Sentiment-Enhanced Multivariate Time Series Forecasting for Stock Price

Bryan Huang, Jin Zhang, Thomas Nussbaum

ABSTRACT

Forecasting stock prices is a long-standing, unsolved challenge due to the complex nature of the financial market. Where traditional models primarily rely on historical price and volume data, recent advances in NLP have enabled the incorporation of external signals, such as sentiment, as input features into forecasting models. In this project, we investigate the effectiveness of combining sentiment analysis with time series forecasting using a Long Short-Term Memory (LSTM) model. Sentiment scores are extracted from financial news headlines using FinBERT [1], a domain-specific language model. We then conduct an ablation study comparing the forecasting of detrended stock prices with and without sentiment scores. Our results indicate that incorporating additional features—particularly sentiment scores—can modestly enhance the accuracy of short-term forecasts. These results support the notion that integrating structured financial data with unstructured textual signals can lead to more effective forecasting models.

1 METHODOLOGY

1.1 Datasets

This project uses 2 main data sources: one for our structured financial data and another for unstructured textual sentiment.

1. Stock Price Data

We obtain daily stock metrics from **Yahoo Finance**, using the daily *closing price* as the basis for our target variable. To capture short-term fluctuations, the closing price is detrended by subtracting a 30-day moving average. The dataset is further preprocessed to address missing values and maintain a consistent daily frequency.

2. Financial News Sentiment Data

For textual data, we use the **FNSPID** [2] dataset, a large-scale corpus of financial news headlines aligned with S&P 500 companies. The dataset provides headline-level data with associated dates, alongside generated article summaries.

1.2 Sentiment Analysis

The objective of this part is to obtain a time series containing the most relevant information related to the stock we want to forecast, which we will later add to our model to improve its performance. This time series needs to be as close as possible to the real sentiment of the market for a given stock.

1.2.1 General idea and mathematical framework. Most sentiment-based models in finance filter the news they use by keeping only those containing specific keywords such as the stock name, or by classifying them by sector. In our model, we decided to try another approach—one that weights the news by relevance and publication date, in order to use the maximal amount of information with a reasonable amount of noise and fast computation. Indeed, our model is adapted for high- or at least medium-frequency trading.

To achieve that goal, we propose the following expression for our sentiment time series:

$$s(t) = \frac{\sum_{t_i : t_i \leq t} f(r_i, s_i)\phi(t - t_i)}{\sum_{t_i : t_i \leq t} r_i \phi(t - t_i)}$$

where:

- t_i: publication time of news i
- r_i: relevance score of news i
- s_i: sentiment score of news i
- f: a function of two variables, $f(r_i, s_i)$, representing the relevance-weighted sentiment at the time of publication of news i
- ϕ: a function of one variable, $\phi(t - t_i)$, representing the decay of influence of the news with time

The denominator normalizes the expression.

1.2.2 Sentiment scores. To obtain the s_i, we aim to find a model that gives a good approximation of the sentiment of a short financial text. We decided to use FinBERT because of its low weight, computational efficiency, and financial specialization. FinBERT is a BERT (Bidirectional Encoder Representations from Transformers) model fine-tuned for financial text analysis.

FinBERT takes a text as input, and its softmax output gives a three-dimensional vector containing the probabilities of the text being positive, negative, or neutral from a financial perspective. We denote them respectively as P^i_{pos}, P^i_{neg}, and P^i_{neu}, where i is the index of the news.

To find a suitable expression for s_i, we could let the model learn the best function of the three probabilities. However, for the sake of simplicity and computational efficiency, we decided to choose a predefined function among a list of candidate functions. Two of them gave better results: $P^i_{\text{pos}} - P^i_{\text{neg}}$ and $(P^i_{\text{pos}} - P^i_{\text{neg}}) \times (1 - P^i_{\text{neu}})$. We kept the first one, as it is simpler and more commonly used.

1.2.3 Relevance scores. There are several ways to define a relevance score. We opted for a simple and computationally fast method. It consists in providing a short description of the company whose stock we want to forecast — for example, for Google: "Alphabet Inc. (Google) is a global technology company specializing in internet-related services and products, including search, advertising, cloud computing, software, and hardware. Its core products include Google Search, YouTube, Android, Google Cloud, and more." — then taking its FinBERT embedding and computing the cosine similarity with the embeddings of the news. Mathematically:

$$r_i = \cos(\text{FinBERT}(d), \text{FinBERT}(\text{news}_i))$$

where d is the description of the company and news_i is the title of news i.

