$ is the corresponding predicted value. Both RMSE and MAE are non-negative, and a smaller value indicates closer agreement between actual and predicted closing prices. RMSE is more sensitive to large individual errors, whereas MAE represents the average absolute error, with each deviation contributing in direct proportion to its magnitude.

4.2. Experimental results

Table 4 reports the latest test-set results for the five companies. The minimum RMSE is 0.0147, recorded for Apple, whereas the maximum RMSE is 0.0245, recorded for NVIDIA. The minimum MAE is 0.0110 for Apple, and the maximum MAE is 0.0201 for NVIDIA. The average RMSE and MAE across the five companies are 0.0209 and 0.0160, respectively. Apple therefore achieves the highest forecasting accuracy during the July–December 2023 test period under both evaluation measures, while NVIDIA presents the greatest forecasting difficulty. Because RMSE and MAE identify the same best- and worst-performing companies, the overall ranking is consistent across the two metrics rather than dependent on a single error measure. All values are calculated from min–max-normalized closing prices and therefore represent errors on the normalized scale rather than in currency units.

Table 4. Company-level forecasting errors for min-max-normalized test-set closing prices

Company	RMSE	MAE
Amazon	0.0243	0.0181
Apple	0.0147	0.0110
Google	0.0239	0.0171
Microsoft	0.0170	0.0136
NVIDIA	0.0245	0.0201
Average	0.0209	0.0160

Note: RMSE and MAE are calculated from the min-max-normalized actual and predicted closing prices shown in Figures 1–5. Lower values indicate better test-set forecasting accuracy.

Apple achieves the best performance under both measures, with RMSE = 0.0147 and MAE = 0.0110. Microsoft ranks second (RMSE = 0.0170; MAE = 0.0136), followed by Google (0.0239; 0.0171), Amazon (0.0243; 0.0181), and NVIDIA (0.0245; 0.0201). The identical ranking under RMSE and MAE indicates a consistent ordering of company-level forecasting difficulty in this test period.

NVIDIA records the largest errors, although its predicted series still follows the main direction of the actual normalized price. The curve shows wider deviations around rapid rises, peaks, and reversals, which is consistent with the greater difficulty of tracking abrupt short-term movements. The NVIDIA dataset contains 582 news headlines, substantially fewer than the 6,174–19,010 headlines available for each of the other companies. This lower volume of coverage may produce more trading days with no matched news and therefore more zero-valued daily sentiment inputs; it may also make the aggregated sentiment feature more sensitive to individual headlines. These effects could reduce the stability and informational richness of the sentiment signal. However, company-specific missing-news frequencies and feature ablations are not evaluated, so the higher NVIDIA error cannot be attributed causally to news coverage alone.

Across the five companies, the predicted series generally follow the direction and level of the actual min–max-normalized closing prices. Figures 1–5 compare the actual and predicted series over the test period from

July to December 2023. The forecasts are generally smoother than the actual closing prices, and the largest local deviations occur around abrupt turning points, steep declines, and rapid recoveries.

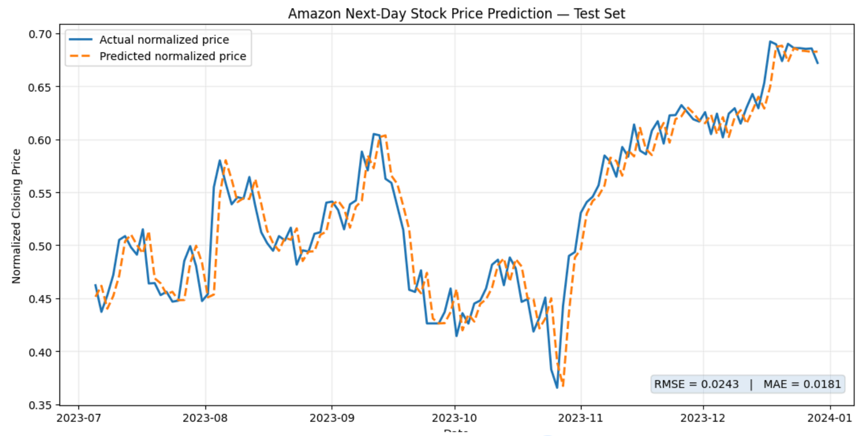

Figure 1. Amazon actual and predicted min-max-normalized next-trading-day closing prices on the test set

