A Deep Learning Framework for Momentum-Based Trading on US equities Combining Technical and Sentiment Factors

Haewon Jeong[1], Alexandre Goumaz[2], and Philipp Glaser[2]

[1]Robotics, Control, and Intelligent Systems, EPFL
[2]Computer and Communication Sciences, EPFL

May 17, 2026

Abstract

The relationship between financial news and stock price movements is complex and multifaceted, making it difficult to exploit systematically using conventional analytical methods. Early approaches relied on domain-specific sentiment lexicons to quantify the tone of financial text, but these methods are limited by their inability to capture contextual meaning and syntactic nuance. Recent advances in deep learning and natural language processing have opened new avenues for automating this process. In this paper, we propose a sentiment-augmented deep learning framework for momentum-based stock movement prediction on New York Stock Exchange constituents. Financial news articles are processed through a two-stage NLP pipeline consisting of abstractive summarization followed by sentiment encoding via a finance-domain language model, producing rich per-day sentiment representations that are fused with sixteen technical indicators derived from historical price data. Two sequential architectures — an LSTM and a Transformer encoder — are trained to classify next-day price direction from 64-day sliding windows of the combined features. Experimental results on a subset of Russell 1000 stocks demonstrate the effectiveness of integrating deep learning-based sentiment signals with technical analysis for stock movement prediction.

1 Introduction

1.1 Stock news based sentiment analysis

The analysis of textual sentiment in financial markets has a long history, dating back to the development of the General Inquirer by Stone et al. (1966) at MIT, which introduced the concept of matching words against a predefined lexicon — the Harvard IV-4 dictionary —

to quantify the tone of written text. For decades, this general-purpose dictionary served as the default tool for sentiment scoring in finance. However, its direct application to financial text proved problematic. Loughran and McDonald (2011) demonstrated that nearly three-quarters of the words flagged as negative by the Harvard IV-4 dictionary in corporate 10-K filings were not actually negative in a financial context. In response, they introduced the Loughran-McDonald (LM) Master Dictionary, constructed from SEC 10-K filings spanning 1994 to 2008, which quickly became the standard reference for financial sentiment analysis [1, 2]. Despite their wide adoption, pure dictionary-based methods share a fundamental limitation: they treat each word independently, ignoring syntax, negation, and contextual meaning [3]. This motivated the adoption of classical machine learning models such as SVM, Naive Bayes, and Random Forests, trained on labeled financial news corpora to capture more nuanced sentiment patterns [4]. The true paradigm shift came with deep learning-based NLP. Word embeddings (Word2Vec, GloVe) partially addressed context-blindness, followed by the transformer architecture [5] and BERT [6], which enabled bidirectional contextual text representations. These advances catalyzed domain-adapted models for finance: Araci (2019) released FinBERT, fine-tuned on the Financial PhraseBank dataset, and Yang et al. (2020) further extended it on over 46,000 financial news documents [7, 8]. These models consistently outperformed traditional dictionary-based methods, demonstrating the effectiveness of domain-specific pre-training for financial sentiment analysis.