1.2.4 Choice of f. To choose f, we tested several functions and ultimately selected a simple one:

$$f(r_i, s_i) = \begin{cases} r_i \times s_i & \text{if } r_i > 0.2 \\ 0 & \text{otherwise} \end{cases}$$

We drop news with a relevance score below 0.2 to limit noise.

1.2.5 Choice of ϕ. Our first idea for ϕ was inspired by Hawkes processes, which are often used in quantitative finance to model excitation on order books. Their intensity function with an exponential kernel looks like the following plot:

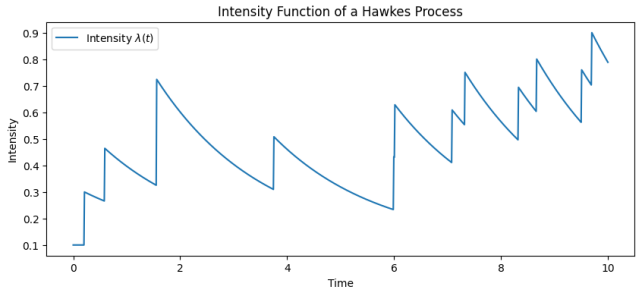

Figure 1: Intensity function of a Hawkes process

Our intuition is that the influence of news decays over time, likely exponentially, after its publication date. So that $\sum_{t_i:t_i \le t} f(r_i, s_i) \exp(-\beta(t - t_i))$ would have the same shape as a Hawkes process with an exponential kernel, with β being a parameter to determine. It would be interesting to explore how β depends on news i and whether the assumption that β is independent of the news is too strong.

However, this model is most useful when time intervals are short (e.g., a few minutes), but we did not find enough intraday data to test it. Therefore, we opted for a simpler function with no parameters:

$$\phi(t - t_i) = \delta_{t-t_i,\,0}$$

meaning that to calculate $s(t)$, we use only the news published on day t.

1.2.6 Final model. Based on the previous sections, the final expression for $s(t)$ is:

$$s(t) = \frac{\sum_{t_i:t_i=t} f(r_i, s_i)}{\sum_{t_i:t_i=t} r_i}$$

1.2.7 Potential improvement ideas. An interesting idea would be to use the news embeddings directly—for example, those given by FinBERT—without relying on the intermediate variables r_i and s_i:

$$s(t) = \sum_{t_i:t_i \le t} f(\text{FinBERT}(\text{news}_i))\, \phi(t - t_i)$$

where f is a function of n variables, with n being the size of the embedding vector, and $s(t)$ is the non-normalized sentiment score. There are multiple ways to define f, and we present two.

The first approach is to define f as a nonlinear function using a two-layer MLP:

$$\begin{aligned} x_i &= \text{FinBERT}(\text{news}_i) \in \mathbb{R}^n, \\ h_i^{(1)} &= \sigma(W^{(1)} x_i + b^{(1)}), \\ h_i^{(2)} &= \sigma(W^{(2)} h_i^{(1)} + b^{(2)}), \\ f_\theta(x_i) &= w_{\text{out}}^\top h_i^{(2)} + b_{\text{out}}, \end{aligned}$$

where $\sigma(\cdot)$ is the ReLU activation, and

$$\theta = \{W^{(1)},\, b^{(1)},\, W^{(2)},\, b^{(2)},\, w_{\text{out}},\, b_{\text{out}},\, \beta\}.$$

We then define the time-decayed sentiment signal as:

$$s(t) = \sum_{t_i:t_i \le t} f_\theta(x_i)\, \phi(t - t_i), \quad \text{with} \quad \phi(\tau) = \exp(-\beta\tau).$$

Finally, we fit the model by minimizing the following loss on future returns:

$$\mathcal{L}(\theta) = \sum_{k=1}^{K_{\text{train}}} \left[r(t^{(k)} + \Delta) - s(t^{(k)}; \theta) \right]^2 + \lambda \|\theta\|_2^2.$$

This method has a higher computational cost and requires more data, but is likely more accurate.