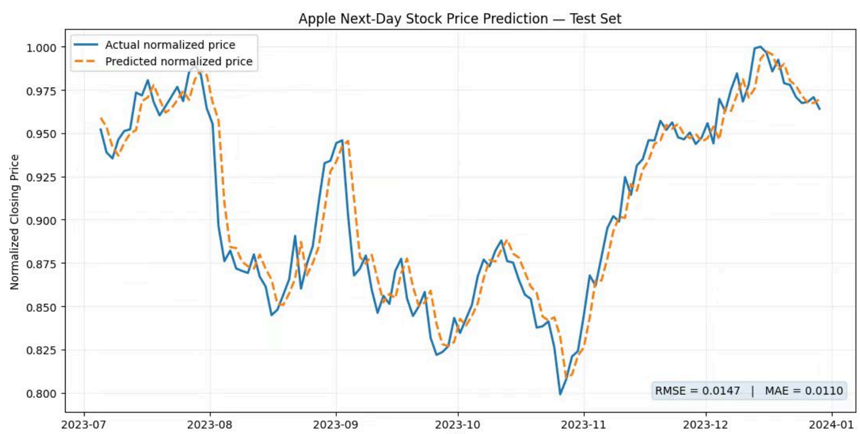

Figure 2. Apple actual and predicted min-max-normalized next-trading-day closing prices on the test set

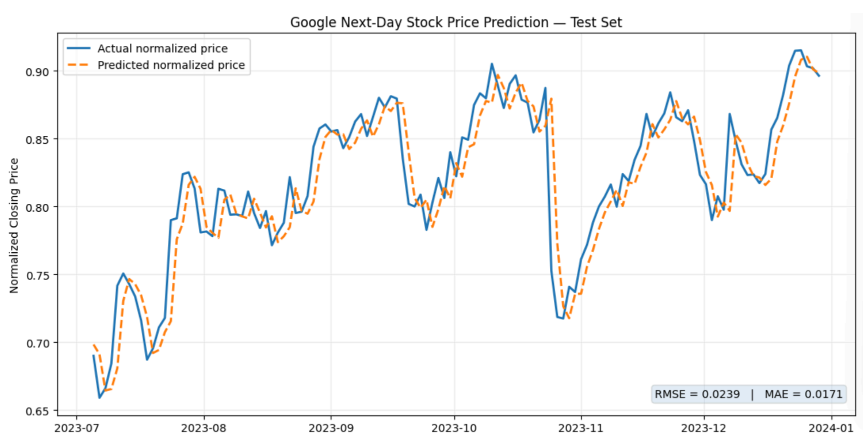

Figure 3. Google actual and predicted min-max-normalized next-trading-day closing prices on the test set

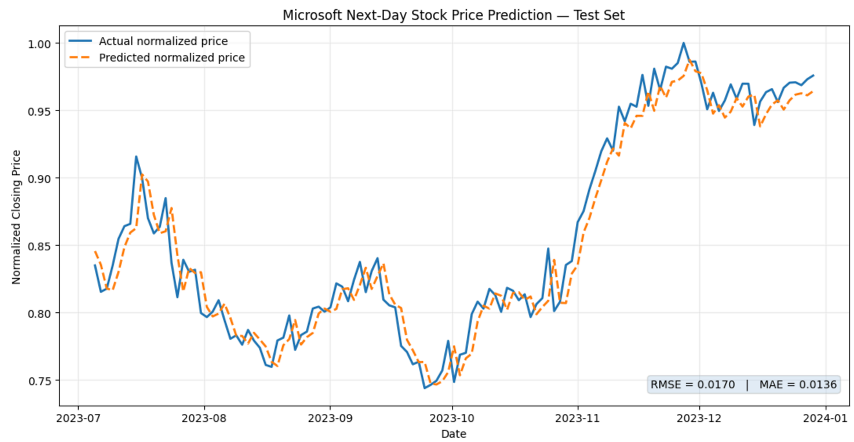

Figure 4. Microsoft actual and predicted min-max-normalized next-trading-day closing prices on the test set

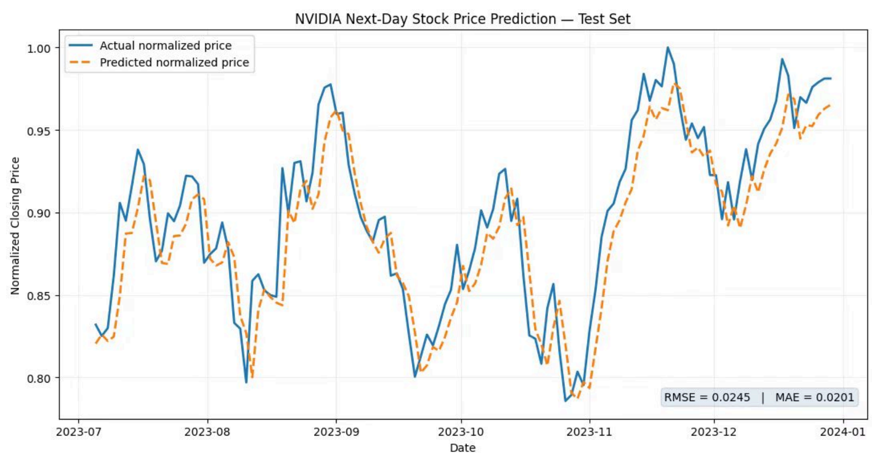

Figure 5. NVIDIA actual and predicted min-max-normalized next-trading-day closing prices on the test set

