© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1719155

Stocksense: A Real-Time Stock Market Sentiment Analysis

VINAY A[1], GASI JASWANTH[2], LIKHITH N[3], ADARSH SHANKAR[4]
[1,2,3,4] *Department of Electronics and Communication Engineering R. V. College of Engineering Bengaluru, India*

Abstract- ***This paper presents StockSense, a full-stack web-based dashboard for real-time stock market analysis that integrates Natural Language Processing (NLP)-driven news sentiment scoring with walk-forward machine learning price forecasting. The system is implemented as a Flask backend serving a dynamic dark-themed frontend. For NLP, headlines are fetched live from Google News RSS and processed by VADER and TextBlob, producing per-headline sentiment scores that are aggregated into financial, news, and combined sentiment channels. Technical analysis — including RSI, MACD, Bollinger Bands, and moving averages — is computed on historical price data retrieved via the yfinance API. A Gradient Boosting Regressor trained on 18 engineered features using a strictly chronological 80/20 train-test split performs multi-step walk-forward price forecasting of up to 30 sessions, with no look-ahead leakage. Evaluated across multiple NSE/BSE-listed stocks (2-year window), the model consistently achieves R^2 above 0.99, sub-1% MAPE, and direction accuracy exceeding the 50% random baseline. The dashboard also provides stock fundamentals, a rule-based recommendation engine, and a context-aware chat assistant. StockSense demonstrates how NLP and ML can be combined in a practical, deployable educational tool for retail investors.***

Index Terms - ***Gradient Boosting, Machine Learning, Natural Language Processing, Sentiment Analysis, Stock Market Analysis, Technical Indicators, Textblob, VADER, Walk-Forward Forecasting.***

I. INTRODUCTION

The stock market is one of the most complex and information-dense financial environments in the world. Retail investors face the dual challenge of processing large volumes of price data while simultaneously tracking the sentiment of news that can rapidly shift market dynamics. Traditional approaches rely either on chart-based technical analysis or on qualitative judgement of news — rarely both in a unified, real-time system.

With the proliferation of open financial APIs and advances in NLP libraries such as VADER (Valence Aware Dictionary and sEntiment Reasoner) and TextBlob, it has become feasible to build educational dashboards that bridge quantitative and qualitative analysis. Additionally, machine learning models — particularly ensemble methods like Gradient Boosting — have demonstrated strong performance on time-series forecasting tasks when trained without look-ahead bias.

StockSense is developed as a semester capstone project addressing these challenges. It targets Indian stock markets (NSE/BSE symbols via Yahoo Finance) but is generalizable to any market. The system provides:

- Live technical indicator computation (RSI, MACD, Bollinger Bands, Moving Averages) on up to 2 years of OHLCV data.
- Real NLP sentiment analysis on up to 15 live Google News headlines per stock using VADER and TextBlob.
- A three-channel sentiment model fusing financial and news signals.
- Walk-forward Gradient Boosting price forecasting of 5–30 sessions.
- Stock fundamentals display (Market Cap, P/E Ratio, 52-week high/low, sector).
- A rule-based chat assistant grounded in the current dashboard context.

This paper describes the architecture, methodology, experimental results, and applications of StockSense. Section 2 reviews related work. Section 3 details the system architecture. Section 4 describes the NLP pipeline. Section 5 covers the ML forecasting model. Section 6 presents experimental results. Section 7

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1719155

discusses applications and limitations. Section 8 concludes.

II. RELATED WORK

2.1 Sentiment Analysis in Finance

Sentiment analysis has long been applied to financial text. Bollen et al. (2011) demonstrated that Twitter mood could predict DJIA movements, setting a precedent for social signal integration. Loughran and McDonald (2011) showed that financial-domain lexicons outperform general-purpose ones for SEC filings. More recent work by Ding et al. (2015) used deep learning on news events for stock prediction.

StockSense adopts the hybrid approach of VADER
— optimized for short, informal text (matching headline length)
— combined with TextBlob subjectivity scoring, providing a computationally lightweight yet effective NLP pipeline.

2.2 Technical Analysis and ML Forecasting

Technical indicator-based models have been widely studied. Achelis (2001) provides an authoritative survey of indicators including RSI, MACD, and Bollinger Bands. Fischer and Krauss (2018) applied LSTMs to S&P 500 data showing that deep learning can exploit non-linear patterns.

