Quantitative Sentiment Trading: Model Evaluation, Strategy Generation, and Backtesting

Dipen Prajapati
Department of Electrical and Computer Engineering
Rutgers University - New Brunswick

***Abstract*—This paper presents a two-stage pipeline that combines deep learning with traditional financial modeling for quantitative sentiment-based trading. In the initial step, we compare a few sentiment classification models, such as Logistic Regression, Naive Bayes, Support Vector Machine, Random Forest, and FinBERT, that are trained and tested on a labeled Kaggle financial news dataset to determine the top classifier. FinBERT, a transformer model pre-trained on financial text, is fine-tuned and outperforms all models compared. In the second phase, the fine-tuned FinBERT is employed to predict sentiment from real-time unseen news headlines. These sentiment predictions are aggregated with technical indicators to generate daily trading signals such as BUY/HOLD/SELL. A custom backtesting engine simulates trades on these signals using historical prices. The findings show that integrating domain-specific NLP models and price-based indicators can greatly improve decision-making in systematic trading approaches.**

***Index Terms*—Quantitative Trading, Sentiment Analysis, NLP, FinBERT, Technical Indicators, Algorithmic Trading, Backtesting, Transformer Models**

I. Introduction

Financial markets are highly sensitive to information, particularly news that reflects investor sentiment, economic developments, and corporate events. Stock prices can be greatly influenced by the timely provision of news, and thus sentiment analysis constitutes an important part of predictive trading strategy development. With the emergence of Natural Language Processing (NLP) techniques, especially transformer-based architectures, there has been increased enthusiasm in applying sentiment analysis to financial decision-making.

Traditional machine learning techniques such as Logistic Regression, Naive Bayes, Support Vector Machines (SVM), and Random Forests, have been widely utilized for text categorization, including sentiment analysis. While these models perform well with structured datasets as well as provide interpretability, they often struggle to capture the context-dependent semantics of financial language. FinBERT, a pre-trained version of BERT on financial texts, proved to be better performing with domain-specific sentiment categorization.

In this work, we present a sentiment-driven trading pipeline to generate and examine trading signals in a structured manner. In the first phase, we contrast several traditional machine learning models, namely Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest, with FinBERT, a pre-trained transformer model on financial text. FinBERT is further fine-tuned on a labeled financial news dataset downloaded from Kaggle to prepare it for the task of sentiment classification. When tested, we find that FinBERT outperforms all baseline models regarding accuracy and other metrics and is the most suitable model for deployment in real-world financial systems.

In the second stage, the fine-tuned FinBERT model is utilized to identify the sentiment of real-time, unseen news headlines collected using NewsAPI. These sentiment scores are then combined with momentum-based technical indicators, namely, the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) to produce composite trading signals that are either BUY, HOLD, or SELL in classification. This multiple-parameter decision-making process maximizes the success of trading activities by enabling traders to make effective choices based on textual as well as price information. The produced signals are evaluated with a specialized backtesting framework that simulates trading activities using historical stock price information. The resultant strategy realizes high-performance operation, generating high returns for the portfolio and affirming the effectiveness of combining domain-specific language models with technical indicators in quantitative trading systems.

II. Dataset and Preprocessing

Our trading pipeline relies on three main data sources: (1) a labeled financial news dataset for model training, (2) real-time financial headlines for sentiment inference, and (3) historical stock price data for strategy backtesting.

A. Labeled Financial News Dataset (Kaggle)

To train and test sentiment classification models, a publicly available dataset from Kaggle is used, comprising 10,688 financial news headlines labeled as *positive*, *neutral*, or *negative*. Each sample is a short sentence or headline pertaining specifically to the financial markets. Standard preprocessing steps are performed to ready the text, including conversion to lowercase, removal of punctuation, removal of links and digits, and normalization of whitespace. The dataset is then transformed to numerical labels using a sentiment-to-index dictionary mapping and split into training and testing subsets in a ratio of 80:20. Traditional models use TF-IDF feature vectors, while FinBERT uses tokenized input processed through a pre-trained tokenizer obtained from Hugging Face.