Apple produces the closest fit, consistent with its lowest RMSE and MAE. Its predicted curve follows both the broad decline from July into October and the subsequent recovery, with comparatively small discrepancies around the sharp early-August and late-October changes. Microsoft also shows close tracking, but the model modestly underestimates parts of the strong November–December rise and several local peaks.

Amazon and Google exhibit similar normalized-scale errors. In both cases, the model follows the medium-term trend and the recovery into year-end, while smoothing sharp movements; the most visible discrepancies occur around the late-October troughs and their immediate rebounds. NVIDIA retains the overall cyclical pattern, but its predictions understate several peaks and the late-year rise, producing the highest RMSE and MAE. Collectively, the curves support the conclusion that the model captures persistent price dynamics more accurately than isolated short-term extremes.

5. Discussion

The latest results show that the integrated LSTM framework tracks min–max-normalized next-trading-day closing prices across all five companies using the same five-feature representation. The average RMSE and MAE across the five companies are 0.0209 and 0.0160, respectively. Apple and Microsoft show the closest numerical fit, whereas Amazon, Google, and NVIDIA exhibit larger deviations, particularly near sudden changes in direction. The predicted curves are smoother than the actual series, suggesting that the LSTM captures the dominant temporal pattern but responds less sharply to short-lived extremes.

These comparisons support the theoretical motivation for the present feature design, but they do not replace an experiment-specific baseline. The supplied output contains results only for the integrated model and therefore does not quantify the incremental contribution of FinBERT sentiment or technical indicators relative to price-only LSTM. Accordingly, the current evidence establishes the feasibility and company-level performance of the integrated framework; a price-only baseline and feature-group ablations are necessary before attributing the observed accuracy specifically to either news or technical information.

The latest metrics are internally consistent: RMSE is greater than MAE for every company, as expected when both measures are calculated from the same prediction errors. This resolves the ordering inconsistency in the earlier draft. The numerical values should nevertheless be interpreted as normalized-scale errors; they do not represent dollar deviations unless predictions and observations are inverse-normalized with the corresponding training-set minimum and maximum.

6. Conclusion and future work

6.1. Conclusion

This study develops an LSTM-based framework for stock price forecasting that integrates historical market information, selected technical indicators, and FinBERT-derived news sentiment. GDELT headlines are classified into positive, negative, and neutral categories, converted into confidence-weighted scores, and aggregated by company and date to obtain the average daily FinBERT sentiment score. This feature is aligned with closing price, EMA26, MACD, and RSI14. Fifteen-day sequences of these five min-max-normalized features are then used to predict the min-max-normalized next-trading-day closing price directly; the model output can subsequently be inverse-normalized to the original closing-price scale.

The framework was applied consistently to Amazon, Apple, Google, Microsoft, and NVIDIA in a Google Colab environment using Python 3.10 and PyTorch 2.6.0. For min-max-normalized test-set closing prices, RMSE ranges from 0.0147 to 0.0245 and MAE ranges from 0.0110 to 0.0201, with averages of 0.0209 and 0.0160, respectively. Apple records the lowest error under both measures, followed by Microsoft, while NVIDIA records the highest. The actual-versus-predicted curves confirm that the model captures the principal company-level price movements, although its outputs are smoother and less responsive at abrupt turning points.

The principal contribution is the reproducible integration of finance-aware textual information with structured temporal features. The design preserves a clear boundary between FinBERT sentiment extraction and LSTM price forecasting, avoiding the inaccurate implication that the recurrent model directly processes transformer embeddings. The findings accord with literature showing that financial text can complement price and technical information [4, 6, 11, 16, 20, 21]. For practitioners, the framework offers a transparent pipeline for enriching sequential forecasts with average daily FinBERT sentiment scores. For researchers, it provides a multi-company setting in which alignment, company-specific news coverage, feature contribution, and temporal robustness can be examined systematically.