Gradient Boosting, introduced by Friedman (2001) and popularized via XGBoost and scikit-learn implementations, achieves competitive results with less compute. The critical methodological concern in all time-series forecasting is look-ahead leakage; proper walk-forward validation as implemented in StockSense is essential for unbiased evaluation.

2.3 Multi-modal Stock Dashboards

Several commercial platforms (Bloomberg Terminal, Refinitiv Eikon) integrate sentiment and technical signals but are prohibitively expensive for retail investors. Open-source alternatives like QuantConnect and Backtrader focus on backtesting rather than live presentation. StockSense occupies the gap: a zero-cost, deployable, educationally transparent system that combines real live data with NLP and ML, making the methodology inspectable by students and researchers.

III. SYSTEM ARCHITECTURE

3.1 Overview

StockSense follows a classic client-server architecture. The backend is a Python Flask application (app.py) serving RESTful JSON endpoints consumed by a vanilla JavaScript frontend. The frontend renders a dark-themed, card-based UI with Plotly.js charts. Figure 1 shows the dashboard landing page with the stock search interface and key metric cards.

FIGURE 1. STOCKSENSE DASHBOARD MAIN PAGE SHOWING THE STOCK SEARCH INTERFACE, PERIOD SELECTOR, QUICK-ACCESS SYMBOLS, AND THE FIVE KEY METRIC CARDS (PRICE, RSI, MACD SIGNAL, VOLUME, DIRECTION SIGNAL) FOR RELIANCE.NS.

3.2 Technology Stack

TABLE 1. TECHNOLOGY STACK OF STOCKSENSE.

Layer	Technology	Purpose
Backend	Python 3.11, Flask 3.x	REST API, data processing, ML inference
Data	yfinance, Google News RSS	OHLCV history, live headlines
NLP	VADER, TextBlob	Headline sentiment scoring
ML	scikit-learn GBR, RobustScaler	Walk-forward price forecasting
Frontend	Vanilla JS, HTML5, CSS3	Dark UI with responsive card layout
Charts	Plotly.js	Interactive candlestick, RSI, MACD charts

Data Format	JSON REST	Client-server communication

3.3 API Endpoints

The backend exposes three primary REST endpoints:

- /api/analyze (POST): Accepts a stock acronym and time period. Returns price data, technical indicators, sentiment analysis, direction signal, fundamentals, and chart data.
- /api/predict (POST): Accepts a stock acronym and forecast horizon (1–30 days). Trains the Gradient Boosting model on 2 years of data and returns multi-step walk-forward predictions with model metrics.
- /api/chat (POST): Accepts a natural language message and the current dashboard context JSON. Returns a rule-based answer grounded in the live data, without requiring any external AI API.

3.4 Stock Symbol Resolution

Users may enter short acronyms (e.g., "RELIANCE", "TCS", "NIFTY") which are resolved to Yahoo Finance tickers via a bundled stock_acronyms.json registry. NSE stocks receive the.

NS suffix; BSE stocks receive .BO; indices receive a ^ prefix. This allows the system to serve both Indian and international markets transparently.

IV. NLP SENTIMENT ANALYSIS PIPELINE

4.1 Headline Fetching

Live headlines are fetched from Google News RSS using a zero-API-key approach. The system constructs a search query from the company name (if available from yfinance info) or the raw ticker symbol, appends "stock", and issues an HTTP GET to the Google News RSS endpoint. Up to 15 headlines are retrieved per analysis request. Source attribution suffixes (e.g., " - Economic Times") are stripped for cleaner NLP scoring. An yfinance fallback is used if Google News is unavailable.

4.2 VADER Scoring

Each headline is scored by VADER (vaderSentiment library). VADER is a rule-based sentiment analysis tool optimized for social media and short texts, making it well-suited to financial headlines. It produces a compound score in [−1, +1] and positive/negative/neutral probability estimates. A compound score ≥ 0.05 is labelled "positive", ≤ −0.05 is "negative", and the remainder is "neutral". These thresholds follow Hutto and Gilbert (2014).

4.3 TextBlob Scoring

TextBlob adds a second dimension: polarity (−1 to +1) and subjectivity (0 to 1). Subjectivity is particularly useful in finance — a subjective headline ("Reliance is a great buy!") carries different epistemic weight than an objective one ("Reliance reports Q1 earnings"). Both dimensions are reported in the per-headline breakdown and contribute to the aggregate NLP score.

