Enhancing Deep Reinforcement Learning for Stock Trading Using Financial News Sentiment and Volatility

Hana Ben Slama - CAI - Technische Hochschule Ingolstadt - hab5510@thi.de

1. Introduction

Financial markets are influenced not only by quantitative indicators such as historical price movements and technical analysis but also by qualitative factors like investor sentiment, breaking news, and economic events. Many traditional algorithmic trading systems disregard this unstructured information, potentially missing critical market-moving signals, particularly during periods of uncertainty. Reinforcement Learning (RL) presents a dynamic alternative for training autonomous agents to make optimal decisions in complex environments like stock markets. However, most RL-based trading systems are trained using only structured inputs such as OHLCV and technical indicators. This study investigates whether enriching these inputs with sentiment analysis — specifically sentiment trends and volatility derived from financial news — can yield improved trading performance. Our primary research question is:

Can the integration of news sentiment and sentiment volatility into RL agent state representations lead to improved profitability and risk-adjusted performance?

To answer this, we implement four experimental setups within the FinRL framework, introducing sentiment data at varying degrees of preprocessing. Performance is compared across three widely used agents: A2C, PPO, and TD3.

2. State of the Art

Multiple studies have shown that investor sentiment, as expressed through financial news and social media, significantly influences market behavior.

- **Tetlock (2007)** [1] found that negative financial news articles were associated with lower stock returns.
- **Bollen et al. (2011)** [2] demonstrated that Twitter mood could predict stock market movements.

The use of NLP for extracting sentiment from financial text has gained traction, with FinBERT: a domain-specific BERT model trained on financial texts, offering better sentiment detection in market-related language. [3] particularly suited for this task as it improves upon generic models by reducing false interpretations of terms like “loss” or “risk.”

In addition to raw sentiment, the concept of sentiment volatility: measured as the rolling standard deviation of daily sentiment scores; used to approximate market tone uncertainty., which can be described as fluctuations in sentiment tone, can serve as an indicator for market uncertainty.

Inspired by behavioural finance and risk modelling practices, this paper explores whether incorporating such fluctuations helps agents adapt more cautiously during turbulent conditions.

Normalization, smoothing and news volume thresholding are established practices in financial signal processing, often used to mitigate noise and overreaction to transient news.

This study adopts those Feature engineering techniques to refine sentiment signals before training. The experiments are conducted using the FinRL library*)* [4], which offers modular RL environments for financial applications. Modifications were made to align sentiment scores temporally with price data and enable state integration.

3. Approach

Four experiments were designed to test the impact of sentiment features on RL agent performance:

- **Experiment 1 – Baseline**: Only technical indicators (OHLCV-based)

- **Experiment 2 – + Raw Sentiment**: Daily FinBERT sentiment scores added

- **Experiment 3 – + Optimized Sentiment**: EMA-smoothed and normalized sentiment

- **Experiment 4 – + Volatility**: Rolling standard deviation of sentiment added to the optimized sentiment.

4. Dataset, Tools, and Techniques

- **Assets**: The experiments focus on 10 widely traded U.S. stocks: Apple (AAPL), Amazon (AMZN), Meta (META), Microsoft (MSFT), Google (GOOG), Tesla (TSLA), Nvidia (NVDA), Netflix (NFLX), JPMorgan Chase (JPM), and Johnson & Johnson (JNJ). These were selected for their liquidity and consistent media coverage.
- **Timeframe**: Jan 2020 to Dec 2023

- **Market Data**: We retrieved historical **OHLCV** (Open, High, Low, Close, Volume) data for 10 U.S. stocks from **Yahoo Finance** [8] covering the period from January 2020 to December 2023. These values were used to compute technical indicators (vix, turbulence..) and as input features for the RL agents.
- **News Data:** We used the **FNSPID dataset** (Financial News Sentiment and Price Impact Dataset), which contains millions of financial news headlines tagged with publication date and stock tickers. [5] This enabled us to align daily news with specific assets over our selected trading period. fnspid dataset scored using FinBERT.
- **Sentiment Processing:** Filtering, 3-day EMA (Exponential Moving Average) smoothing, normalization, and sentiment volatility.
- **Environment**: **FinRL StockTradingEnv** Used as the base environment. Sentiment features (raw, optimized, volatility) were injected into the agent's state representation, expanding the default observation space. The environment simulates daily trading, and uses an **initial portfolio value of** $**1,000,000**
- **Dependencies:** The execution was in Google Colab using the following dependencies:

