News-Enhanced Stock Price Forecasting: A Multi-Source Textual Analysis Approach with Hierarchical Model Training

Harsh Milind Tirhekar* Atharva Vishwas Kulkarni†

January 2026

Abstract

Hypothesis: Does incorporating news sentiment and textual features improve stock price prediction beyond traditional time series methods?

Method: We combine 26 years of Apple Inc. (AAPL) price data (1999–2025) with 57+ million financial news articles from multiple sources, extracting sentiment via VADER, TextBlob, and FinBERT, along with topic features from LDA modeling. We employ a hierarchical training strategy where foundational models (SARIMAX, Linear Regression) trained on the full historical dataset provide predictions as input features for neural networks (LSTM, GRU, Transformer) trained on recent data.

Results: We find incremental but meaningful predictive power from news incorporation. The 7-day rolling mean of VADER sentiment provides a statistically significant 1.5% RMSE improvement for SARIMAX models. Our hierarchical approach substantially improves neural network performance, with Transformer R^2 increasing from -1.7 to 0.87 and GRU improving by 25%. However, simple linear regression with sentiment features achieves the best overall performance (RMSE = \$1.83, $R^2 = 0.999$), suggesting that for limited training samples, model complexity should match data availability.

Keywords: Stock forecasting, News sentiment, Financial text mining, Machine learning, Time series analysis, Hierarchical training

JEL Classification: G14, G17, C45, C53

*Department of Statistics & Data Science, Carnegie Mellon University. Email: htirhekar@cmu.edu
†Department of Statistics & Data Science, Carnegie Mellon University. Email: akulkarni@cmu.edu

1 Introduction and Motivation

The efficient market hypothesis (EMH) posits that asset prices fully reflect all available information, making consistent outperformance through prediction theoretically impossible [Fama, 1970]. However, a substantial body of empirical research documents predictable patterns in stock returns, particularly around corporate events such as earnings announcements [Ball and Brown, 1968, Bernard and Thomas, 1989], merger announcements [Andrade et al., 2001], and product launches [Chaney et al., 1991].

This paper investigates whether incorporating news sentiment and textual features can improve stock price forecasting beyond traditional time series methods. We focus on Apple Inc. (AAPL), one of the most extensively covered and liquid stocks in U.S. markets, providing an ideal laboratory for examining the marginal contribution of news-based features.

1.1 Research Questions

We address three primary research questions:

Research Question 1 (RQ1): Does news sentiment provide incremental predictive power for next-day stock prices beyond technical indicators?

Research Question 2 (RQ2): What is the optimal temporal aggregation (rolling window) for sentiment features to balance noise reduction against information lag?

Research Question 3 (RQ3): Can a hierarchical training strategy—where traditional models inform neural network inputs—overcome the limitations of training deep learning models on limited financial time series?

1.2 Main Contributions

Our study makes the following contributions to the financial data science literature:

1. **Multi-source data integration:** We construct a comprehensive news corpus spanning 1999–2025 by combining multiple open-source datasets, addressing the common limitation of short sample periods in prior work.

2. **Hierarchical training strategy:** We propose using predictions from models trained on long-term data as features for models trained on recent data, enabling neural networks to leverage historical patterns without suffering from distribution shift.

3. **Systematic sentiment comparison:** We rigorously compare three sentiment extraction methods (VADER, TextBlob, FinBERT) across multiple rolling windows, providing practical guidance for practitioners.

4. **Transformer rehabilitation:** We demonstrate that the poor performance of Transformers in financial forecasting literature can be substantially improved through appropriate input engineering, increasing R^2 from -1.7 to 0.87.

1.3 Preview of Findings

Our key findings can be summarized as:

Finding 1. *News sentiment provides statistically significant but economically modest improvements to forecasting accuracy. The 7-day rolling mean of VADER sentiment reduces SARIMAX RMSE by 1.5% ($p < 0.05$).*

Finding 2. *Model complexity should be calibrated to sample size. With approximately 1,000 training observations, linear regression with 56 features outperforms neural networks with 50,000+ parameters.*

Finding 3. *The hierarchical training strategy substantially improves neural network performance by providing long-term trend information as an input feature, enabling Transformers to achieve competitive results.*

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes our data sources and preprocessing. Section 4 presents our modeling approach. Section 5 reports empirical results. Section 6 discusses limitations. Section 7 concludes.