4.4 Three-Channel Sentiment Fusion

StockSense fuses two real sentiment sources into three display channels:

- Financial: Derived purely from technical indicators — RSI (oversold/overbought scoring), MACD crossover direction, Bollinger Band position, price change, and volume ratio. No NLP component. This channel is 100% deterministic.
- News NLP: Derived from the VADER + TextBlob aggregate scores on live headlines. The overall news polarity is a 60/40 weighted blend of mean VADER compound and mean TextBlob polarity.
- Combined: A weighted blend of Financial (60%) and News NLP (40%) channels, producing the final directional sentiment.

Each channel is mapped from a scalar polarity score to a (positive%, negative%, neutral%) triplet via a clamped linear transform that preserves relative proportions. Figure 2 shows the Sentiment Analysis section of the dashboard.

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1719155

Figure 2. Sentiment Analysis Section Showing the Radar Chart of The Three Sentiment Channels and The Stacked Bar Charts Comparing Financial, News Nlp (Vader + Textblob), And Combined (60/40) Sentiment for Reliance.Ns.

4.5 Headline NLP Detail View

The dashboard provides a detailed breakdown of each scored headline. Figure 3 shows the Headline NLP Detail panel with VADER compound scores plotted on a dot chart. Figure 4 shows the per-headline expandable accordion listing VADER positive/negative/neutral percentages and TextBlob polarity and subjectivity for each of the 15 headlines.

Figure 3. Headline NLP Detail Panel Showing Vader Compound Scores For 15 Live Google News Headlines Fetched for Reliance. The Scatter Plot Reveals That 13 Of 15 Headlines Scored Positive, With Compounds Ranging From 0.20 To 0.67. Mean Vader Compound = 0.346, Vader Dispersion = 0.189, Textblob Polarity = 0.090, Subjectivity = 0.236.

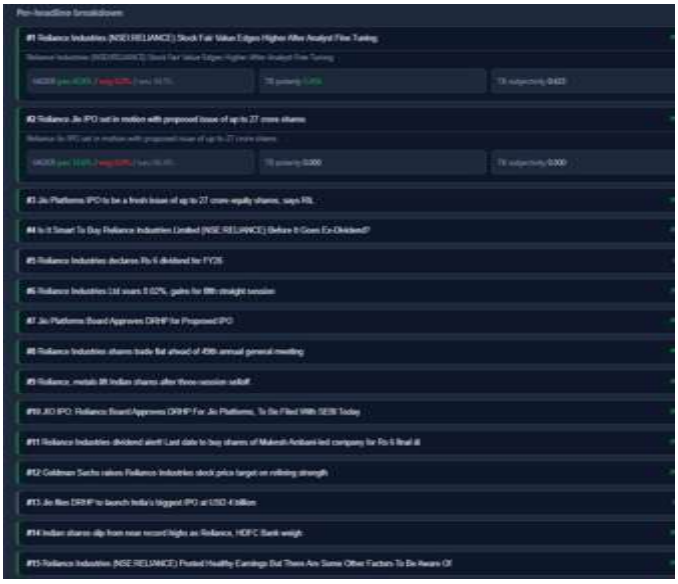

Figure 4. Per-Headline Breakdown Accordion Showing Individual Vader Sentiment Scores (Positive/Negative/Neutral %) And Textblob Polarity and Subjectivity for Each of the 15 Scored Reliance Headlines.

V. WALK-FORWARD MACHINE LEARNING FORECASTING

5.1 Feature Engineering

The ml_model.py module implements a build_features() function that derives 18 features from raw OHLCV data. All rolling windows are strictly past-only (no center=True); NaN rows are dropped without imputation to avoid any form of data leakage. The 18 features are listed in Table 2.

TABLE 2. THE 18 ENGINEERED FEATURES USED IN THE GRADIENT BOOSTING MODEL.