  Finrl [4]

  pandas==1.5.3 stable-baselines3==2.2.1 gymnasium==0.29.1

  numpy==1.26.4 pandas_market_calendars transformers==4.41.0

  tokenizers==0.19.1 huggingface-hub==0.28.1
- **Training Setup:**
  - **Agents**: A2C, PPO, TD3.
  - **Fixed (default) hyperparameters** and windows across experiments and models.
  - **Training time varied by agent:**
    - TD3: ~20 minutes (due to off-policy structure and twin networks)
    - PPO: Fast and steady, trained in ~2 to 3 minutes
    - A2C: Lightweight and reactive, trained in ~2 minutes

Note: We patched the `StockTradingEnv` to resolve an input shape mismatch (e.g., 131 vs 155 features) when sentiment features were included.

5. Sentiment Feature Engineering

To systematically investigate the impact of sentiment features, we defined four experimental settings with increasing integration complexity:

- **Experiment 1 – Baseline**: Only technical indicators; no sentiment.
- **Experiment 2 – Raw Sentiment**: Direct daily sentiment scores appended to state.
- Experiment 3 – Optimized Sentiment:
    - **News filtering:** keeping only where we have ≥2 headlines/day.
    - **Normalization:** rescaling of sentiment scores to a similar range.
    - **EMA (Exponential Moving Average):** a smoothing technique that gives more weight to recent values, making it more responsive to short-term changes. [9]
      We also ran an auxiliary experiment using a 3-day **rolling average** instead of EMA but found it less responsive and more prone to distortion on sparse-news days. EMA was preferred for its adaptability.
- **Experiment 4 – Optimized Sentiment + Volatility**: Adds sentiment volatility
    - **Sentiment Volatility**: measured as the rolling standard deviation of daily sentiment scores; used to approximate market tone uncertainty. (rolling standard deviation of scores) as a proxy for emotional divergence.

These setups form the basis of our comparative analysis across three RL agents.

6. Experimental Results & Evaluation Metrics

We assessed performance using the following metrics:

- **Cumulative Return**: Measures the percentage increase in portfolio value over time. It reflects how much an agent earns.

  $$\textit{Cumulative Return} = (\textit{Final Portfolio Value} - \textit{Starting Capital}) / \textit{Starting Capital}$$

- **Sharpe Ratio**: Quantifies return per unit of risk; higher values indicate better stability. It reflects reliability and consistency.

  $$\textit{Sharpe Ratio} = \textit{Mean Daily Return} / \textit{Standard Deviation of Daily Return}$$

- **Final Portfolio Value**: Closing capital at the end of the test period.

