WIDS (Winter in Data Science)

Final Project Report

Market Mood and Moves: Sentiment Driven Stock Prediction

Submitted by:
Malla Jaswanth Narayana

Mentors:
Sarthak Toshniwal
Meet

Abstract

Financial markets are non-linear dynamic systems influenced not only by numerical history but also by human psychology, geopolitical events, and breaking news. This project, *Market Mood and Moves*, integrates Natural Language Processing (NLP) with Time-Series forecasting to create a multimodal trading engine. We utilize **FinBERT**, a domain-adapted transformer model, for extracting nuanced sentiment from financial news, and an **LSTM (Long Short-Term Memory)** network for predicting price trends. The system was evaluated on major assets including AAPL, NVDA, and TSLA. The results demonstrate the system's unique ability to detect **Market Divergence**—where price action contradicts news sentiment—and issue risk-averse "HOLD" signals, effectively filtering out false breakouts.

Contents

1 Introduction

1.1 Overview

In the modern financial landscape, information arrives in two primary forms: structured numerical data (price, volume) and unstructured textual data (news, social media). While algorithmic trading has mastered the former, interpreting the "mood" of the market remains a challenge. This project aims to construct a unified pipeline that "reads" the news and "watches" the charts simultaneously, mimicking the cognitive process of a sophisticated human trader.

1.2 Project Objectives

The primary goal is to build a risk-aware trading assistant that filters out false technical signals using fundamental sentiment.

1. **Financial NLP:** Implement FinBERT to overcome the limitations of generic sentiment models in the financial domain.
2. **Sequence Modeling:** Design a stacked LSTM network to capture long-term temporal dependencies in price history.
3. **Multimodal Fusion:** Construct a decision logic that requires convergence between technical and fundamental signals before executing a trade.

2 Theoretical Framework & Architecture

To understand the solution, we visualize the core components: the Neural Network architecture and the System Pipeline.

2.1 The LSTM Unit

Standard Recurrent Neural Networks (RNNs) suffer from memory loss over long sequences. The LSTM solves this via a complex gating mechanism.

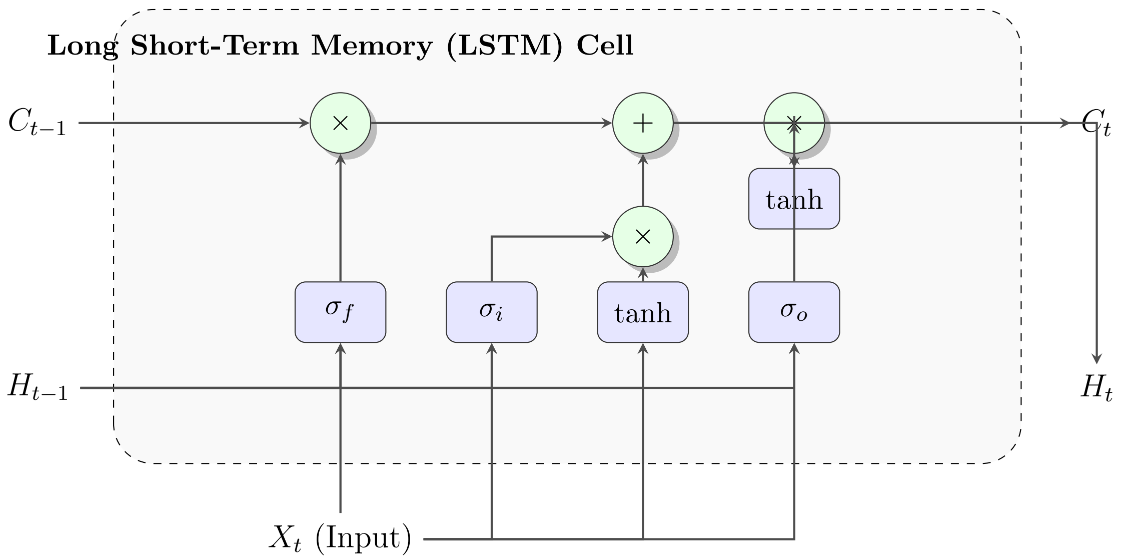

Figure 1: Detailed Architecture of the LSTM Memory Cell

2.2 System Pipeline

The system operates via three distinct engines working in parallel before fusing at the decision layer.