6.2. Future work

Future work should first conduct controlled ablations comparing price-only LSTM, price plus the selected technical indicators, price plus sentiment, and the complete five-feature model under identical splits and hyperparameters. Rolling-window or walk-forward evaluation, repeated training with multiple random seeds, and statistical tests would provide stronger evidence of robustness. News processing can be improved by mapping after-market and weekend publications to the next tradable session, weighting sources and event types, retaining separate positive, negative, and neutral intensities, and testing attention-based aggregation rather than a daily mean. Joint or cross-attention architectures could also connect textual representations more directly with market sequences, although their additional complexity should be evaluated against simpler FinBERT score integration.

The empirical scope should be extended to additional sectors, smaller firms, international markets, and stress periods. Comparisons with GRU, temporal convolutional networks, Transformers, and hybrid models would clarify whether the observed performance depends on LSTM specifically. Interpretability methods could identify which days and features drive individual forecasts, while probabilistic forecasting could quantify uncertainty rather than provide only point predictions. Most importantly, the final evaluation pipeline should record the selected hyperparameter configuration, train-validation-test boundaries, random seeds, min-max normalization and inverse-normalization procedures, and metric code so that all reported results are fully reproducible.

References

[1] Gu, W., Zhong, Y., Li, S., Wei, C., Dong, L., Wang, Z., & Yan, C. (2024). Predicting stock prices with FinBERT-LSTM: Integrating news sentiment analysis. In *Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing* (pp. 67–72). https://doi.org/10.1145/3694860.3694870

[2] Maqbool, J., Aggarwal, P., Kaur, R., Mittal, A., & Ganaie, I. A. (2023). Stock prediction by integrating sentiment scores of financial news and MLP-regressor: A machine learning approach. *Procedia Computer Science, 218*, 1067–1078. https://doi.org/10.1016/j.procs.2023.01.086

[3] Deveikyte, J., Geman, H., Piccari, C., & Provetti, A. (2022). A sentiment analysis approach to the prediction of market volatility. *Frontiers in Artificial Intelligence, 5*, Article 836809. https://doi.org/10.3389/frai.2022.836809

[4] Li, X., Wu, P., & Wang, W. (2020). Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong. *Information Processing & Management, 57*(3), Article 102212. https://doi.org/10.1016/j.ipm.2020.102212

[5] Fazlija, B., & Harder, P. (2022). Using financial news sentiment for stock price direction prediction. *Mathematics, 10*(13), Article 2156. https://doi.org/10.3390/math10132156

[6] Jin, Z., Yang, Y., & Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing and Applications, 32*, 9713–9729. https://doi.org/10.1007/s00521-019-04504-2

[7] Ko, C.-R., & Chang, H.-T. (2021). LSTM-based sentiment analysis for stock price forecast. *PeerJ Computer Science, 7*, Article e408. https://doi.org/10.7717/peerj-cs.408

[8] Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J., Tang, C., & Li, J. (2022). A survey on machine learning models for financial time series forecasting. *Neurocomputing, 512*, 363–380. https://doi.org/10.1016/j.neucom.2022.09.003

[9] Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., & Anastasiu, D. C. (2019). Stock price prediction using news sentiment analysis. In *Proceedings of the 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications* (pp. 205–208). https://doi.org/10.1109/BigDataService.2019.00035

[10] Shahi, T. B., Shrestha, A., Neupane, A., & Guo, W. (2020). Stock price forecasting with deep learning: A comparative study. *Mathematics, 8*(9), Article 1441. https://doi.org/10.3390/math8091441

[11] Wu, S., Liu, Y., Zou, Z., & Weng, T.-H. (2022). S_I_LSTM: Stock price prediction based on multiple data sources and sentiment analysis. *Connection Science, 34*(1), 44–62. https://doi.org/10.1080/09540091.2021.1940101

[12] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[13] Araci, D. (2019). *FinBERT: Financial sentiment analysis with pre-trained language models.* arXiv preprint arXiv:1908.10063.

[14] Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance, 66*(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

[15] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

[16] Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). Deep learning for stock prediction using numerical and textual information. In *Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science* (pp. 1–6). https://doi.org/10.1109/ICIS.2016.7550882

[17] Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015). Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence* (pp. 2327–2333).

[18] Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T.-Y. (2018). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining* (pp. 261–269). https://doi.org/10.1145/3159652.3159690

[19] Xu, Y., & Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 1970–1979). https://doi.org/10.18653/v1/P18-1183

[20] Picasso, A., Merello, S., Ma, Y., Oneto, L., & Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications, 135*, 60–70. https://doi.org/10.1016/j.eswa.2019.06.014

[21] Kim, J., Kim, H.-S., & Choi, S.-Y. (2023). Forecasting the S&P 500 index using mathematical-based sentiment analysis and deep learning models: A FinBERT transformer model and LSTM. *Axioms, 12*(9), Article 835. https://doi.org/10.3390/axioms12090835