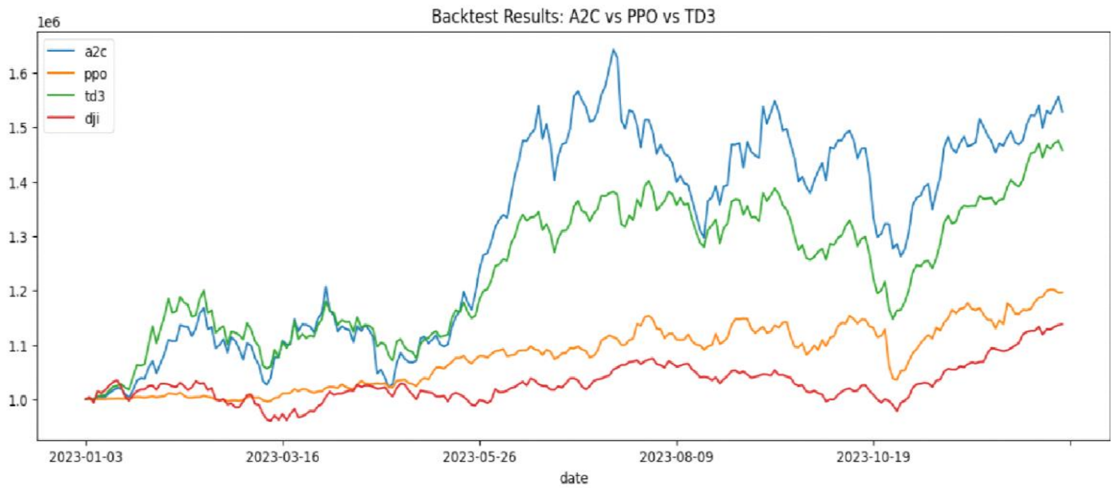
Accuracy, in a trading sense, corresponds to producing consistent, high Sharpe and cumulative profit, not just correct predictions.

Final Portfolio Value:

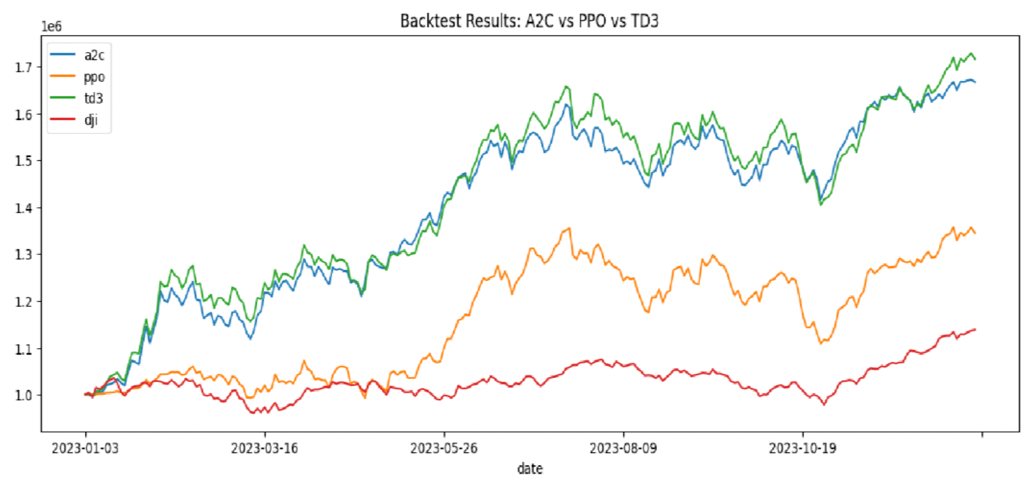
Each model began with an initial capital of **$1,000,000,** and here we are assessing the final portfolio values, as of 2023-12-28, achieved by each model across all four experiments.

Figures 1A–1D show the portfolio evolution across time for each model in each experiment. These plots reveal not only how much each agent earned, but also how steadily the growth occurred, which is a key factor in understanding risk and reward profiles:

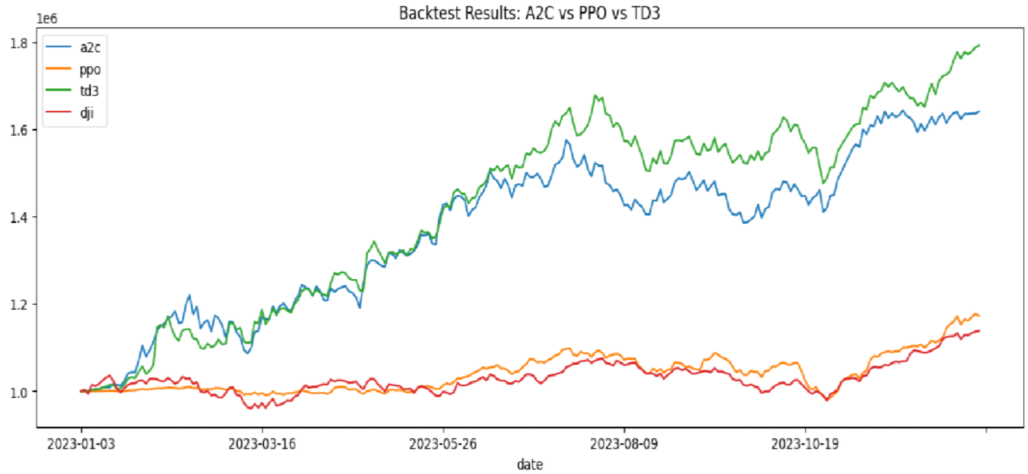
- *Fig 1A. Portfolio Growth (No Sentiment)*