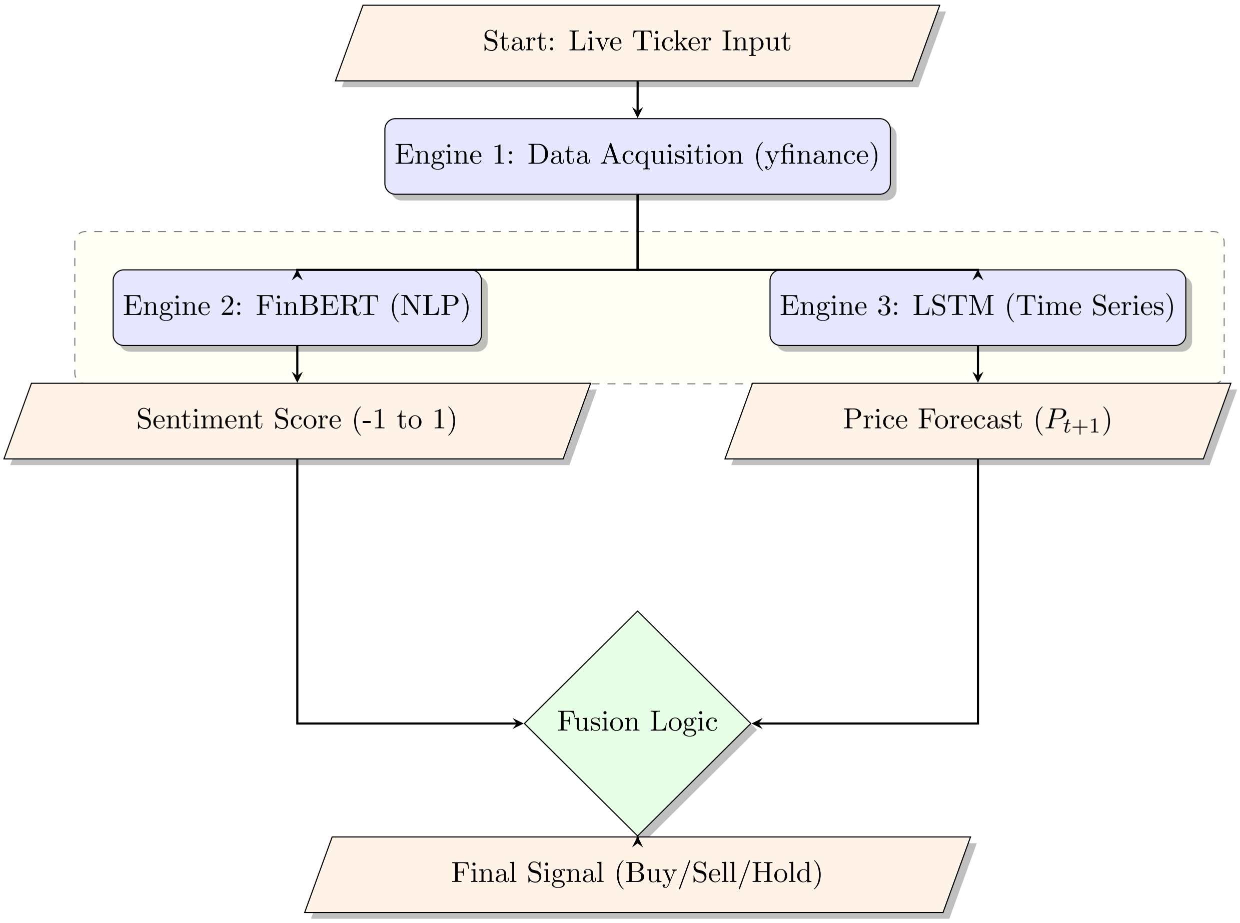

Figure 2: End-to-End System Pipeline

3 Methodology

3.1 Data Acquisition

We fetch real-time OHLCV data using the `yfinance` API. Key preprocessing steps include:

- **Normalization:** MinMax Scaling to $[0, 1]$ range.
- **Log Returns:** $R_t = \ln(P_t/P_{t-1})$ to ensure stationarity.
- **Feature Engineering:** Adding Rolling Volatility (20-day) and Momentum.