2 Related Work

Our research connects to three strands of the finance and machine learning literature: the informational content of news, machine learning for financial prediction, and transformer architectures for time series.

2.1 Informational Content of Financial News

A long literature documents that corporate announcements affect stock prices. Ball and Brown [1968] established that earnings surprises drive abnormal returns, while Bernard and Thomas [1989] showed that this drift persists for several months—the "post-earnings announcement drift" (PEAD) that challenges market efficiency.

Beyond formal earnings, researchers have examined various news types:

- **Press releases:** Tetlock [2007] finds that negative words in Wall Street Journal columns predict lower next-day returns and higher trading volume.

- **Product announcements:** Chaney et al. [1991] documents significant stock price reactions to new product announcements, with the magnitude depending on the firm's innovation history.
- **Social media:** Bollen et al. [2011] shows that Twitter sentiment predicts stock market movements with 87.6% accuracy in direction.

Our work differs from this prior literature by: (1) using a broader definition of "news" encompassing all financial articles rather than specific announcement types, (2) focusing on price level prediction rather than return prediction or directional accuracy, and (3) employing modern NLP methods including transformer-based sentiment extraction.

2.2 Machine Learning for Stock Prediction

The application of machine learning to financial prediction has grown substantially. Key contributions include:

Fischer and Krauss [2018] applied LSTM networks to S&P 500 constituents, finding that deep learning outperforms random forests and logistic regression for return prediction. Ding et al. [2015] combined convolutional neural networks with event embeddings, achieving improved directional accuracy.

More recently, Xu and Cohen [2018] proposed StockNet, a variational autoencoder that jointly models price and tweet embeddings, while Feng et al. [2019] introduced attention mechanisms for multi-scale temporal patterns.

However, a critical challenge remains: financial time series are short relative to the parameter counts of deep learning models. Zhang et al. [2023] find that simpler models often outperform deep networks on financial datasets with fewer than 5,000 observations. Our hierarchical training approach directly addresses this limitation.

2.3 Transformers for Time Series Forecasting

The transformer architecture [Vaswani et al., 2017] has revolutionized NLP and is increasingly applied to time series. Specialized architectures include:

- **Informer:** Zhou et al. [2021] introduces ProbSparse attention for long-sequence forecasting.
- **Autoformer:** Wu et al. [2021] decomposes series into trend and seasonal components within the transformer framework.
- **Temporal Fusion Transformer:** Lim et al. [2021] incorporates variable selection for interpretable multi-horizon forecasting.

Despite these advances, standard transformers often fail on financial data. We demonstrate that this failure stems from architectural mismatch (single-step prediction degenerates self-attention) rather than fundamental limitations, and that appropriate input engineering restores competitive performance.

3 Data Description

This section describes our data sources, preprocessing steps, and the construction of the analysis dataset.

3.1 Data Sources

We compile data from multiple sources to construct a comprehensive dataset spanning January 1999 to January 2025:

Table 1: Data Sources and Coverage

Data Type	**Source**	**Coverage**	**Access**
Stock prices	Yahoo Finance	1999–2025	Public API
News (2018–2023)	HuggingFace Dataset	5 years	API key
News (1999–2017)	Archived CSV	18 years	Open source
Related stocks	Yahoo Finance	1999–2025	Public API
Market indices	Yahoo Finance	1999–2025	Public API

3.1.1 Stock Price Data

We obtain daily OHLCV (Open, High, Low, Close, Volume) data for Apple Inc. (AAPL) via the `yfinance` Python package. The data spans 6,542 trading days from January 4, 1999 to January 15, 2025. Prices are adjusted for stock splits (7:1 in 2014, 4:1 in 2020), representing a 1,040× increase from \$0.25 to \$260 over the sample period.

3.1.2 News Data

We construct the news corpus by combining two sources:

1. **HuggingFace Financial News Dataset (2018–2023):** Contains approximately 50 million financial news articles with headlines, snippets, and publication dates. We filter for AAPL-relevant articles using keyword matching.

2. **Historical News Archive (1999–2017):** We augment the HuggingFace data with an archived CSV dataset containing pre-2018 financial news. This extension is critical for training models on longer historical windows.

After filtering and deduplication, our final corpus contains 57.3 million unique articles spanning 26 years.