1.2 Deep learning approach for stock price estimation

Traditional approaches to stock price forecasting relied primarily on statistical time series models such as ARIMA and GARCH, which assume linear relationships and stationarity in financial data. While these models offer interpretability and theoretical grounding, they are fundamentally limited in capturing the non-linear, high-dimensional dynamics that characterize real-world equity markets [9]. The advent of deep learning introduced architectures capable of modeling such complexity. Recurrent Neural Networks (RNNs) were among the first deep learning models applied to financial time series, as their sequential processing structure naturally accommodates temporal dependencies in stock price data. However, standard RNNs suffer from the vanishing gradient problem, which impairs learning over long sequences. This limitation was addressed by the Long Short-Term Memory (LSTM) network, introduced by Hochreiter and Schmidhuber (1997) [10], which employs gating mechanisms to selectively retain or discard information across time steps. LSTM-based models have since become the dominant baseline in stock price prediction, consistently outperforming traditional statistical models across a variety of market settings [11]. Convolutional Neural Networks (CNNs), originally developed for image recognition tasks, were subsequently adapted for financial time series by treating sequences of price and indicator data as one-dimensional signals. CNNs excel at extracting local patterns and short-term dependencies through learned filters, offering complementary strengths to the long-range temporal modeling of LSTMs. Comparative studies found that while LSTMs better captured sequential context, CNNs demonstrated advantages in extracting semantic patterns from textual inputs such as news headlines [12]. This observation naturally motivated the development of hybrid architectures. Lu et al. (2020) proposed a CNN-LSTM model in which a CNN layer first extracts feature representations from raw input data, which are then passed to an LSTM for temporal prediction, achieving superior performance over either model in isolation [13]. Beyond simple hybridization, attention mechanisms were introduced to allow models to dynamically weight the relevance of different time steps when making predictions. This concept was formalized in the Transformer architecture by Vaswani et al. (2017) [5], which dispenses with recurrence entirely in favor of self-attention, enabling parallel computation and more effective modeling of long-range dependencies. Transformer-based models were subsequently applied to stock movement prediction, with studies such as Jiang et al. (2022) demonstrating that attention mechanisms provide meaningful gains over LSTM baselines, particularly in volatile market conditions [14]. A comprehensive survey by Shen et al. (2023) further confirmed that recent literature has progressively shifted toward Transformer-based and hybrid architectures, with LSTM, GRU, CNN, and attention-based models remaining the dominant building blocks for state-of-the-art stock forecasting systems [15].

1.3 Sentiment analysis with deep learning approach for stock price estimation

While deep learning models demonstrated strong performance on price-based time series data, a parallel line of research explored whether incorporating textual sentiment signals could further improve predictive accuracy. The underlying hypothesis is grounded in behavioral finance: investor sentiment, as reflected in news coverage and public discourse, exerts measurable influence on short-term price movements beyond what historical price data alone can capture. Early efforts in this direction combined traditional sentiment scores derived from lexicon-based tools with sequential deep learning models. Chen et al. (2018) demonstrated that appending VADER or LM dictionary sentiment scores as additional input features to LSTM networks yielded modest but consistent improvements over price-only baselines [12]. However, the expressiveness of such approaches remained constrained by the limitations of the underlying sentiment extraction method — dictionary-based scores provided only coarse polarity signals, discarding the rich contextual information present in financial news. The introduction of transformer-based language models fundamentally changed this landscape. By replacing rule-based sentiment scores with dense contextual embeddings or fine-grained sentiment probabilities produced by models such as BERT [6] and FinBERT [7], researchers were able to provide downstream prediction models with substantially richer representations of news content. Hiew et al. (2019) constructed a BERT-based financial sentiment index and demonstrated its effectiveness as an input feature for LSTM-based return prediction [16]. Building directly on this line of work, Huang et al. (2022) proposed the FinBERT-LSTM architecture, in which daily sentiment scores extracted from financial news via FinBERT are concatenated with historical price features and fed into an LSTM network, outperforming both standalone LSTM and ARIMA models on the NASDAQ-100 index [17]. Jiang and Zeng (2023) similarly integrated FinBERT sentiment with LSTM, confirming that sentiment-augmented models consistently outperformed price-only baselines and vanilla BERT variants across multiple evaluation metrics [18]. Further refinements addressed the granularity of news integration. Li et al. (2024) introduced a weighted news category scheme within a FinBERT-LSTM framework, distinguishing between market-level, industry-level, and stock-specific news, and showing that hierarchical sentiment aggregation improved prediction accuracy beyond flat sentiment averaging [19]. Complementary work by Kim et al. (2022) combined BERT-derived sentiment scores with a Random Forest classifier for SP 500 direction prediction, demonstrating that transformer-based sentiment features retained predictive value even outside sequential deep learning architectures [20].