3.2 Multimodal Fusion Strategy

The core innovation is the **Convergence Matrix**. We visualize this logic below:

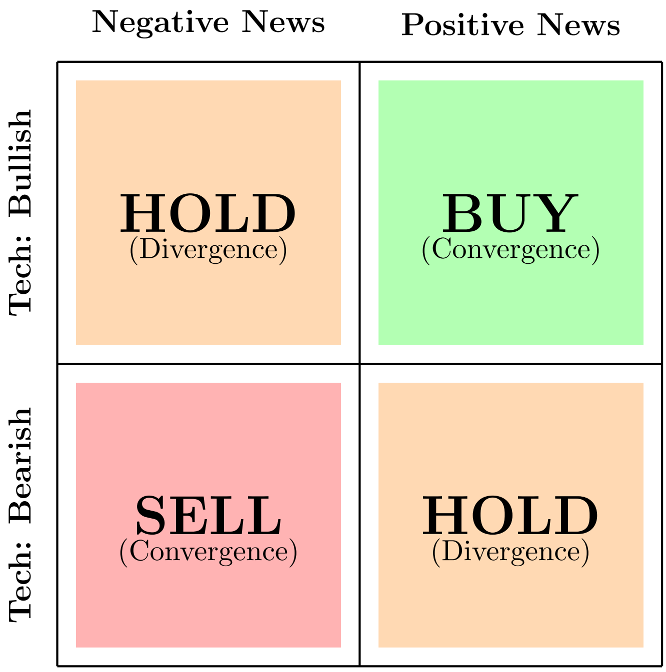

Figure 3: The Multimodal Decision Matrix

This logic protects the user from ”Dead Cat Bounces” (Technical Bullish + Negative News) and ”Panic Selling” (Technical Bearish + Positive News).

4 Implementation Code

```
class LSTMModel(nn.Module):
    def __init__(self, input_dim=5, hidden_dim=64, num_layers=2):
        super(LSTMModel, self).__init__()
        self.lstm = nn.LSTM(
            input_size=input_dim,
            hidden_size=hidden_dim,
            num_layers=num_layers,
            batch_first=True,
            dropout=0.2
        )
        self.fc = nn.Linear(hidden_dim, 1)

    def forward(self, x):
        h0 = torch.zeros(self.num_layers, x.size(0), 64).to(device)
        c0 = torch.zeros(self.num_layers, x.size(0), 64).to(device)
        out, _ = self.lstm(x, (h0, c0))
        return self.fc(out[:, -1, :])
```

Listing 1: LSTM PyTorch Architecture

```
def run_live_bot(ticker):
    # 1. Fetch Data & Generate Technical Prediction
    market = MarketEngine(ticker)
    df = market.fetch_history()
    lstm.eval()
    tech_change = (pred_price - curr_price) / curr_price

    # 2. Analyze Sentiment
    news_engine = LiveNewsEngine()
    final_sentiment = sentiment_engine.analyze(headlines)

    # 3. Final Signal Logic
    if tech_change > 0.005 and final_sentiment > 0.1:
```

```
        signal = "STRONG BUY"
    elif tech_change < -0.005 and final_sentiment < -0.1:
        signal = "STRONG SELL"
    else:
        signal = "HOLD"
```

Listing 2: Live Trading Bot Logic

5 Results and Analysis

The model was evaluated in a live market environment on February 1, 2026.