B. Real-Time News Headlines (NewsAPI)

For real-world sentiment testing and inference, we obtain unseen financial headlines using the NewsAPI. Articles are collected from a wide range of domains (e.g., `bbc.co.uk` and `techcrunch.com`) and news sources (e.g., BBC News and The Verge), with English language and coverage filters applied. Each headline is preprocessed in the same way as the training set and then run through the fine-tuned FinBERT model to generate sentiment scores. An example article used during testing includes:

"President Donald J. Trump Secures Historic \$600 Billion Investment Commitment in Saudi Arabia"

C. Stock Market Data (Yahoo Finance)

To enable the generation of trade signals, historical OHLCV data for individual stock tickers is acquired from Yahoo Finance. The stock ticker is extracted for each news headline using an anchor dictionary that maps the names of the firms to their respective ticker symbols. Having identified the ticker, along with the news date, we then go ahead to download the last 60 days' worth of OHLC data for the particular ticker. The past price data is then utilized to calculate technical indicators, namely Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD). These indicators are then coupled with the forecasted sentiment to generate an absolute trade signal, which is labeled as BUY, SELL, or HOLD for each news item. As a result, trading choices are strongly based on both recent price momentum and simultaneous sentiment analysis.

III. Model Architecture and Training

Our model development is divided into two parts: (1) training and evaluation of traditional machine learning models for sentiment classification, and (2) fine-tuning FinBERT, a transformer-based language model pre-trained on financial corpora.

A. Traditional Machine Learning Models

We first evaluate the performance of four baseline models: Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF). All models are trained using TF-IDF feature vectors extracted from the preprocessed news text. The TF-IDF matrix is limited to a maximum of 5,000 features to ensure computational efficiency. Each model is trained on 80% of the Kaggle dataset, and evaluated on the remaining 20% using accuracy, precision, recall, and F1-score metrics.

Support Vector Machines (SVM) perform better compared to the baseline models with a test accuracy of 80.2% and a macro F1-score of 0.74. Random Forest and Logistic Regression also give good results with test accuracies of 79.1% and 76.1%, respectively. Naive Bayes performs poorly, especially in its capability to identify minority classes.

Figure 1 to 4 show confusion matrices for each model, and Figure 5 shows a comparison of training vs testing accuracy across all models.

