StockHark Sentiment Analysis Whitepaper

Executive Summary

StockHark is an advanced sentiment analysis platform designed to extract, analyze, and aggregate financial sentiment from Reddit discussions. Utilizing multi-layered AI models, time-decay weighting, and sophisticated confidence metrics, StockHark enables actionable insights for investors and analysts.

---

1. Introduction

Problem Statement

Monitoring market sentiment across social media platforms is challenging due to volume, noise, and ambiguity. Reddit discussions often contain valuable insights but require careful filtering and weighting.

Objective

StockHark leverages AI-powered sentiment detection to provide precise, timely, and confidence-weighted sentiment scores for stocks discussed in financial subreddits.

---

2. System Architecture

5-Stage Pipeline

1. **Data Collection**: Reddit posts fetched every 30 minutes with stock validation.
2. **FinBERT Analysis**: Financial transformer models classify sentiment.
3. **Time Decay Weighting**: Recent posts carry more influence.
4. **Source & Volume Weighting**: Reliability and post counts adjust influence.
5. **Dual Sentiment Output**: Raw and aggregated sentiment scores.

Diagram

```
[Data Collection] -> [FinBERT Analysis] -> [Time Decay] -> [Source/Volume Weighting] -> [Dual Sentiment Output]
```

---

3. Methodology

Per-Mention Scoring

- **FinBERT**: Outputs label and confidence; POSITIVE -> +confidence, NEGATIVE -> -confidence, NEUTRAL -> 0.
- **Rule-Based**: Financial lexicon boosts scores; multi-word phrases add 2.0; intensifiers multiply the score.
- **Clipping**: All raw scores are clamped to [-1, 1].

Time Decay

Formula:

```
w_t = exp(-λ × Δt_hours)
```

- λ = 0.1 (typical)
- Example: 24h old post → weight ≈ 0.091

Source Weights

- reddit_wsb: 0.8, reddit_other: 0.6, twitter: 0.7, news: 1.0

Symbol Heuristics

- Ambiguous symbols penalized (0 < weight ≤ 1)
- Example: FREE → 0.05, AMD → 1.0

Post-Count Weight

```
post_count_weight = 1.0 + min(max_bonus, log(unique_post_count) × post_count_multiplier)
```

- Encourages aggregation across multiple mentions

Aggregation Formula

```
weighted_avg = Σ_i(raw_sentiment_i × total_weight_i) / Σ_i(total_weight_i)
final_sentiment = clamp(weighted_avg, -1, 1)
```

Example

Mentions: +0.9, +0.2, -0.6 with weights 1.0637, 0.6526, 0.1206 → Final sentiment = +0.553

---

4. Confidence Metrics

Components: - **Weight Confidence**: Derived from total weight - **Consensus Confidence**: Inversely proportional to standard deviation of raw scores - **Sample Confidence**: Function of unique post count

Combined to yield final confidence score [0,1]

---

5. Output Interpretation

5-Tier Sentiment Scale

- **Strong Bearish**: -1.0 to -0.3
- **Weak Bearish**: -0.3 to -0.1
- **Neutral**: -0.1 to +0.1
- **Weak Bullish**: +0.1 to +0.3
- **Strong Bullish**: +0.3 to +1.0

Mapping

- Sentiment + confidence determines dashboard ranking, alerting, and labels

---

6. Data Quality & Filtering

Bot Filtering

• Posts are checked for bot-like characteristics before analysis:
  – Deleted or missing author
  – Very young accounts (configurable minimum age)
  – Low karma (configurable minimum)
  – Bot-like usernames (“bot”, “auto”, etc.)
  – Suspicious posting rates (max posts/hour via Redis)
• If a post matches these heuristics, it is skipped and logged as a probable bot post.
• Feature toggles and thresholds are configurable via environment variables.

Duplicate & Near-Duplicate Detection

• Exact duplicates are detected using SHA256 hashes stored in Redis. If a post’s hash is already present, it is skipped before analysis.
• Near-duplicates are detected using SimHash and Hamming distance checks. Posts with similar content are skipped if they fall within the configured threshold.
• Database-level checks ensure no duplicate (symbol, post_url) entries are inserted, even if upstream filters are disabled.
• All skips are logged for observability and tuning.

These steps reduce noise, save compute, and ensure only unique, high-quality posts contribute to sentiment metrics.

7. Technical Architecture

- **AI/ML Stack**: FinBERT (primary/fallback), spaCy NER
- **Data & Storage**: SQLite, 4,278+ validated symbols, 30-min collection cycles
- **Processing Parameters**: λ=0.1, token limit 512, volume multiplier 0.2
- **Reliability & Performance**: <50ms per analysis, 99.9% uptime

---

8. Competitive Advantages

- Advanced AI Integration: Multi-model FinBERT + spaCy NER
- Temporal Intelligence: Exponential decay weighting
- Dual Sentiment Metrics: Raw + aggregated
- Production-Grade Performance: Real-time, <50ms analysis
- Robust Validation: Hybrid AI + pattern matching
- Mathematical Rigor: Normalized, confidence-scored aggregation

---

9. Disclaimer

This sentiment analysis is informational only and not financial advice. Investment decisions should be made with qualified professionals. Past performance does not guarantee future results.

Appendix

Constant	Value	Description
λ (time decay)	0.1	Controls exponential decay of mention influence
Source weights	{wsb: 0.8, twitter: 0.7, news: 1.0}	Reliability multipliers
Sentiment clamp	[-1, +1]	Hard limits on aggregated score