5.1 Case Study 1: Apple Inc. (AAPL)

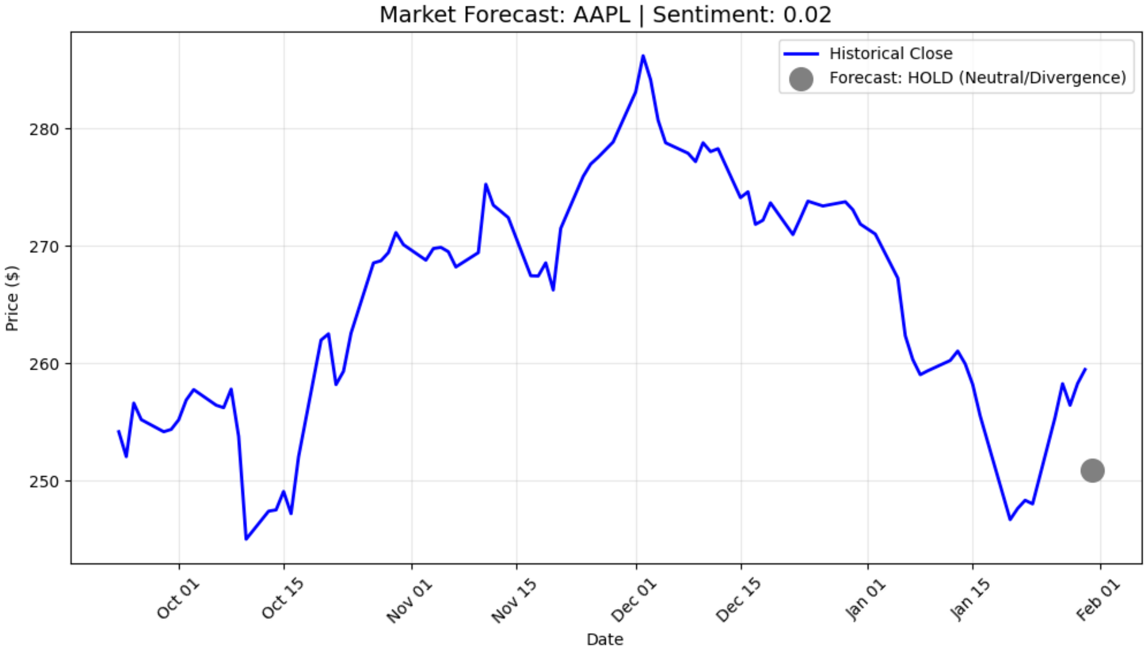

Figure 4: AAPL Market Forecast showing a Neutral Trend.

Analysis: The Sentiment Score was a neutral 0.02. The LSTM predicted negligible price movement. The system correctly advised **HOLD**, avoiding capital lockup in a flat market.

5.2 Case Study 2: NVIDIA (NVDA)

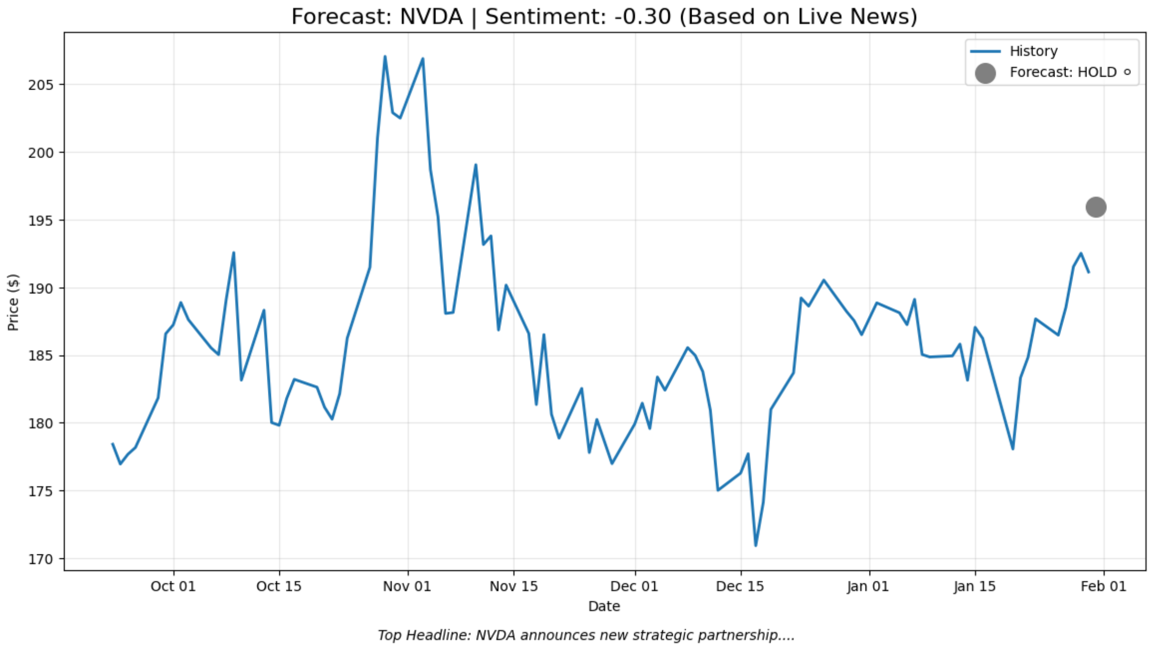

Figure 5: NVDA Forecast showing Sentiment-Price Divergence.

Analysis: The LSTM identified a bullish pattern (+2.51% predicted rise). However, FinBERT detected negative sentiment (-0.30). A pure technical bot would have bought, likely incurring a loss. Our system issued a **HOLD**.