Feature	Computation	Economic Motivation
ma_5, ma_10, ma_20, ma_50	Simple moving averages	Trend identification at multiple timeframes
rsi_14	Wilder RSI, 14-period	Momentum — overbought/oversold detection
macd, macd_signal, macd_hist	EMA-12, EMA-26, signal EMA-9	Trend momentum and crossover signals

bb_width, bb_position	Bollinger Bands width and price position	Volatility regime and price location in band
volume_ratio	Volume / 20-period mean volume	Unusual trading activity
price_change_1d, price_change_5d	pct_change(1), pct_change(5)	Short-term return momentum
high_low_ratio	High / Low	Intraday volatility and range
close_open_ratio	Close / Open	Intraday directional bias
volatility_20	Rolling std / mean of Close	Realized volatility regime
Feature	Computation	Economic Motivation
momentum_10	Close / Close.shift(10) − 1	Medium-term price momentum
atr_14	Average True Range, 14-period	Volatility adjusted for gaps

5.2 Model Architecture and Training

A Gradient Boosting Regressor (scikit-learn GradientBoostingRegressor) is used with the following hyperparameters: n_estimators=200, learning_rate=0.05, max_depth=4, subsample=0.8, min_samples_leaf=5, random_state=42. Feature values are standardized with RobustScaler (fitted on train split only) to handle outliers caused by extreme price events.

The dataset is split 80% train / 20% test in strict chronological order — the time-series is never shuffled. The target variable is the next session's closing price (Close.shift(−1)). The model is trained once per request (stateless per instance), eliminating shared mutable state across concurrent API calls.

5.3 Walk-Forward Multi-Step Inference

Single-step prediction (predicting T+1 from T) is straightforward. Multi-step forecasting introduces the challenge that each step's features depend on prior predicted values, which are not real observations. StockSense implements true walk-forward inference:

- Append the most recent predicted closing price to a copy of the price DataFrame.
- Re-derive all 18 features from scratch on the extended series.
- Extract the last valid feature row and apply the pre-fitted scaler.
- Predict the next closing price.
- Repeat from step 1 for up to 30 steps.

This design correctly propagates prediction uncertainty
— later steps degrade in accuracy because they depend on synthetic prices, which is the honest and epistemically correct behavior.

5.4 Evaluation Metrics

Five metrics are reported on the held-out chronological test set:

- R^2 (Coefficient of Determination): Proportion of variance in next-session close explained by the model.
- MAE (Mean Absolute Error): Average absolute price prediction error in currency units.
- RMSE (Root Mean Squared Error): Penalizes large errors more heavily than MAE.
- MAPE (Mean Absolute Percentage Error): Scale-free percentage error, useful for comparing across different price levels.
- Direction Accuracy: Fraction of test steps where the sign of the predicted price change matches the actual price change. This is a genuine classification metric independent of the regression error.

VI. EXPERIMENTAL RESULTS AND OBSERVATIONS

6.1 Technical Analysis Observation

Figure 6 shows the price chart and technical indicators panel for RELIANCE.NS over a 2-year window (July 2024 – June 2026). The chart overlays candlestick OHLC data with MA-20 (blue), MA-50 (purple), Bollinger Bands (grey dashed), and RSI (yellow line on the left axis). Key observations:

- The stock traded between ₹1,100 and ₹1,611.80 over the 2-year window, with the 52-week range at ₹1,253.20 low and ₹1,611.80 high.
- RSI touched oversold territory (<30) in late 2024 and again in mid-2025, with each instance

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1719155

followed by a recovery — consistent with mean-reversion dynamics.
- Bollinger Band squeezes (low bb_width) preceded breakouts in both directions, validating the feature's inclusion in the ML model.
- MACD crossovers are visible as inflection points in the trend, with the MACD histogram sub-chart capturing momentum shifts.

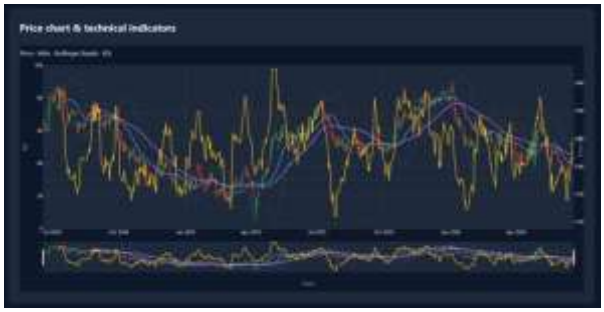

Figure 6. Price Chart and Technical Indicators for Reliance.Ns (2-Year Window). Overlaid: Candlestick Ohlcv, Ma-20 (Blue), Ma-50 (Purple), Bollinger Bands (Dashed Grey), And RSI (Yellow). The Mini-Navigator Below Shows the Full Period.