A lighter model could be to replace the MLP $f_\theta(x)$ with a simple linear map:

$$f_w(x) = w^\top x + b, \qquad w \in \mathbb{R}^n,\ b \in \mathbb{R}.$$

$$x_i = \text{FinBERT}(\text{news}_i) \in \mathbb{R}^n, \quad f_w(x_i) = w^\top x_i + b.$$

and perform regression on the same loss function defined above.

1.3 Forecasting Model

To perform our forecasting, we chose to use a Long Short-Term Memory (LSTM) model. LSTM is a variant of recurrent neural networks (RNNs) specifically designed to address the limitations of standard RNNs in capturing long-term dependencies in sequential data. It achieves this through a gated architecture that allows the model to selectively retain or forget information across time steps. This is particularly useful for financial time series, where price movements are influenced by both short-term fluctuations and delayed responses to past events.

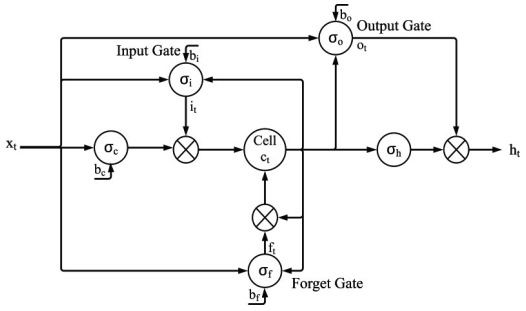

Figure 2: Demonstration of the LSTM architecture from Gao, Chai & Liu (2017)

Compared to traditional models like ARIMA or VAR, which assume linearity and stationarity, LSTM offers superior flexibility in

modeling complex, nonlinear relationships. Additionally, while classical machine learning models such as Random Forests or XGBoost can handle multivariate inputs, they do not inherently capture temporal dependencies, often requiring extensive feature engineering. LSTM, by contrast, learns both the sequence structure and inter-variable interactions directly from the data, making it well-suited for forecasting based on multiple correlated signals such as price, volume, and sentiment.

Although Transformer-based models have shown strong performance in time series tasks due to their ability to model long-range dependencies through self-attention mechanisms, their advantages are more pronounced in settings with large datasets or where long-horizon forecasting is required. In short-term stock prediction—where recent market behavior tends to dominate—such long-range context is often less critical. Additionally, Transformers are typically more resource-intensive and sensitive to hyperparameter choices. In this context, LSTM provides a balanced and efficient approach well-suited to the forecasting task at hand.

1.4 Ablation Study

To further see a relationship between stock price and sentiment data, we compute a detrended price volatility series by subtracting the 30-day simple moving average from each day's closing price. This produces a zero-mean time series of short-term deviations, which we then scale to unit variance.

All experiments use daily data from November 2021 through October 2022, with an 80/20 train/test split on clip-aligned windows. We process sentiment data daily using the stock market's closing time as our timestamp. To handle weekend publications, all news items from Saturday and Sunday are attributed to the next Monday's sentiment feature. We trained two LSTM variants on each of Tesla, Google, and NVIDIA. The two LSTM variants—(1) the stock price only baseline trained on detrended close-price deviations, and (2) the stock price only + Sentiment model that ingests both detrended volatility and z-normalized daily news sentiment. Both variants share the same architecture (2 × LSTM layers, 64 units each, dropout = 0.2) and training hyperparameters.

1.5 Evaluation Metric

To assess each ablation's performance, we use Root Mean Squared Error (RMSE) as the primary evaluation metric. RMSE quantifies the square root of the average squared differences between predicted and actual stock prices. It is preferred in this context because it penalizes larger errors more heavily, which is important in financial forecasting where even small deviations can have significant implications. Additionally, RMSE is expressed in the same units as the target variable, making it intuitively interpretable and suitable for comparing model outputs directly against actual stock prices.