3.1.3 Related Stocks and Market Context

To provide market context, we collect price data for:

- Related technology stocks: MSFT, GOOGL, AMZN
- Market indices: S&P 500 (^GSPC), Dow Jones (^DJI), NASDAQ (^IXIC)

All features from these securities use one-day lagged values to prevent lookahead bias.

3.2 Data Preprocessing

3.2.1 News-Price Alignment

A critical challenge is aligning news publication times with trading days. We implement the following rules:

- News published before market close on day t is assigned to day t.
- News published after market close or on weekends is assigned to the next trading day.
- Multiple articles on the same day are aggregated by averaging sentiment scores.
- Days without news are assigned NaN, later filled with 7-day backward rolling mean.

3.2.2 Sentiment Extraction

We extract sentiment using three methods:

1. **VADER (Valence Aware Dictionary and sEntiment Reasoner):** A rule-based model incorporating intensity modifiers and emoticons. Outputs compound score in $[-1, 1]$.
2. **TextBlob:** A pattern-based approach providing polarity in $[-1, 1]$ and subjectivity in $[0, 1]$.
3. **FinBERT:** A BERT model fine-tuned on financial text. Outputs probability distribution over {negative, neutral, positive}, from which we compute $S = P(\text{positive}) - P(\text{negative})$.

3.2.3 Topic Modeling

We apply Latent Dirichlet Allocation (LDA) with 5 topics to the news corpus after standard preprocessing (tokenization, stopword removal, lemmatization). This yields 5 topic proportion features per day, capturing thematic variation in news coverage.

3.3 Dataset Characteristics

Table 2 presents summary statistics for key variables:

Table 2: Summary Statistics

Variable	**Mean**	**Std**	**Min**	**Max**	**N obs**
Close price ($)	52.87	62.41	0.25	259.02	6,542
Daily return (%)	0.12	2.84	-51.9	13.9	6,541
VADER compound	0.18	0.32	-0.98	0.99	6,542
TextBlob polarity	0.12	0.19	-0.87	0.93	6,542
FinBERT score	0.24	0.41	-0.95	0.98	6,542
News coverage (articles/day)	8.76	12.3	0	147	6,542

Notable characteristics include:

- **Non-stationarity:** The 1,040× price increase violates stationarity assumptions of many time series models.
- **Sentiment asymmetry:** All three sentiment measures show positive mean, reflecting generally favorable AAPL coverage.
- **Sparse news coverage:** Only 31% of trading days have at least one AAPL-specific article in our corpus.

3.4 Data Quality and Limitations

We acknowledge several data limitations:

1. **Source heterogeneity:** Combining multiple news sources introduces potential inconsistencies in coverage and quality across time periods.
2. **Survivorship bias:** We study AAPL, which has survived and thrived—results may not generalize to the average stock.
3. **Missing news data:** The 69% of days without dedicated AAPL articles requires imputation, potentially attenuating sentiment signal.

4. **Google News limitation:** Real-time news feeds (e.g., Google News API) only provide recent data, preventing extension to historical periods.

4 Methodology

This section describes our modeling framework, including feature engineering, the hierarchical training strategy, and the models employed.

4.1 Feature Engineering

We construct 55 features organized into four categories:

Table 3: Feature Categories

Category	**Description**	**Count**
Sentiment	VADER, TextBlob, FinBERT (raw + rolling)	15
Text/Topic	LDA topics, adjective counts, keywords	15
Market context	Related stocks, indices (lagged)	18
Technical	Price/volume rolling means	7
Total		**55**

For sentiment features, we compute rolling means over windows $w \in \{3, 7, 14, 30\}$ days:

$$S_t^{(w)} = \frac{1}{w} \sum_{i=0}^{w-1} s_{t-i} \tag{1}$$

where s_t is the raw daily sentiment score. This smoothing reduces noise at the cost of introducing lag.

4.2 Hierarchical Training Strategy

A key methodological contribution is our hierarchical training approach, designed to address the challenge of training neural networks on short financial time series.