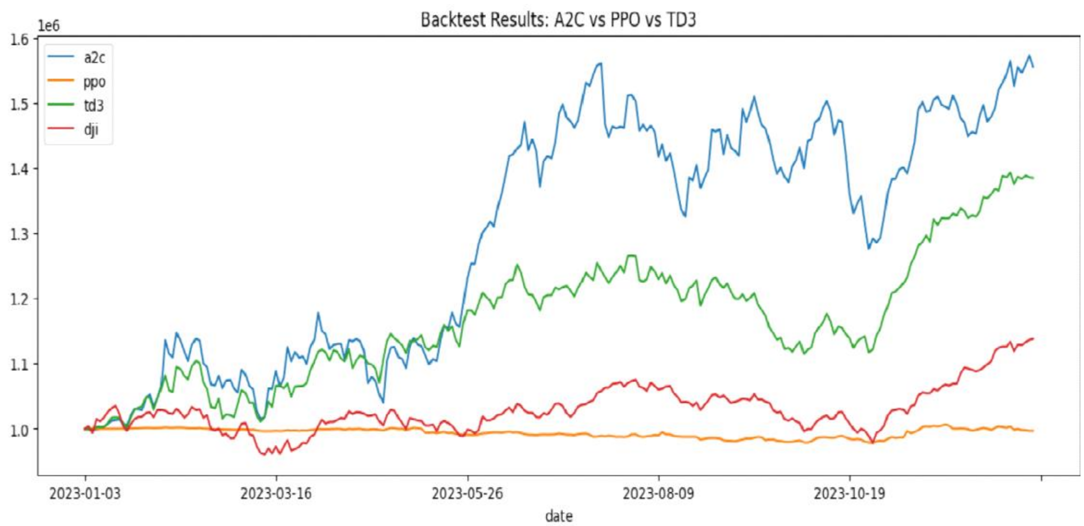
- *Fig 1B. Portfolio Growth (Raw Sentiment)*



- *Fig 1C. Portfolio Growth (Optimized Sentiment):*



- *Fig 1D. Portfolio Growth (Optimized Sentiment + Volatility)*



We summarize the Final Portfolio Value findings below

- *Table 1 – Final Portfolio Values by Experiment and Agent (on 2023-12-28)*

Model	No Sentiment	Raw Sentiment	Optimized Sentiment	Optimized Sentiment + Volatility
A2C	$1,528,590	**$2,040,000**	$1,640,656	$1,554,927
PPO	$1,196,072	$1,344,424	$1,172,948	$996,754
TD3	$1,457,936	$1,730,000	**$1,791,986**	$1,384,668
DJI	**$1,138,028**	$1,138,028	$1,138,028	$1,138,028

Cumulative Return and Sharpe Ratio:

Here we evaluate each model using two key performance metrics: Cumulative Return and Sharpe Ratio. These metrics provide insight into both the profitability and stability of each agent's strategy.

These charts provide a clear snapshot of performance changes across experimental designs.