2 PERFORMANCE ANALYSIS

Table 1 presents the RMSE and MAE achieved by two LSTM variants—one using only the detrended close-price volatility series and the other augmenting that series with daily news-sentiment scores—across Google, Tesla, and NVIDIA. For Google, the addition of sentiment yields a marked improvement: RMSE falls from 3.138 to 2.742 and MAE from 2.389 to 2.141. This suggests that news-driven sentiment provides complementary predictive information, enabling the model to better anticipate intraday price deviations beyond what is captured by short-term volatility alone.

By contrast, Tesla exhibits a slight performance degradation when sentiment is included: RMSE increases from 8.596 to 8.760 and MAE from 7.585 to 7.826. We hypothesize that Tesla's highly volatile news cycle—and the prevalence of informal social-media commentary—introduces noise into our news-sentiment pipeline, which can outweigh any marginal signal. Finally, NVIDIA benefits again from sentiment augmentation, with RMSE decreasing from 0.778 to 0.713 and MAE from 0.632 to 0.586. NVIDIA's more structured corporate and product announcements appear to yield cleaner sentiment signals that enhance forecast accuracy. Collectively, these results underscore that while sentiment integration can meaningfully reduce forecast error for some equities, its effectiveness depends on the alignment and volatility of each stock's news environment.

Variant	Google		Tesla		NVIDIA	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Price Only	3.138	2.389	8.596	7.585	0.778	0.632
Price + Sentiment	2.742	2.141	8.760	7.826	0.713	0.586

Table 1: Models Forecasting Results

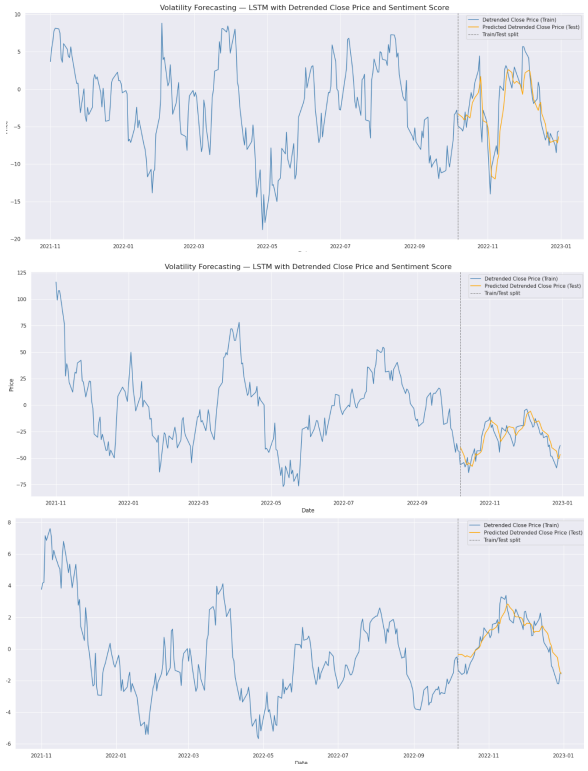

Figure 3: Stock Price Forecast Results (GOOG, TSLA, NVDA)

The three figures provided illustrate the volatility forecasting results for Google, Tesla, and Nvidia using an LSTM model trained on detrended close price and sentiment score data. For Google, the LSTM model demonstrates strong alignment with the actual detrended close price during the test period. The predicted curve closely follows the true trend, successfully capturing both the amplitude and direction of price movements post-split. Tesla's chart presents a more volatile pattern compared to Google. The amplitude of the detrended close prices is significantly higher, with rapid changes and more pronounced peaks and troughs. While the predicted line does generally follow the trend of the actual prices, it exhibits more lag and deviation, especially at sharp turning points. This can be attributed to the higher intrinsic volatility of Tesla's stock, which is harder to capture using LSTM models trained on smoothed sentiment features. Nvidia's forecast reveals also strong results. The predicted detrended close prices closely track the actual values throughout the test window, with a consistent phase and amplitude match. The stock appears to have moderately smooth fluctuations, and the model effectively captures the upward and downward trends.

3 DISCUSSION

3.1 Positive Aspects

The calculation of sentiment scores from financial news effectively captured market mood, consistent with prior studies demonstrating the predictive power of public sentiment in financial markets. By aggregating sentiment over a seven-day window, the model identified a reliable correlation with GOOGL stock price movements, as evidenced by the Mean Absolute Error (MAE) derived from visualizations comparing normalized stock prices and smoothed sentiment. This correlation highlights the value of news sentiment as a predictive feature, particularly when processed to reduce noise.