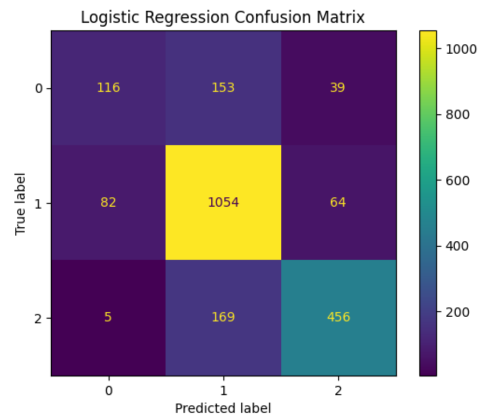

Fig. 1. Logistic Regression Confusion Matrix

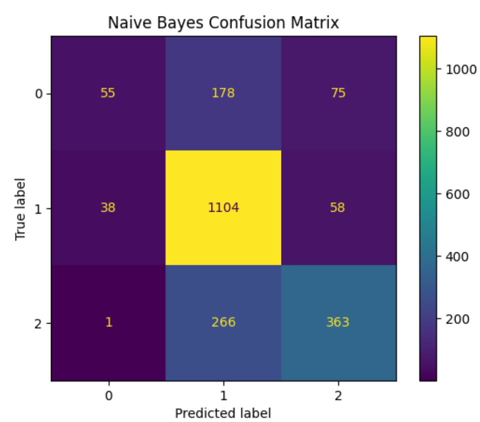

Fig. 2. Naive Bayes Confusion Matrix

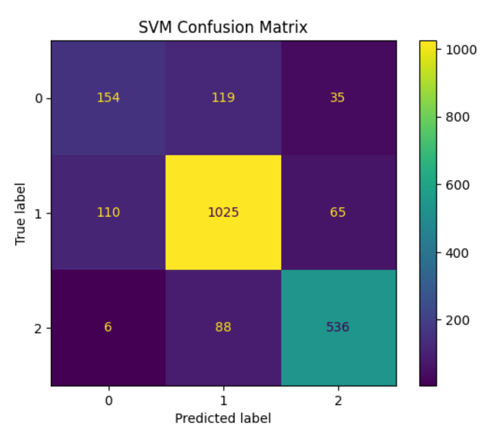

Fig. 3. Support Vector Machine (SVM) Confusion Matrix

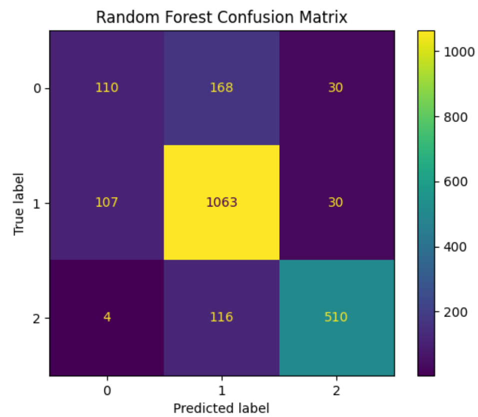

Fig. 4. Random Forest Confusion Matrix

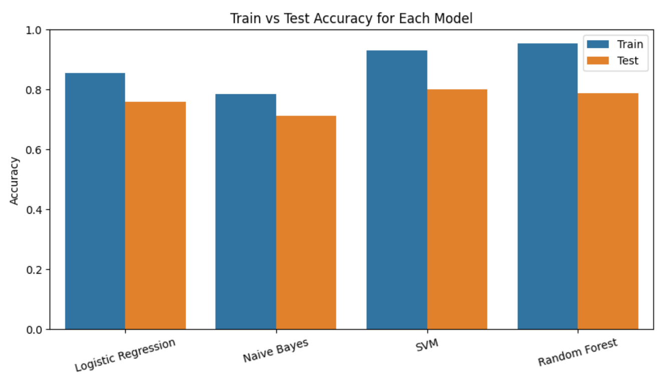

Fig. 5. Training vs Testing Accuracy for Traditional Models

B. FinBERT Fine-Tuning

To improve sentiment classification performance, we use FinBERT— a BERT variant pre-trained specifically on financial text. We fine-tune FinBERT on the same Kaggle dataset using Hugging Face Transformers. The dataset is converted into Hugging Face 'Dataset' objects and tokenized using the pre-trained FinBERT tokenizer. The training process uses the 'Trainer' API with the following configuration:

- Epochs: 10
- Learning Rate: 2×10^{-5}
- Batch Size: 16
- Optimizer: AdamW with weight decay
- Evaluation Strategy: Accuracy and macro F1-score

The classification report is summarized as follows:

- **Accuracy:** 89.0%
- **Macro F1-Score:** 0.85
- **Precision/Recall (positive):** 0.91 / 0.94
- **Precision/Recall (neutral):** 0.95 / 0.87
- **Precision/Recall (negative):** 0.65 / 0.83