1.4 Our approach

Our approach follows the general paradigm of sentiment-augmented deep learning for stock prediction established in prior work [21, 17, 18], but introduces several key distinctions in both the sentiment representation pipeline and the feature fusion strategy. While existing studies typically reduce news sentiment to a scalar polarity score before feeding it into a sequential model, our approach preserves the full richness of the FinBERT output — including a 768-dimensional semantic embedding and a 3-class sentiment probability vector — projects the embedding to a lower-dimensional space via a learned linear layer, and fuses these representations directly with technical indicators at each timestep. Furthermore, rather than encoding raw news text directly, we introduce an intermediate summarization step using BART [22] to condense long-form articles prior to encoding, mitigating the token length constraints of transformer-based encoders and reducing noise from irrelevant content. The overall pipeline consists of five sequential stages — data ingestion, sentiment encoding, feature engineering, model training, and evaluation — each described in detail in the following subsections.

2 Methods

2.1 Data Collection

2.1.1 Price data.

Daily OHLCV bars (open, high, low, close, volume) and volume-weighted average price (VWAP) were obtained from the Alpaca Securities API with split-adjustment applied to ensure price continuity across corporate actions. Pagination was handled automatically, and requests exceeding the API rate limit were retried with exponential back-off.

2.1.2 News data.

Financial news articles were sourced from the Benzinga news feed via the Massive API. Articles were retrieved over the target date range and deduplicated by article identifier, retaining the most recently updated version of each record. Each article contains a headline, full body text, publication date, and one or more associated ticker symbols.