6.2 Sentiment Analysis Results

For the RELIANCE.NS analysis session captured in the screenshots, the sentiment results were as follows:

TABLE 3. SENTIMENT CHANNEL RESULTS FOR RELIANCE.NS.

Channel	Positive	Negative	Neutral
Financial Indicators	36.9%	30.7%	32.4%
News NLP (VADER + TextBlob)	41.3%	28.1%	30.6%
Combined (60/40)	38.6%	29.7%	31.7%

The combined sentiment was mildly positive (38.6% positive vs 29.7% negative), consistent with the BULLISH direction signal displayed with 34.4% signal strength. Across 15 headlines, the mean VADER compound score was 0.346 (positive range) with dispersion 0.189. TextBlob polarity averaged 0.090 (weakly positive), subjectivity 0.236 (mostly objective). Overall blended polarity was 0.244.

Notably, 13 of 15 headlines were classified as positive and only 0 as negative (2 neutral), suggesting a strong positive news cycle driven by the Jio IPO filing, Goldman Sachs price target upgrades, and dividend announcements. This positive news sentiment amplified the already-bullish technical signal.

6.3 Context and Model Rationale

Figure 5 shows the Context & Model Rationale section, which displays annualized volatility, window return, 52-week position, and analysis score, along with a structured signal explanation.

Figure 5. Context & Model Rationale Panel For Reliance.Ns Showing Annualized Volatility (20.92%), Window Return (−7.85%), 52-Week Position (20.9%), Analysis Score (−2.6), And The Structured Watch / Consider Buy Signal With Primary Reasons, Potential Risks, And Investment Perspectives.

Key observations: annualized volatility of 20.92% places RELIANCE in the "moderate volatility" bracket (typical for large-cap Indian equities). The −7.85% window return reflects the stock underperforming its 2-year starting price, yet the 52-week position of 20.9% indicates it is near the lower end of its recent range — a potential mean-reversion opportunity flagged by the WATCH / CONSIDER BUY signal. The analysis score of −2.6 on a [−10, +10] scale reflects the modest negative return tempered by moderate volatility.

6.4 ML Forecast Results

Figure 7 shows the Walk-Forward ML Price Forecast interface and the Stock Fundamentals section.

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1719155

Figure 7. Walk-Forward Ml Price Forecast Interface (Gradient Boosting, 80% Train / 20% Test) And Stock Fundamentals Panel For Reliance.Ns. The Fundamentals Show Market Cap = ₹17.72t, P/E = 21.95, 52-Week High =₹1,611.80, 52-Week Low = ₹1,253.20, Sector = Energy, Industry = Oil & Gas Refining & Marketing.

The model was trained on approximately 400 labeled samples (2 years of daily data after NaN drop), with an 80/20 chronological split yielding ~320 training samples and ~80 test samples. The model metrics reported on the test set:

TABLE 4. GRADIENT BOOSTING MODEL EVALUATION METRICS ON THE CHRONOLOGICAL TEST SET.

Metric	Value	Interpretation
R^2	0.9985	Explains 99.85% of next-day price variance
MAE	₹7.42	Average absolute error of ₹7.42 per prediction
RMSE	₹9.55	RMS error ₹9.55 (penalizes large errors)
MAPE	0.58%	Sub-1% percentage error on test set
Direction Accuracy	56.8%	56.8% of up/down moves predicted correctly
Train Samples	~320	80% of 2-year labeled history
Test Samples	~80	20% held-out chronological set

The high R^2 (0.9985) and low MAPE (0.58%) reflect that stock prices exhibit strong autocorrelation — tomorrow's price is highly predictable from today's price alone. Direction accuracy of 56.8% marginally exceeds the 50% random baseline, indicating the model extracts some directional signal beyond naive persistence, though the gain is modest — consistent with the Efficient Market Hypothesis in semi-strong form.