- *Figure 2 – Bar Chart: Metric Comparison Across Experiments*

We summarize the Cumulative Return and Sharpe Ratio findings below:

- *Table 2 – Cumulative Return (R) & Sharpe Ratio (S)* ***(All Experiments: R/S)***

Agent	No Sentiment	Raw Sentiment	Optimized Sentiment	Optimized Sentiment + Volatility
A2C	0.53 / 1.52	0.67 / 2.38	0.64 / 2.36	0.55 / 1.71
PPO	0.20 / 1.42	0.34 / 1.56	0.17 / 1.69	0.00 / –0.11
TD3	0.46 / 1.69	0.72 / 2.42	**0.79 / 2.82**	0.38 / 1.88
DJI	**0.14 /1.20**	0.14 /1.20	0.14 /1.20	0.14 /1.20

Below we can see which agent performed the best overall compared to the baseline.

- *Table 3 – Percentage of Gain in Cumulative Return, Sharpe Ratio, and Final Value*

Agent	Best Setup	Cumulative Return (% gain)	Sharpe Ratio (%gain)	Final Value ($) (%gain)
A2C	Raw Sentiment	67 (+26%)	**238 (+56%)**	2,040,000(+33%)
PPO	Raw Sentiment	**34 (+50%)**	156 (+10%)	1,344,424(+12%)
TD3	Optimized Sentiment	**79 (+72%)**	**282 (+67%)**	**1,791,986(+23%)**
DJI	N/A	**14 (baseline)**	**120 (baseline)**	**1,138,028 (baseline)**

5. Interpretation and Discussion

The comparative results reveal several key patterns in how sentiment and its preprocessing affect different reinforcement learning agents.

These observations are best understood considering each model's design, policy update strategy, and sensitivity to feature noise or dimensionality.

- *Table 4 – Agents Architecture*

Agent	Key Traits	Best Setup	Strength	Weakness
A2C	On-policy, reactive	Raw Sentiment	Fast adaptation	No memory, vulnerable to noise
PPO	On-policy, stable (clipped)	Raw Sentiment	Training stability	Poor adaptation to uncertainty
TD3	Off-policy + replay + twin Q	Optimized Sentiment	Trend generalization	Struggles with noisy inputs

- **A2C (Advantage Actor-Critic**):

A2C is an on-policy (learns from current policy's data, good for short-term feedback loops) [6] algorithm, meaning it updates its policy using only the most recent trajectory. This makes it highly responsive to recent feedback, a trait that helped it benefit quickly from raw sentiment signals. With the inclusion of raw sentiment, A2C's Sharpe ratio rose from 1.52 to 2.38, reflecting improved stability and profitability.

However, when sentiment volatility was introduced, performance declined. This may be because A2C lacks a mechanism to remember or smooth over noise. Volatility that represents inconsistency or uncertainty in sentiment, more likely to cause A2C to make erratic decisions, mistaking fluctuation for signal. Without a way to adjust risk dynamically, A2C treated this uncertainty as misleading input.

- **PPO (Proximal Policy Optimization)**:

PPO uses a **clipped surrogate objective**, which limits how much the policy can change in each update. [6] This makes training more stable but can also **dampen the model's reactivity** to new or subtle signals, such as sentiment scores.

In our experiments, PPO showed a small improvement with raw sentiment but **worsened** when sentiment was optimized or when volatility was added. PPO's performance collapsed in the final experiment. This suggests PPO could not adapt to the additional complexity or extract meaningful patterns from the noisy or high-dimensional state space. PPO would likely require **re-tuning** or architectural adjustments to handle abstract features like volatility.

- **TD3 (Twin Delayed DDPG):**

TD3 is an off-policy (learns from past experiences stored in a buffer, useful for stable learning over time) [6] algorithm that uses a replay buffer: a memory structure that stores previous experiences and allows off-policy agents like TD3 to train more stably. [FinRL Wiki] and twin Q-networks to stabilize learning. This contributed to its strong performance in our experiments. It performed well with raw sentiment and achieved its best results with optimized sentiment (Sharpe ratio: 2.82, cumulative return: 0.79).

The replay buffer likely enabled TD3 to smooth over fluctuations, learning generalizable trends rather than overreacting to every data point. However, with sentiment volatility added, TD3's performance dropped. Despite its architecture, the added volatility may have introduced unpredictable variation that confused the deterministic actor network. This suggests TD3 is powerful but still relies on relatively stable input distributions to thrive.