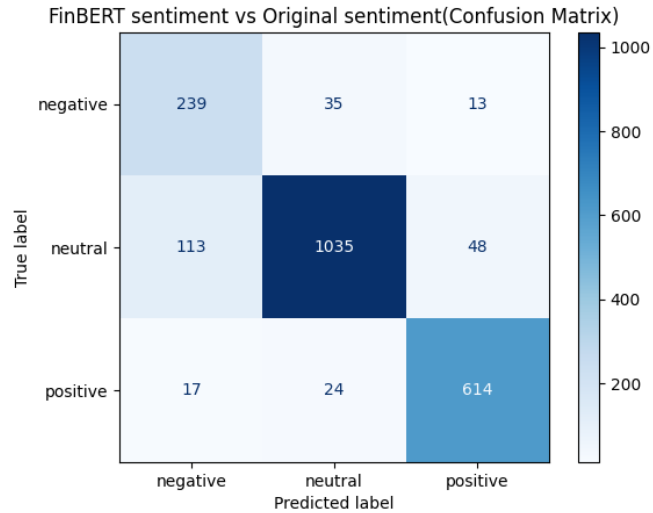

Fig. 6. FinBERT Confusion Matrix on Test Data

These results confirm that FinBERT, with its domain-specific pretraining and fine-tuning, significantly outperforms classical models and is thus selected for the real-world sentiment inference pipeline.

IV. Trading Strategy and Signal Aggregation

To convert sentiment predictions into actionable trading decisions, we design a multi-stage pipeline that integrates natural language processing with technical indicator analysis. This section outlines how we extract company tickers, compute technical indicators, generate aggregated trade signals, and prepare them for execution.

A. News Retrieval and Ticker Extraction

We utilize the NewsAPI to retrieve the latest financial news headlines from a specific date and search term (e.g., "stock market"). Every headline is searched for mentions of particular companies or ticker symbols by utilizing a dictionary mapping common company names to their respective stock tickers.

B. Sentiment Classification with FinBERT

All the news headlines undergo processing by the fine-tuned FinBERT model, which provides both the predicted sentiment label, that is, whether positive, neutral, or negative, and the associated confidence score. The sentiment is then converted to a numerical signal: +1 for `positive`, 0 for `neutral`, and -1 for `negative`.

C. Technical Indicator Computation

Alongside the sentiment scores provided by FinBERT, we also add two commonly utilized momentum-driven technical indicators such as Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) to enhance the decision-making process even more utilizing price-derived signals.

1) Relative Strength Index (RSI): The Relative Strength Index, also known simply as the J. Welles Wilder momentum oscillator, is an indication of the recent price changes' magnitude. This specific gauge helps to point to the overbought or oversold conditions for a stock price.

The RSI value is computed using the formula:

$$RSI = 100 - \left(\frac{100}{1 + RS}\right), \quad \text{where } RS = \frac{\text{Average Gain}}{\text{Average Loss}}$$

Typically, RSI is calculated over a 14-day window. A value above 70 suggests that the stock is overbought (potential SELL signal), whereas a value below 30 suggests oversold conditions (potential BUY signal).

We convert the RSI value into a discrete trading signal:

- $RSI < 30$: BUY (+1)
- $RSI >\ 70$: SELL (-1)
- Otherwise: HOLD (0)

This indicator helps detect when a stock may reverse due to extreme price movements, making it a valuable counterbalance to sentiment.

2) Moving Average Convergence Divergence (MACD): MACD is a trend-following momentum indicator that shows the relationship between two exponential moving averages (EMAs) of a stock's price. It is defined as:

$$\text{MACD Line} = EMA_{12} - EMA_{26}$$

$$\text{Signal Line} = EMA_9(\text{MACD Line})$$

Where:

- EMA_{12} is the 12-day exponential moving average
- EMA_{26} is the 26-day exponential moving average
- EMA_9 is a 9-day EMA of the MACD line itself

A bullish signal is generated when the MACD Line crosses above the Signal Line, and a bearish signal when it crosses below. We discretize MACD as follows:

- $MACD > Signal$: BUY (+1)
- $MACD < Signal$: SELL (-1)
- MACD = Signal: HOLD (0)

MACD captures trend strength and direction, making it complementary to RSI's mean-reversion properties and FinBERT's sentiment trends.