Hypothesis 1. *Models trained on long historical data can provide useful features for models trained on recent data, even when the long-term data distribution differs substantially from the recent period.*

We implement this through a two-stage process:

Stage 1: Foundational Models

- Train Linear Regression, SARIMAX, and TCN on full 26-year dataset (1999–2025)

- These models learn long-term patterns and trend dynamics
- Generate out-of-sample predictions for recent period (2020–2025)

Stage 2: Neural Networks with Foundational Features

- Add foundational model predictions as the 56th feature
- Train LSTM, GRU, BiLSTM, CNN-LSTM, Transformer on recent 5-year data
- Neural networks learn to correct foundational model errors using recent patterns

This approach addresses three key challenges:

1. **Distribution shift:** Neural networks avoid learning outdated patterns from distant history.
2. **Sample efficiency:** Foundational predictions encode long-term information compactly.
3. **Trend awareness:** The 56th feature provides explicit trend signal that neural networks otherwise struggle to capture.

4.3 Models

We evaluate the following model classes:

4.3.1 Traditional Time Series: SARIMAX

The Seasonal ARIMA with eXogenous regressors:

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D y_t = \theta(B)\Theta(B^s)\epsilon_t + \beta X_t \tag{2}$$

where X_t contains sentiment features. We select order $(p, d, q) = (2, 1, 1)$ via AIC minimization.

4.3.2 Linear Regression

Simple OLS on full feature set:

$$y_{t+1} = \beta_0 + \sum_{j=1}^{55} \beta_j x_{jt} + \epsilon_t \tag{3}$$

Despite its simplicity, this model provides a strong baseline given our feature engineering.

4.3.3 Recurrent Neural Networks

We implement LSTM, GRU, and BiLSTM with architecture:

- 2 recurrent layers with 64 hidden units
- Dropout rate 0.2
- Dense output layer
- Training: 100 epochs, batch size 32, Adam optimizer, early stopping (patience=15)

4.3.4 CNN-LSTM Hybrid

Convolutional layers extract local features before LSTM processing:

- 1D convolution with 32 filters, kernel size 3
- MaxPooling followed by LSTM layer
- Combined parameter count: 26K

4.3.5 Transformer

Standard transformer encoder with:

- $d_{model} = 64$, $n_{heads} = 4$, $n_{layers} = 2$
- Feed-forward dimension: 256
- Parameter count: 51K

A key finding is that standard transformers fail for single-step forecasting because self-attention degenerates when sequence length is 1. We address this by providing multi-step historical context through our foundational features.

4.4 Evaluation Framework

4.4.1 Walk-Forward Validation

We employ expanding-window walk-forward validation to simulate realistic forecasting:

1. Initial training window: first 500 observations
2. Predict next observation
3. Expand training window by 1 observation
4. Repeat until end of sample

This yields 85+ out-of-sample predictions for evaluation while maintaining temporal integrity.

4.4.2 Metrics

We report:

- **RMSE:** $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- **MAE:** $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- **MAPE:** $\frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$
- R^2: $1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$

We note that R^2 on price levels is inflated by the trending nature of the series. For interpretability, we also report results for return (stationary) prediction.

5 Empirical Results

This section presents our main findings organized around the three research questions.

5.1 RQ1: Does News Sentiment Improve Forecasting?

Table 4 compares SARIMAX performance across sentiment configurations:

Table 4: SARIMAX Performance by Sentiment Configuration

Sentiment Feature	**RMSE ($)**	**MAE ($)**	**MAPE (%)**	R^2
No sentiment (baseline)	2.71	1.96	1.24	0.9982
VADER raw	2.70	1.95	1.23	0.9982
VADER RM7	**2.66**	**1.91**	**1.21**	**0.9984**
VADER RM14	2.68	1.93	1.22	0.9983
VADER RM30	2.72	1.97	1.25	0.9981
TextBlob raw	2.73	1.97	1.24	0.9981
TextBlob RM7	2.70	1.94	1.22	0.9982
FinBERT raw	2.71	1.95	1.23	0.9982
FinBERT RM7	2.70	1.94	1.22	0.9982

Key finding: The 7-day rolling mean of VADER sentiment achieves the lowest RMSE ($2.66), representing a 1.5% improvement over the no-sentiment baseline. This difference is statistically significant at the 5% level based on Diebold-Mariano tests.

The 7-day window balances noise reduction (61% improvement in signal-to-noise ratio) against information lag (3-day effective lag). Longer windows (14, 30 days) introduce excessive lag that offsets noise reduction benefits.

FinBERT observation: Despite FinBERT's superior performance on sentiment classification benchmarks, it shows minimal improvement from rolling means. We attribute this to the inherent smoothing in BERT's multi-layer attention mechanism, which already aggregates contextual information.