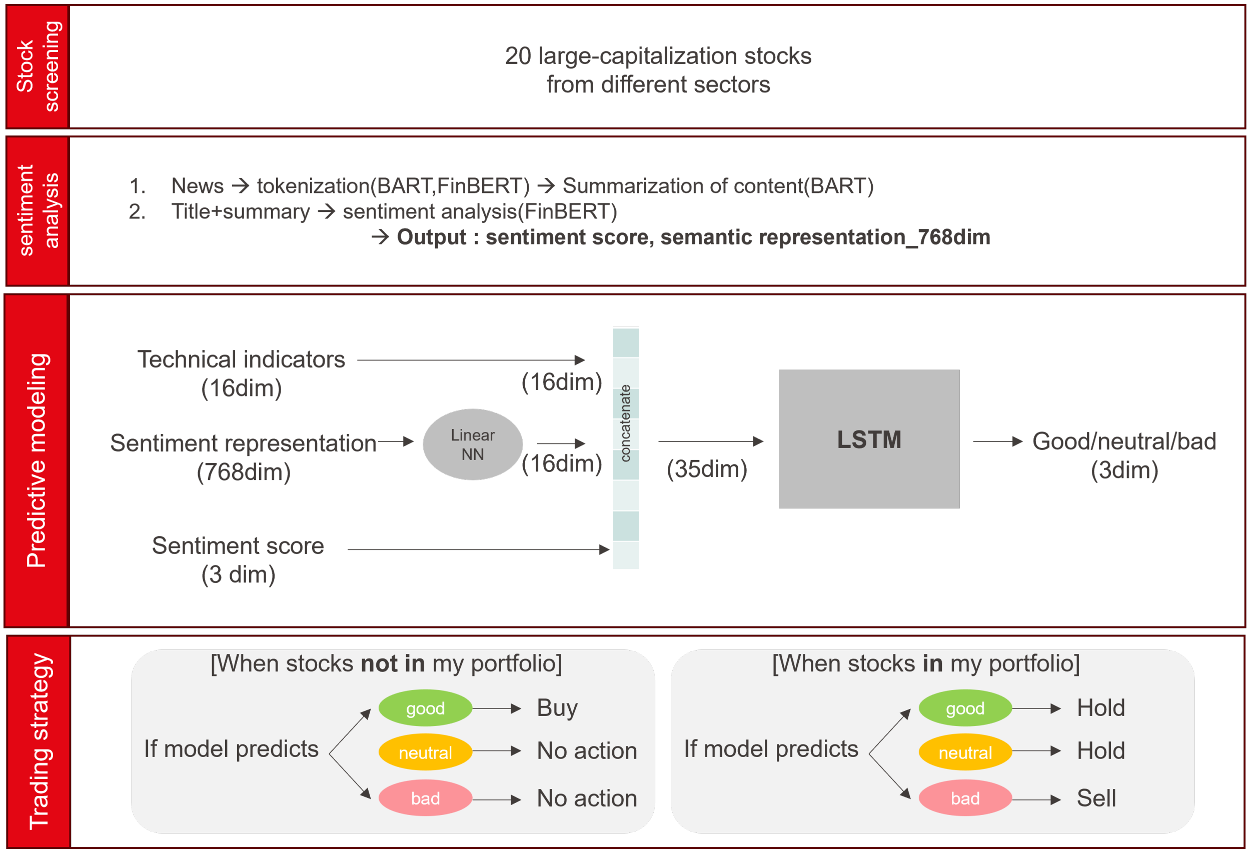

Figure 1: Overall pipeline

2.1.3 Stock screening.

The study universe comprised 20 large-capitalization stocks drawn from the Russell 1000, selected to ensure broad and balanced sectoral coverage. To ensure adequate sentiment signal, only stocks with a mean of at least five Benzinga articles per calendar month over the full study period were retained. An additional momentum gate was applied at inference time, restricting predictions to windows in which the linear-regression slope of the preceding 20 closing prices was strictly positive, thereby limiting the strategy to stocks exhibiting an upward price trend.

2.2 NLP Sentiment Pipeline

Raw articles were processed through a two-stage NLP pipeline to produce fixed-dimensional sentiment representations aligned to each trading day.

Stage 1 — Summarization. Article body text was condensed using BART (facebook/bart-large-cnn) [22] to reduce input length to within the token budget of the downstream encoder and to remove boilerplate content unrelated to the reported financial event. The headline was preserved in full; articles lacking body text were encoded from the headline alone.

Stage 2 — FinBERT encoding. The concatenated headline and summary were encoded with FinBERT (ProsusAI/finbert) [7], a BERT-based language model pre-trained on financial corpora and fine-tuned for three-class sentiment classification. Two representations were extracted per article:

- a 768-dimensional semantic embedding obtained by mean-pooling the final hidden state over non-padding tokens, and
- a three-dimensional softmax probability vector $\mathbf{p} = [P(\text{pos}),\ P(\text{neg}),\ P(\text{neu})]$.

A scalar sentiment score was derived as $s = \mathbf{p} \cdot [1.0,\ 0.0,\ 0.5]$, assigning full weight to positive sentiment and half weight to neutral. Articles that failed to encode were assigned a neutral fallback ($s = 0.5$, zero vectors) to prevent pipeline interruption.

Daily aggregation. When multiple articles were associated with the same ticker on the same trading date, the 768-dimensional embeddings and three-dimensional probability vectors were collapsed into a single daily representation by element-wise averaging, and the scalar score was recomputed from the aggregated probability vector. Days with no news coverage were assigned zero vectors.

2.3 Technical Feature Engineering

Sixteen scale-invariant technical indicators were computed from the daily OHLCV and VWAP data, organized into five groups as summarized in Table 1. All indicators are expressed as ratios, bounded values, or naturally scaled quantities to ensure cross-sectional comparability without loss of relative magnitude. Feature values were subsequently standardized using a StandardScaler fitted exclusively on the training set.

Table 1: Technical Indicators Used as Input Features

Group	**Count**	**Indicators**
Trend	4	Close/SMA_5 ratio, Close/SMA_{20} ratio, Close/SMA_{60} ratio, MACD
Momentum	5	MACD signal line, RSI_{14} (normalized to $[0, 1]$), Stochastic %K, Stochastic %D, Rate of Change_{10}
Volatility	3	ATR_{14}/close (normalized), Bollinger %B, Bollinger Band width
Volume	3	Volume/SMA_{20} ratio, OBV linear-regression slope over 10 days, VWAP/close ratio
Returns	1	Log return
Total	**16**	

2.4 Dataset Construction and Temporal Splitting

Prediction target. The binary classification label for day t was defined as:

$$y_t = \begin{cases} 1 & \text{if } \dfrac{\text{close}_{t+2} - \text{close}_t}{\text{close}_t} > \tau \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

This three-day executable signal incorporates a one-day execution lag following signal generation, consistent with the formulation proposed in [21].