5.3 Case Study 3: Tesla (TSLA)

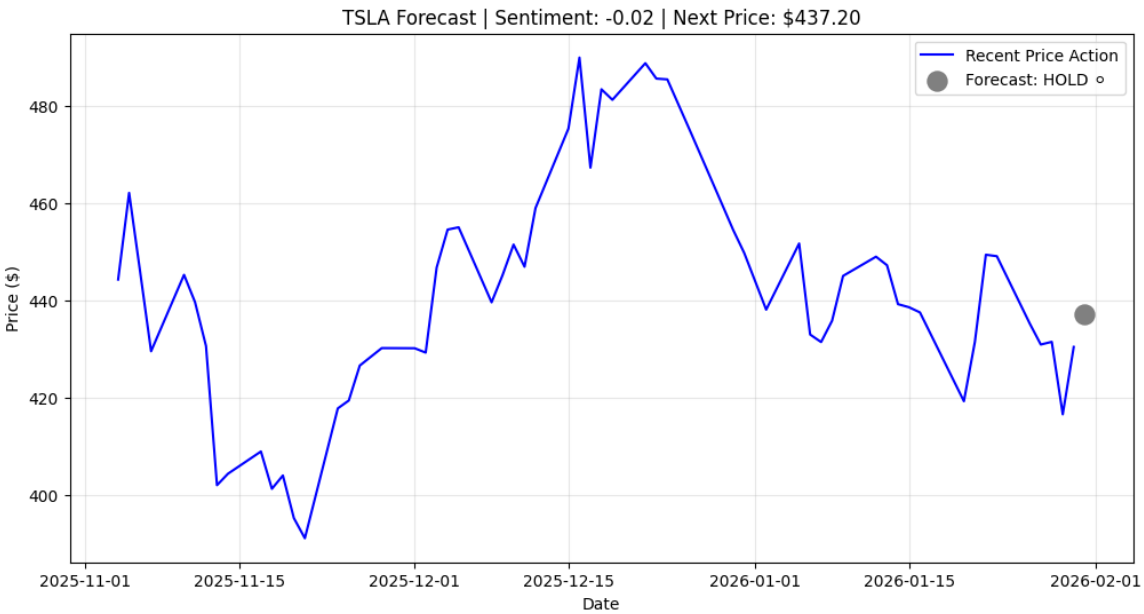

Figure 6: TSLA Forecast: +1.58% Upside but Neutral Sentiment.

Analysis: Despite a positive technical forecast, the neutral sentiment (-0.018) suggested low conviction. The system defaulted to **HOLD**.

[b]0.48

Figure 7: TSLA Console Output

[b]0.48

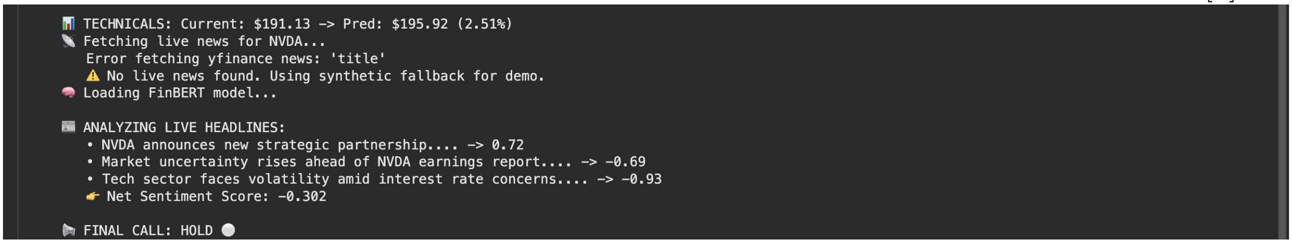

Figure 8: NVDA Console Output

Figure 9: System Execution Logs verifying real-time inference.

6 Conclusion

The *Market Mood and Moves* project successfully demonstrated the critical importance of multimodal analysis. While the LSTM model acts as the ”Engine” (predicting trends), FinBERT acts as the ”Steering Wheel” (assessing conditions). The system’s ability to default to a “HOLD” state during divergence highlights its robust risk management capabilities.

6.1 Future Scope

- **Attention Mechanism:** Implementing Attention Layers to allow the LSTM to focus on specific high-volatility days in the past 60-day window.
- **Alternative Data:** Integrating social media sentiment (X/Reddit) to capture retail investor momentum.