5.2 RQ2: Model Comparison

Table 5 presents results across all models:

Table 5: Model Performance Comparison

Model	Training Data	RMSE ($)	MAE ($)	MAPE (%)	R^2
Traditional Models (26-year training)					
Linear Regression	Full	**1.83**	**1.34**	**0.94**	**0.9992**
SARIMAX (VADER RM7)	Full	2.66	1.91	1.21	0.9984
TCN	Full	21.16	18.34	9.87	0.8912
Neural Networks without hierarchical features					
LSTM	5-year	14.21	12.18	5.42	0.6812
GRU	5-year	11.83	10.01	4.31	0.7234
Transformer	5-year	97.01	77.41	44.89	-1.17
Neural Networks with hierarchical features (56th feature)					
LSTM	5-year	12.12	10.58	4.54	0.8909
GRU	5-year	7.63	6.44	2.78	0.9356
BiLSTM	5-year	7.77	6.33	2.81	0.9012
CNN-LSTM	5-year	**7.34**	**6.01**	**2.64**	**0.8939**
Transformer	5-year	8.42	7.21	3.12	0.8734

Key findings:

1. **Linear regression dominates:** Despite its simplicity, linear regression achieves the best overall performance (RMSE = \$1.83). This reflects the limited sample size—with only ∼1,000 training observations, the 55-feature linear model achieves better generalization than neural networks with 50K+ parameters.

2. **Hierarchical training transforms neural network performance:** Adding the foundational model prediction as the 56th feature dramatically improves all neural networks. Most notably:

   - GRU R^2: 0.72 → 0.94 (+25% relative improvement)
   - Transformer R^2: -1.17 → 0.87 (from failure to competitive)

- LSTM R^2: 0.68 $\rightarrow$ 0.89 (+31% improvement)

3. **Transformer rehabilitation:** The standard Transformer's catastrophic failure ($R^2 = -1.17$) is not inherent to the architecture but rather a consequence of single-step input causing self-attention degeneracy. With hierarchical features providing trend context, Transformer achieves $R^2 = 0.87$.

5.3 RQ3: News Events and Stock Reactions

We examined the relationship between major Apple news events and stock price movements. Figure 1 shows that product launch announcements (e.g., iPhone releases, WWDC) produce short-term volatility spikes but limited sustained directional movement.

Figure 1: Model performance comparison. Left panel shows RMSE across model classes. Right panel illustrates the improvement from hierarchical training (56th feature) for neural networks.

Quantitatively, we find:

- Average absolute return on Apple event days: 2.1%
- Average absolute return on non-event days: 1.4%
- Difference: statistically significant ($p < 0.01$)

However, the directional sign of event-day returns is not consistently predictable from pre-event sentiment, limiting the utility for forecasting.

5.4 Return-Level Prediction

To provide context for the high price-level R^2 values, Table 6 reports results for next-day return prediction:

Table 6: Return Prediction Performance

Model	R^2 **(Returns)**	**Directional Accuracy**
Linear Regression	0.084	54.2%
SARIMAX	0.063	53.1%
GRU (hierarchical)	0.071	52.8%
Naive (predict 0)	0.000	52.0%

The return-level R^2 of 0.084 is modest but consistent with prior literature on short-horizon return prediction. The directional accuracy of 54.2% exceeds random chance but remains far from actionable for trading.

6 Limitations

We acknowledge several limitations that qualify our findings:

6.1 Data Limitations

1. **Single-stock study:** Our analysis focuses exclusively on AAPL. As a large-cap, high-liquidity stock with extensive news coverage, AAPL may not represent typical stocks. Generalization to smaller, less-covered stocks requires further validation.

2. **Data source heterogeneity:** Our news corpus combines multiple sources with different collection methodologies, coverage patterns, and potential quality variations over the 26-year period.

3. **Missing data imputation:** With 69% of days lacking dedicated AAPL news, we rely heavily on rolling mean imputation, which may attenuate true sentiment signals.

4. **Survivorship bias:** AAPL's exceptional performance (1,040× return) makes it an outlier. Our hierarchical training approach may be particularly suited to strongly trending assets.

6.2 Methodological Limitations

1. **Stationarity assumptions:** The extreme non-stationarity of AAPL prices (1,040× increase) challenges many statistical assumptions. Our high price-level R^2 values are inflated by this trend.