3) Role in Trading Decision: Both RSI and MACD give us an internal view of the market from the perspective of price and momentum that is not dependent on any news. Combining them with FinBERT sentiment signals, we have a strong hybrid decision-making system that utilizes both the external (news sentiment) and the internal (price movement) information to generate trades.

D. Signal Aggregation

We compute a composite score by averaging the sentiment, RSI, and MACD signals for each headline. This final score is then thresholded to determine a discrete trade action:

$$\text{Final Decision} = \begin{cases} \texttt{BUY}, & \text{if score} > 0.33 \\ \texttt{SELL}, & \text{if score} < -0.33 \\ \texttt{HOLD}, & \text{otherwise} \end{cases}$$

Additionally, the most influential component (i.e., sentiment, RSI, or MACD) is identified by comparing absolute magnitudes, and a textual reason is attached to each decision for interpretability.

E. Ticker-Level Aggregation

Multiple news headlines may refer to the same ticker on a single day. We group decisions by `ticker` and `date`, and resolve conflicts as follows:

- If all headlines agree, the majority decision is used.
- Otherwise, we compute the average sentiment score to break ties.

This results in one final decision per ticker per trading day, simplifying the execution logic.

F. Signal Visualization

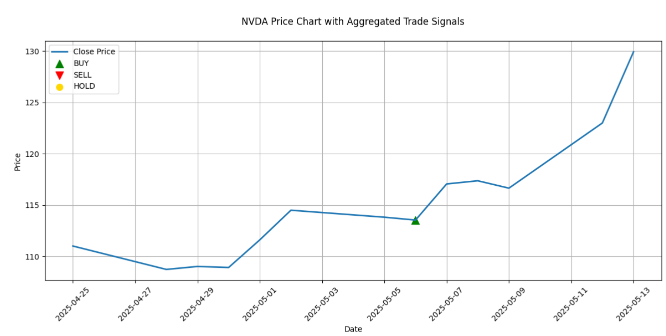

Fig. 7. Aggregated Trade Signals for NVDA

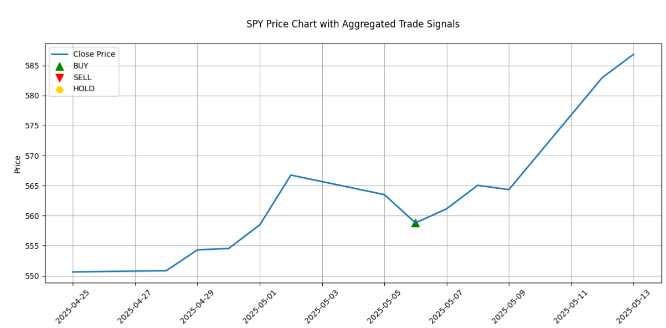

Fig. 8. Aggregated Trade Signals for SPY

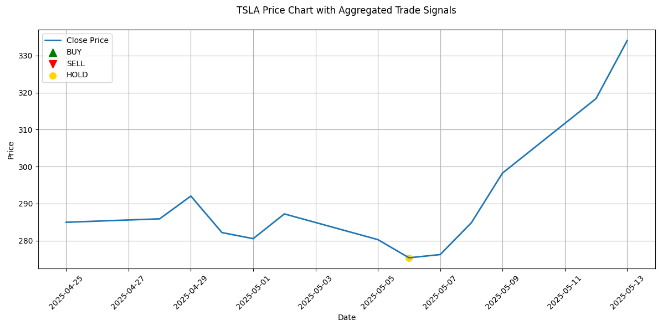

Fig. 9. Aggregated Trade Signals for TSLA

We create signal charts overlaying BUY/SELL/HOLD markers on top of each ticker's closing price trend. Green triangles pointing upwards are BUY signals, red triangles pointing downwards are SELLS, and yellow dots are HOLDS. This graphical illustration is for observing the signal's movement with the market, which is shown in Figure 7 to 8 for a few of the tickers.