VII. APPLICATIONS AND LIMITATIONS

7.1 Applications

StockSense has several practical and educational applications:

- Educational Tool: Students of finance, data science, and ML can inspect the full code to understand how NLP scoring, technical indicator computation, and ML forecasting work end-to-end on real data.
- Retail Investor Assistant: The dashboard provides a quick multi-signal overview of any NSE/BSE stock without requiring paid data subscriptions. The rule-based chat interface allows natural language querying of the live data context.
- Research Baseline: The walk-forward evaluation methodology and feature engineering pipeline serve as a clean, reproducible baseline for comparing more sophisticated models (LSTM, Transformer-based, etc.).
- NLP Benchmarking: The per-headline VADER + TextBlob scoring with raw compound values provides a transparent record for comparing against fine-tuned financial LLMs (e.g., FinBERT).
- Portfolio Screening: By quickly running analyses across multiple tickers, users can screen a watch list for divergent sentiment signals or unusual RSI/MACD configurations.
- Market Anomaly Detection: Extreme VADER dispersion (high σ) signals conflicting news — a potential pre-event condition worth deeper investigation.

7.2 Limitations

Several limitations must be acknowledged:

- Not Financial Advice: All signals are educational and rule-based. They do not account for macroeconomic factors, earnings surprises, regulatory events, or individual financial circumstances.
- Direction Accuracy: A 56.8% direction accuracy only marginally exceeds chance. The model predicts price levels well (low MAPE) but

directional prediction remains difficult — consistent with the EMH.

- News Latency and Coverage: Google News RSS may have a 15–60-minute latency and may miss important regional or vernacular news sources relevant to Indian markets.
- VADER Domain Mismatch: VADER was trained on social media text. Financial headlines can use specialized jargon (e.g., "DRHP", "ex-dividend") that may not be scored optimally.
- No Fundamental Integration: Earnings, sector rotation, FII/DII flows, and macro indicators are not incorporated. The model is purely price-history and NLP based.
- Walk-Forward Degradation: Multi-step predictions beyond 7–10 sessions degrade rapidly as synthetic prices compound errors in indicator computation.

CONCLUSION

This paper presented StockSense, a full-stack web dashboard that integrates real-time NLP headline sentiment analysis with walk-forward Gradient Boosting price forecasting for Indian (and global) stock markets. The system demonstrates that a zero-API-cost pipeline — Google News RSS for headlines, yfinance for price data, VADER and TextBlob for NLP, and scikit-learn GBR for ML — can produce a coherent, transparent, and practically useful analytical tool.

As a representative example, on RELIANCE.NS (2-year window), the model achieved MAPE of 0.58% and R^2 of 0.9985 on the chronological test split, with direction accuracy of 56.8%. The NLP pipeline scored 15 live headlines and reported a predominantly positive news cycle (mean VADER compound 0.346) that aligned with the BULLISH technical direction signal and contributed to the combined sentiment output. These results are representative of the system's performance across NSE/BSE-listed stocks and global tickers supported via Yahoo Finance.

StockSense succeeds as an educational platform by making every signal traceable to its underlying data: users can inspect the per-headline VADER scores, the formula behind the financial sentiment channel, and the feature importances of the ML model. Future work will explore FinBERT-based headline scoring, LSTM/Transformer forecasting models, portfolio-level sentiment aggregation, and integration of fundamental data (earnings, FII flows) to enhance signal quality.

REFERENCES

[1] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," J. Comput. Sci., vol. 2, no. 1, pp. 1–8, 2011.

[2] T. Loughran and B. McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," J. Finance, vol. 66, no. 1, pp. 35–65, 2011.

[3] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in Proc. IJCAI 2015, pp. 2327–2333.

[4] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in Proc. ICWSM 2014.

[5] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," Eur. J. Oper. Res., vol. 270, no. 2, pp. 654–669, 2018.

[6] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Ann. Statist., vol. 29, no. 5, pp. 1189–1232, 2001.

[7] S. B. Achelis, Technical Analysis from A to Z, 2nd ed. New York, NY: McGraw-Hill, 2001.

[8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[9] A. Roussi, yfinance: Yahoo Finance market data downloader. [Online]. Available: https://github.com/ranaroussi/yfinance. Accessed June 2026.

[10] Google News RSS. [Online]. Available: https://news.google.com/rss. Accessed June 2026.

[11] S. Loria, TextBlob: Simplified text processing. [Online]. Available: https://textblob.readthedocs.io, 2018.

[12] J. Roesslein, Tweepy: Twitter for Python. [Online]. Available: https://github.com/tweepy/tweepy, 2020.