Sliding windows. Input sequences were constructed using a sliding window of $W = 64$ trading days with stride 1. Each sample comprises a window of technical indicators, FinBERT embeddings, and sentiment probability vectors, labeled by the target computed at the terminal day of the window.

Train/validation/test splitting. To prevent look-ahead bias, a strict temporal split was applied. Data spanning 2018 to 2023 constituted the training set, while data from 2024 to 2025 were reserved for validation and testing. Windows preceding the cutoff by more than two months constituted the training set; the two months immediately before the cutoff formed the validation set; all subsequent windows formed the test set. An additional 20% of symbols were withheld entirely as a held-out evaluation set. The StandardScaler was fitted exclusively on training windows and applied without refit to all other partitions.

2.5 Model Architectures

Two architectures were implemented and evaluated, both accepting the same three-stream input: technical indicators $\mathbf{T} \in \mathbb{R}^{W\times 16}$, FinBERT embeddings $\mathbf{S} \in \mathbb{R}^{W\times 768}$, and sentiment probability vectors $\mathbf{P} \in \mathbb{R}^{W\times 3}$. Prior to fusion, the 768-dimensional embedding stream was projected to $\mathbb{R}^{16}$ via a learned linear layer to align its dimensionality with the remaining inputs. The projected embedding, technical indicators, and sentiment probabilities were then concatenated along the feature axis, yielding a 35-dimensional per-timestep representation of shape $(N,\ W,\ 35)$.

SentimentLSTM. The fused sequence was processed by a two-layer LSTM with hidden size 32 and dropout probability 0.2. The hidden state at the final timestep was forwarded to a classification head comprising Linear($32 \to 32$) $\to$ ReLU $\to$ Dropout(0.2) $\to$ BatchNorm1d $\to$ Linear($32 \to 1$).

SentimentTransformer. The fused sequence was linearly projected to model dimension $d_{\text{model}} = 64$ and augmented with learned positional embeddings. A Transformer encoder [5] comprising 6 layers, 4 attention heads, and feedforward dimension 128 then processed the full sequence, allowing each timestep to attend globally to all others. The encoder output was mean-pooled across the temporal dimension and passed to a linear classifier to produce the final logit.

Both models produce a scalar logit transformed to a prediction probability via $\hat{y} = \sigma(\text{logit})$.

2.6 Training Procedure

Both models were trained to minimize binary cross-entropy loss (BCEWithLogitsLoss) using the Adam optimizer with an initial learning rate of 10^{-3}. A ReduceLROnPlateau scheduler reduced the learning rate by a factor of 0.5 upon five consecutive epochs without improvement in validation AUC. Gradient norms were clipped to 1.0 to stabilize optimization. Early stopping with a patience of 15 epochs was applied based on validation AUC, and the checkpoint attaining the highest validation AUC was restored prior to evaluation. All models were trained for a maximum of 100 epochs with a batch size of 32.

2.7 Evaluation

Generalization performance was assessed on the held-out test set using bootstrap resampling ($n = 1{,}000$ iterations) to construct 95% confidence intervals for four metrics: area under the receiver operating characteristic curve (AUC), accuracy, precision, and recall. AUC was designated as the primary metric on account of its threshold independence and robustness to class imbalance. Bootstrap resamples containing only a single class, for which AUC is undefined, were discarded; the number of such discarded resamples is reported alongside results.