V. Backtesting and Performance Evaluation

For the profitability of the proposed sentiment-driven trading strategy to be established, the strategy is backtested for a virtual fund with an initial capital of \$100,000. The strategy makes trades according to composite indications from FinBERT sentiment, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD). A custom backtesting engine applies the following rules:

- **BUY:** The capital is equally divided across all the tickers that are providing a BUY signal. The buys are executed at the next market opening price.
- **SELL:** We close the position whenever a SELL signal is generated or two trading days have passed since the asset was held whichever happens later.
- **HOLD:** There is no action; the asset is retained in the portfolio.

The system handles weekends and missing data by using a 5-day window to select the most appropriate trading date. It keeps dynamic statistics like portfolio value, cash position, trade executions, and stock holdings throughout the simulation.

A. Performance Summary

The trading simulation produced the following outcomes:

- **Initial Capital:** \$100,000
- **Final Portfolio Value:** \$134,168.73
- **Net Profit:** \$34,168.73
- **Return on Investment (ROI):** +34.17%
- **Total Trades Executed:** 2
- **Final Holdings:** 478 shares
- **Last Trade Action:** HOLD on 2025-05-06

B. Interpretation

Although the backtesting window had generated only 2 trades due to the small amount of real data and conservative signal thresholds, the portfolio earned a respectable return. The BUY signal was triggered around open price, validating the composite power of FinBERT sentiment classification along with technical signals. The result affirms the potential of hybrid sentiment-technical models in building robust quantitative trading systems, particularly when deployed in fast-evolving news-driven markets.

VI. Conclusion and Future Work

This paper presents an end-to-end sentiment-driven trading system that integrates machine learning, natural language processing, and technical indicators to generate actionable trading signals. The approach begins by benchmarking traditional classifiers against a fine-tuned FinBERT model using labeled financial news, with FinBERT achieving superior performance. Real-time headlines are then classified using FinBERT, and the resulting sentiment scores are combined with RSI and MACD indicators to form composite BUY, SELL, or HOLD signals. These signals are aggregated and evaluated through a custom backtesting engine that simulates realistic trading conditions. The strategy achieved a +34.17% return on a \$100,000 virtual portfolio, highlighting the effectiveness of combining sentiment analysis with technical momentum indicators in quantitative trading.

A. Future Work

- **Extended Technical Indicators:** Including other indicators such as Bollinger Bands, Average True Range (ATR), and stochastic oscillators can provide more robust trade signals.
- **High-Frequency Signal Generation:** The system currently supports the daily granularity. Enabling intraday or hourly signal generation can facilitate high-frequency trading strategies(Algorithmic trading) and more frequent decision cycles.
- **Multi-Asset Diversification:** Opening the pipeline to support broader asset classes (e.g., ETFs, commodities, crypto) can enhance portfolio diversification.
- **Live Deployment:** Future releases could include the strategy with brokerage APIs for live implementation and risk management in a production environment.
- **Explainable AI:** Increasing transparency and trust by incorporating model explainability techniques (e.g., SHAP, LIME) into sentiment decisions can assist traders and stakeholders.

References

[1] D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models," *arXiv preprint arXiv:2006.08097*, 2019.

[2] J. Welles Wilder, "New Concepts in Technical Trading Systems," Trend Research, 1978.

[3] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 EMNLP: System Demonstrations*, 2020, pp. 38–45.

[4] S. Dubey and M. Agarwal, "An Integrated Model of Sentiment Analysis and Quantitative Index for Financial Forecasting," *Expert Systems with Applications*, vol. 236, 2024, Art. no. 120334. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417424031658

[5] S. B. Shiwakoti, S. B. Shahi, and P. Singh, "Deep Learning Methods for Financial Sentiment Analysis: A Survey," 2024. [Online]. Available: https://www.researchgate.net/publication/375991280