2. **Look-ahead in feature selection:** While we use walk-forward validation for model evaluation, our feature engineering choices (e.g., 55 features, 7-day window) were informed by the full dataset. True out-of-sample performance may be lower.

3. **No transaction costs:** Our trading strategy evaluation ignores transaction costs, slippage, and market impact that would reduce profitability in practice.

4. **No architectural novelty:** We apply existing ML architectures rather than proposing new ones. Our contribution is methodological (hierarchical training) rather than architectural.

6.3 External Validity

1. **Time period specificity:** Our sample includes the dot-com bubble, 2008 financial crisis, and COVID-19 pandemic—each representing distinct market regimes. Performance in future regimes may differ.

2. **AAPL-specific news dynamics:** Apple's unique position as a consumer-facing technology company with highly anticipated product launches may not translate to other sectors (e.g., industrials, utilities).

7 Conclusion

This paper examines whether incorporating news sentiment and textual features can improve stock price forecasting beyond traditional time series methods. Using 26 years of Apple Inc. data and 57 million financial news articles, we find:

1. **News sentiment provides incremental but meaningful predictive power.** The optimal configuration—7-day rolling mean of VADER sentiment—improves SARIMAX RMSE by 1.5%, statistically significant at the 5% level.

2. **Model complexity should match data availability.** With approximately 1,000 training observations, linear regression with 55 features outperforms deep neural networks with 50,000+ parameters.

3. **Hierarchical training substantially improves neural network performance.** By using predictions from models trained on long-term data as features, we improve Transformer R^2 from -1.7 to 0.87 and GRU R^2 from 0.72 to 0.94.

4. **News events create volatility but not predictable direction.** Apple product launches increase absolute returns but the direction is not consistently predictable from pre-event sentiment.

Our findings have practical implications for quantitative finance practitioners: (1) sentiment features are worth incorporating but should not be expected to transform performance; (2) simpler models often outperform complex ones on limited financial data; and (3) hierarchical training provides a viable path to leveraging deep learning on short time series.

Future research directions include: (1) extending the analysis to multiple stocks and asset classes; (2) incorporating high-frequency intraday data where news effects may be more pronounced; and (3) developing theoretical frameworks for when hierarchical training is most beneficial.

References

Andrade, G., Mitchell, M., and Stafford, E. (2001). New evidence and perspectives on mergers. *Journal of Economic Perspectives*, 15(2):103–120.

Ball, R. and Brown, P. (1968). An empirical evaluation of accounting income numbers. *Journal of Accounting Research*, 6(2):159–178.

Bernard, V.L. and Thomas, J.K. (1989). Post-earnings-announcement drift: Delayed price response or risk premium? *Journal of Accounting Research*, 27:1–36.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Chaney, P.K., Devinney, T.M., and Winer, R.S. (1991). The impact of new product introductions on the market value of firms. *Journal of Business*, 64(4):573–610.

Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). Deep learning for event-driven stock prediction. *IJCAI*, 2327–2333.

Fama, E.F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2):383–417.

Feng, F., Chen, H., He, X., Ding, J., Sun, M., and Chua, T.S. (2019). Enhancing stock movement prediction with adversarial training. *IJCAI*, 5843–5849.

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669.

Lim, B., Arik, S.O., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764.

Tetlock, P.C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, 62(3):1139–1168.

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *NeurIPS*, 5998–6008.

Wu, H., Xu, J., Wang, J., and Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *NeurIPS*, 34:22419–22430.

Xu, Y. and Cohen, S.B. (2018). Stock movement prediction from tweets and historical prices. *ACL*, 1970–1979.

Zhang, Z., Zohren, S., and Roberts, S. (2023). Deep learning for financial time series forecasting: A survey. *Quantitative Finance*, 23(3):451–470.

Zhou, H., Zhang, S., Peng, J., et al. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *AAAI*, 35:11106–11115.

Supplementary Material

Complete code, data preprocessing scripts, and model implementations are available at:

`https://github.com/[repository]/news-enhanced-forecasting`

The repository includes:

- `Run_analysis.py`: Main analysis script
- `src/data_preprocessor.py`: Data collection and preprocessing
- `src/sentiment_comparison.py`: Sentiment extraction (VADER, TextBlob, FinBERT)
- `src/rich_text_features.py`: LDA topic modeling and text features
- `requirements.txt`: Python dependencies
- `README.md`: Detailed usage instructions

All experiments are reproducible with random seed 42.