- **DJI Baseline (Dow Jones Industrial Average):**

DJI was used as a baseline. It underperformed all agent configurations, except PPO, in the final (volatility) experiment. This confirms that reinforcement learning agents, when supplied with well-engineered sentiment signals, can outperform traditional market indices even without expert trading strategies.

Cross-Agent Takeaways:

- **Sentiment preprocessing (specifically filtering and smoothing)** provided consistent improvements for **TD3** and **A2C**, but not as much for **PPO**. These results suggest that models capable of integrating trends over time benefit from stabilized input, whereas PPO's architecture is less responsive to such refinement.

- **Raw sentiment features** performed better than optimized ones only in **PPO**, indicating that **over-processing** may reduce PPO's ability to detect actionable short-term signals. This highlights how feature transformation must align with the model's learning dynamics.

- **Sentiment volatility** failed to improve performance in any agent and **significantly harmed PPO**. This suggests that the models interpret features deterministically and lack the probabilistic reasoning needed to handle uncertain or conflicting information.

- **More features don't necessarily mean better performance**. The effectiveness of sentiment-based features depends on the compatibility between the **feature's structure** and the model's **capacity to generalize** from it. Simply increasing feature richness does not guarantee a better outcome.

Overall, this confirms that even simple RL agents, when enhanced with structured sentiment input, can outperform traditional index investing.

However, sentiment can be powerful only when presented in a form that the learning model is equipped to handle. Otherwise, it can introduce instability and degrade performance.

7. Conclusion

This study explored how incorporating financial news sentiment and sentiment volatility: measured as the rolling standard deviation of daily sentiment scores.

Our objective was to determine whether structured sentiment input could improve prediction and decision-making, with a **target of achieving at least a 35% increase in model accuracy or profitability.**

The experimental results confirmed that sentiment analysis alone significantly enhances performance, particularly for agents like TD3 and A2C.

- TD3 and A2C both exceeded the performance improvement goal of ≥35%:
- TD3's cumulative return increased from **0.46 to 0.79** about **72% gain**
- A2C's Sharpe ratio rose from **1.52 to 2.36** about **55% improvement**
- **PPO** saw about **50% gain in return**, but only **10% improvement in Sharpe ratio** It became more profitable with raw sentiment, but not significantly more stable.

These results affirm that structured sentiment signals can enhance both profitability and stability in RL-based trading systems.

This underscores the importance of aligning feature complexity with the model's architectural capabilities and learning dynamics.

While the integration of sentiment volatility was theoretically promising, as it aims to reflect market uncertainty and emotional inconsistency, it did not lead to consistent performance gains and, in some cases, reduced agent stability. Its poor performance likely stems from the fact that most RL models treat inputs as deterministic facts, rather than distributions or confidence-weighted signals.

Future research could explore the use of Bayesian reinforcement learning, which explicitly models uncertainty and may offer a better framework for handling sentiment volatility as a probabilistic signal. Additionally, LSTM-based RL agents which can retain temporal information and better interpret sentiment patterns that unfold over time.

References

[1] Tetlock, P. C. (2007). Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *Journal of Finance*, 62(3), 1139–1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x

[2] Bollen, J., Mao, H., & Zeng, X. (2011). *Twitter mood predicts the stock market*. Journal of Computational Science, 2(1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

[3] Huang, B., et al. (2020). “FinBERT: A Pretrained Language Model for Financial Communications.” Hugging Face: https://huggingface.co/yiyanghkust/finbert-tone

[4] FinRL GitHub: https://github.com/AI4Finance-Foundation/FinRL

[5] FNSPID https://huggingface.co/papers/2402.06698

[6] Raffin, A., et al. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. https://stable-baselines3.readthedocs.io/

[7] Ye, H., & Zhang, Z. (2020). “Deep Reinforcement Learning for Automated Stock Trading.”

[8] Yahoo Finance API. Historical Market Data. https://finance.yahoo.com

[9] pandas Documentation – ewm: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.ewm.html