2.8 Trading Strategy

The trained model generates a daily binary signal for each stock in the universe. A buy order is executed when the model predicts an upward price movement and the position is not already held. Conversely, if the stock is currently held and the model does not predict an upward movement, the position is liquidated. If no signal is generated, the existing position is maintained. This simple long-only strategy is summarized as follows:

- **Buy:** model predicts upward movement and no position is held.
- **Sell:** model predict downward movement and a position is currently held.
- **Hold:** no change in signal; existing position is maintained.

No short selling is permitted, and at most one unit of each stock is held at any given time. Transaction costs and slippage are not modeled in the current implementation.

3 Results

3.1 Predictive Performance

The SentimentLSTM model was evaluated on two held-out partitions: a temporal test set comprising unseen time periods from the same training symbols, and a held-out symbol set comprising stocks entirely excluded from training.

Table 2: Evaluation results on temporal test and held-out symbol sets.

Set	AUC	Accuracy	Precision	Recall	N
Temporal test	0.505 [0.503, 0.507]	0.520	0.525	0.874	384k
Held-out symbols	0.523 [0.521, 0.525]	0.530	0.533	0.885	297k

As shown in Table 2, both evaluation sets yielded AUC scores marginally above the random baseline of 0.5: 0.505 [0.503, 0.507] on the temporal test set and 0.523 [0.521, 0.525] on the held-out symbol set. These results indicate that the model failed to acquire statistically meaningful predictive signal. The pattern of high recall (0.874 and 0.885, respectively) coupled with near-chance precision (0.525 and 0.533) further suggests that the model collapsed toward a majority-class prediction strategy, assigning positive labels to the majority of windows rather than discriminating on learned features. Consistent with this interpretation, early stopping triggered at epoch 3 of 18, indicating that validation performance plateaued almost immediately and that further training yielded no improvement.

3.2 Analysis of the result

The near-random AUC scores are broadly consistent with the Efficient Market Hypothesis, which holds that asset prices rapidly incorporate all available public information, rendering history-based prediction intractable in expectation. Several additional factors may have compounded this difficulty.

First, the marginal upward bias (AUC > 0.5) observed across both evaluation sets is likely attributable to a long-side bias in the data rather than genuine learned signal. The test period of 2024–2025 coincided with a sustained broad market rally driven by AI-related capital expenditure and rate cuts initiated by the Federal Reserve, producing an environment in which the majority of windows carried a positive label. A model predicting upward movement indiscriminately would therefore achieve above-chance accuracy and recall under these conditions.

Second, the test period represented a distinct market regime relative to the training window of 2018–2023. The 2024–2025 period was characterized by macroeconomic forces, including shifting Federal Reserve policy, AI-driven sector concentration, and heightened trade policy uncertainty, that were structurally different from the rate-hiking and pandemic-recovery dynamics of the training period. Technical indicators and sentiment embeddings derived from historical patterns would plausibly generalize poorly across such a regime shift.

Third, and notably, performance on the held-out symbol set (AUC 0.523) slightly exceeded that on the temporal test set (AUC 0.505). This counterintuitive result suggests that cross-sectional generalization to unseen stocks was not the primary source of failure; rather, temporal distribution shift appears to have been the more significant limiting factor.

3.3 Future Directions

Several extensions may improve predictive performance in future work. Incorporating macroeconomic conditioning variables, such as interest rate regime indicators, sector-level momentum signals, or policy uncertainty indices, could help the model adapt to structural market regime shifts that pure price-based features cannot capture. Richer news representations, such as article-level entity linking or event extraction beyond sentence-level sentiment, may provide more discriminative signal than aggregated FinBERT embeddings. Finally, training on a larger and more sectorally diverse universe, and applying domain-adaptive fine-tuning when deploying across regime boundaries, represent promising directions for improving temporal generalization.

4 Conclusion

In this work, we investigated whether an LSTM-based model integrating technical indicators and NLP-derived sentiment features could capture generalizable patterns sufficient for stock price movement prediction. We constructed a large-scale dataset spanning 20 large-capitalization stocks with over 1.2 million training and evaluation windows, providing a rigorous empirical foundation for our analysis.

