# Sentiment-Driven Stock Prediction Using News and Time-Series Models


## Abstract

Financial markets are complex adaptive systems influenced by both historical price dynamics and continuous information flow. This project investigates whether news sentiment, extracted using a finance-specific transformer model (FinBERT), can enhance traditional time-series prediction when combined with deep learning architectures such as Long Short-Term Memory (LSTM) networks. Over four structured weeks, the project progresses from data understanding and preprocessing, through sentiment modeling and sequence learning, to a full end-to-end predictive and trading pipeline. While the final implementation prioritizes conceptual correctness over optimization, the depth of analysis in Weeks 1–3 establishes a strong theoretical and practical foundation.


## 1. Introduction and Problem Statement (Week 0 Context)

### 1.1 End-to-End Project Overview

Before diving week by week, it is important to clearly explain what was done end-to-end and how each component fits into a single coherent pipeline.

At a high level, the process followed in this project is:

1. Collect historical stock price data
2. Transform raw prices into statistically meaningful signals
3. Collect relevant financial news
4. Convert unstructured text into numerical sentiment scores
5. Align sentiment with market data temporally
6. Construct fixed-length time-series sequences
7. Train an LSTM to predict next-day returns
8. Convert predictions into trading signals

Each step feeds directly into the next. No step is isolated or arbitrary; the design ensures data consistency, temporal causality, and interpretability.

The following sections explain this entire process in depth, both conceptually and practically.

Stock price movements are driven by a mixture of:

- Historical momentum and volatility
- Firm-specific information (earnings, product launches, litigation)
- Macro-economic and geopolitical news

Classical models (ARIMA, GARCH) focus on price history but ignore information causality. Meanwhile, sentiment analysis models capture textual information but lack temporal market context. This project aims to bridge that gap by constructing a multimodal pipeline:

Market Data (Numbers) + News Data (Text) → Unified Predictive Signal

The guiding hypothesis is:

Incorporating domain-specific news sentiment into a sequence model improves directional prediction of stock returns.

# 2. Week 1 – Data Engineering and Market Representation

## 2.1 Financial Time-Series Characteristics

Financial data exhibits several non-ideal properties:

- Non-stationarity
- Volatility clustering
- Heavy-tailed distributions

Directly modeling raw prices leads to unstable learning. Therefore, a transformation into returns and volatility is required.

2.2 Price Data Collection

Daily OHLCV data is sourced using Yahoo Finance. From this, the following features are engineered:

- Log Returns: Ensures approximate stationarity $r_t = \ln(P_t / P_{t-1})$

- Rolling Volatility: Captures local risk regimes $\sigma_t = \text{std}(r_{t-9}, \ldots, r_t)$

These two features together describe direction and uncertainty.

2.3 News Data Collection

For the same ticker, all available news articles are collected using the Yahoo Finance news endpoint. Each article provides:

- Headline text
- Timestamp

Only headlines are used to avoid noise from long-form content and ensure computational feasibility.

2.4 Temporal Alignment

One of the most critical challenges is time alignment:

- Markets operate on trading days
- News can arrive at arbitrary times

To resolve this:

- All news articles are mapped to calendar dates
- Sentiment is aggregated at the daily level
- Missing days are explicitly filled with neutral sentiment

## 2.5 Outcome of Week 1

By the end of Week 1, we obtain:

A clean, aligned dataset where each trading day has:

- Log return

- Rolling volatility
- Placeholder sentiment signal

This step ensures that downstream models are learning from structure, not noise.

# 3. Week 2 – Financial Sentiment Modeling with FinBERT

## 3.1 Motivation for Domain-Specific NLP

Generic sentiment models fail in finance because words like “liability”, “debt”, or “beat” have meanings that differ from everyday language. FinBERT is pretrained on financial filings, analyst reports, and market news, making it suitable for this task.

## 3.2 Sentiment Classification

Each headline is passed through FinBERT, producing probabilities for:

- Positive
- Neutral
- Negative

These are converted into a continuous sentiment score:

$$S = P_{positive} - P_{negative}$$

This formulation:

- Preserves magnitude information
- Avoids brittle categorical labels
- Produces values in the range $[-1, 1]$

## 3.3 Daily Aggregation

Multiple headlines on the same day are aggregated using the mean sentiment score:

$$S_{day} = \frac{1}{N} \sum_{i=1}^{N} S_i$$

This smooths extreme outliers while retaining directional bias.

## 3.4 Interpretation

Sentiment acts as a leading indicator:

- Price reflects realized information
- Sentiment reflects incoming information

This asymmetry is the core reason sentiment can improve prediction.

### 3.5 Outcome of Week 2

A numerically stable, interpretable daily sentiment feature aligned with market data

## 4. Week 3 – Time-Series Learning with LSTM (Core Contribution)

### 4.1 Why Not Simple Regression?

Stock returns violate the i.i.d. assumption. Today's movement depends on:

- Recent momentum
- Volatility regimes
- Delayed reactions to information

This requires sequence-aware models.

### 4.2 Limitations of Vanilla RNNs

Standard RNNs suffer from:

- Vanishing gradients
- Short memory horizons

This makes them unsuitable for financial regimes spanning weeks.

### 4.3 LSTM Architecture

LSTM introduces:

- Cell state for long-term memory
- Forget, input, and output gates

This allows the model to selectively remember patterns such as:

“Sustained negative sentiment during high volatility often precedes drawdowns.”

### 4.4 Multimodal Input Design

Each timestep contains:

- Log return
- Rolling volatility
- Daily sentiment

The input tensor has shape:

(Batch Size, Sequence Length, Feature Dimension)

A sliding window of 30 days is used to balance context depth and sample size.

### 4.5 Learning Objective

The LSTM is trained to predict next-day log return using Mean Squared Error loss. This keeps the task continuous and avoids arbitrary classification thresholds.

### 4.6 Outcome of Week 3

A theoretically grounded, sentiment-aware sequence model capable of learning temporal market patterns

This week represents the intellectual core of the project.

## 5. Week 4 – Integration, Training, and Trading Logic

### 5.1 Model Training

The LSTM is trained on historical sequences using:

- Adam optimizer
- Fixed learning rate
- Limited epochs

The goal is functional correctness rather than convergence optimization.

### 5.2 Prediction and Signal Generation

The final output is a predicted next-day return. A simple decision rule is applied:

- BUY: predicted return > 0 and sentiment > 0
- SELL: predicted return < 0 and sentiment < 0
- HOLD: otherwise

This rule mimics a conservative discretionary trader who acts only when both technical and informational signals agree.

### 5.3 Limitations of Week 4

- No walk-forward validation
- No transaction cost modeling
- No risk or position sizing logic
- Limited evaluation metrics

These limitations are acknowledged and intentionally accepted due to time constraints.

### 5.4 Value of Week 4

Despite its simplicity, Week 4 demonstrates:

End-to-end feasibility and Correct integration of NLP and time-series models and Practical signal extraction

## 6. Results and Qualitative Observations

- Sentiment alone is noisy but directionally informative
- Price-only models lag major information events
- Combined models show better directional consistency

The model is not production-ready, but it validates the project hypothesis.

## 7. Conclusion

This project successfully integrates financial NLP and deep learning-based time-series modeling into a unified predictive framework. The strongest contributions lie in Weeks 1–3, where data engineering, sentiment modeling, and LSTM theory are applied rigorously. Week 4 serves as a proof-of-concept integration rather than a polished trading system.

# 9. References


- Yahoo Finance API
- ProsusAI FinBERT
- Hochreiter & Schmidhuber (1997), Long Short-Term Memory
- WiDS Market Mood & Moves Course Materia