Feature engineering significantly enhanced the model's ability to uncover relationships between stock prices and sentiment. By detrending prices to isolate volatility (subtracting a 30-day moving average), the study revealed meaningful associations between price fluctuations and sentiment signals. Visualizations comparing detrended prices and sentiment further confirmed this relationship, supporting the hypothesis that non-price features improve forecasting accuracy (Tetlock, 2007). These findings suggest that sentiment-augmented models can offer robust insights into market dynamics.

Theoretically, the model's capacity to leverage sentiment-price relationships could yield substantial financial returns if applied in a real-time trading context. While not implemented within this study, the predictive performance on the test set and the seven-day forecast indicate potential for significant investment gains, particularly in stable market conditions.

3.2 Challenges

Several challenges constrained the study's scope and performance. Limited data availability posed a significant obstacle, particularly the lack of minute-level news sentiment data. The FNSPID dataset provided only daily sentiment scores and included gaps, reducing the effective dataset to approximately 300 trading days. This sparsity hindered the model's ability to capture intraday sentiment dynamics, which are critical for high-frequency trading strategies.

The computational intensity of generating sentiment scores further limited the dataset's size. Processing financial news to compute sentiment scores via natural language processing required substantial time, resulting in fewer data points for model training. With a 30-day lookback window and a 90/10 train-test split, the LSTM model had limited training sequences, constraining its ability to learn complex patterns. Future studies could employ cloud-based processing or precomputed sentiment datasets to increase data density.

The model also faced difficulties in forecasting during periods of high stock price volatility, a common challenge in financial time-series modeling. Market turbulence, potentially driven by macroeconomic events or company-specific news, led to increased prediction errors, as reflected in the test set metrics. The reliance on smoothed sentiment may have reduced responsiveness to rapid sentiment shifts, while detrending could have oversimplified price dynamics in volatile conditions. Incorporating additional market indicators, such as momentum or volatility measures, and explicitly modeling volatility could enhance performance.

4 CONCLUSION

This study set out to determine whether market sentiment can enhance short-term stock-price forecasts when fused with traditional price-based signals in an LSTM framework. By building a relevance-weighted, daily sentiment series with FinBERT scores, we created a compact but information-rich feature that complements detrended close-price volatility. Our ablation experiments across Google (GOOG), Tesla (TSLA) and NVIDIA (NVDA) show that introducing sentiment reduced forecast error for two of the three equities—yielding RMSE decreases of 12.60% for Google and 8.4% for NVIDIA—while marginally degrading performance for Tesla. These mixed yet mostly positive results indicate that textual signals can indeed sharpen price forecasts, but that their efficacy is tightly coupled to the noise characteristics of each firm's news ecosystem.

Beyond empirical gains, the work demonstrates a lightweight pipeline that can be applied to other assets with modest computational overhead: sentiment extraction with a domain-specific language model, a simple but effective cosine-similarity filter to discard off-topic articles, and an LSTM architecture that remains tractable on daily data. This design choice strikes a balance between model expressiveness and operational feasibility—a consideration that is often overlooked in finance where latency and cost matter as much as raw accuracy.

Nevertheless, several constraints temper the generality of our findings. The reliance on daily sentiment snapshots masks intraday mood swings that can move prices within hours or even minutes. Dataset sparsity, driven by gaps in the FNSPID corpus and the compute cost of embedding headlines, limits the model's ability to learn richer temporal patterns—especially for highly volatile stocks such as Tesla. Finally, detrending with a fixed 30-day moving average may oversimplify structural regime shifts, and the LSTM's sequential bias may under-represent long-range cross-correlations that modern Transformer architectures capture more naturally.

REFERENCES

[1] Dogu Araci. 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv:1908.10063 [cs.CL] https://arxiv.org/abs/1908.10063

[2] Zihan Dong, Xinyu Fan, and Zhiyuan Peng. 2024. FNSPID: A Comprehensive Financial News Dataset in Time Series. arXiv:2402.06698 [q-fin.ST] https://arxiv.org/abs/2402.06698