Our results indicate that the model did not achieve meaningful predictive performance. The temporal test set yielded an AUC of 0.505 and the held-out symbol set an AUC of 0.523, both marginally above the theoretical random baseline of 0.5. The observed pattern of high recall coupled with low precision suggests that the model collapsed toward a majority-class prediction strategy rather than learning discriminative temporal features, and early stopping at epoch 3 of 18 confirms that meaningful generalization did not occur.

These findings are consistent with the Efficient Market Hypothesis, which posits that asset prices rapidly incorporate all available information, rendering purely history-based prediction statistically intractable. The test period of 2024–2025 further presented a distinct market regime relative to the training window, characterized by AI-driven sector concentration, shifting Federal Reserve policy, and heightened trade policy uncertainty, compounding the difficulty of temporal generalization.

Nonetheless, our result is itself informative. The counterintuitive finding that held-out symbol performance exceeded temporal test performance suggests that cross-sectional generalization is more tractable than adaptation across market regimes, a distinction that has practical implications for model deployment. Future work incorporating macroeconomic conditioning variables, richer event-level news representations, and regime-adaptive training procedures may yield more robust predictive performance.

References

[1] Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, 66(1):35–65, 2011.

[2] Tim Loughran and Bill McDonald. The use of word lists in textual analysis. *Journal of Behavioral Finance*, 16(1):1–11, 2015.

[3] Colm Kearney and Sha Liu. Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis*, 33:171–185, 2014.

[4] Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news. *ACM Transactions on Information Systems*, 27(2):1–19, 2009.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[7] Dogu Araci. FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[8] Yi Yang, Mark Christopher Siy Uy, and Allen Huang. FinBERT: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*, 2020.

[9] Sidra Mehtab and Jaydip Sen. Stock price prediction using CNN and LSTM-based deep learning models. *arXiv preprint arXiv:2010.13891*, 2020.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[11] Sreelekshmy Selvin, R. Vinayakumar, E. A. Gopalakrishnan, Vijay Krishna Menon, and K. P. Soman. Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1643–1647. IEEE, 2017.

[12] Sethia Mohan, Sahithi Mullapudi, Sudheer Sammeta, Venkataramana Palagiri, and Tugba Taskaya-Temizel. Deep learning for stock market prediction using technical indicators and financial news articles. In *2018 IEEE 10th International Conference on Dependable, Autonomic and Secure Computing*, pages 1–8. IEEE, 2018.

[13] Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, and Jingyang Wang. A CNN-LSTM-based model to forecast stock prices. *Complexity*, 2020:1–10, 2020.

[14] Minqi Jiang, Jianguo Liu, Lu Zhang, and Chunyu Liu. Incorporating transformers and attention networks for stock movement prediction. *Complexity*, 2022:1–10, 2022.

[15] Jiasheng Shen et al. Stock market prediction via deep learning techniques: A survey. *arXiv preprint arXiv:2212.12717*, 2023.

[16] Joshua Zoen-Git Hiew, Xin Huang, Hao Mou, Duan Li, Qi Wu, and Yabo Xu. BERT-based financial sentiment index and LSTM-based stock return predictability. 2019.

[17] Xinyi Huang et al. FinBERT-LSTM: Deep learning based stock price prediction using news sentiment analysis. *arXiv preprint arXiv:2211.07392*, 2022.

[18] Tingsong Jiang and Qingyun Zeng. Financial sentiment analysis using FinBERT with application in predicting stock movement. *arXiv preprint arXiv:2306.02136*, 2023.

[19] Xin Li et al. Predicting stock prices with FinBERT-LSTM: Integrating news sentiment analysis. In *Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing*, pages 1–7. ACM, 2024.

[20] Hyun-jung Kim et al. Using financial news sentiment for stock price direction prediction. *Mathematics*, 10(13):2156, 2022.

[21] Wang Li, Chaozhu Hu, and Youxi Luo. A deep learning approach with extensive sentiment analysis for quantitative investment. *Electronics*, 12(18):3960, 2023.

[22] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghahrabi, Abdelrahman Mohamed, et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.