Real-Time Reddit Sentiment Analysis for Stock Movement Prediction

Colin Kirby

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Stock market prediction has traditionally relied on historical price patterns and financial reports; however, in recent years, alternative data sources such as social media have demonstrated significant potential for real-time insight. In this project, we present RTSA (Real-Time Sentiment Analysis), a modular pipeline designed to collect, process, and analyze Reddit discussions to generate short-term stock movement predictions. Our system integrates Reddit post scraping, financial sentiment scoring, trending ticker detection, stock price collection, feature engineering, model training, and actionable forecasting. Early experiments revealed challenges such as ticker-word ambiguity, dataset sparsity, and feature alignment across data sources. To address these, we introduced context-aware filtering, confidence thresholds, and dynamic feature handling strategies to enhance data quality and model reliability. Although this version of RTSA remains in early development, it successfully produces coherent stock movement predictions, achieving moderate class balance and demonstrating consistent feature extraction for multiple trending equities. Future iterations of the pipeline aim to expand coverage across additional platforms such as Twitter, incorporate cryptocurrency markets, and explore ensemble modeling techniques to further strengthen predictive capabilities. This paper presents our methodology, results, key challenges, and future directions for developing a scalable real-time sentiment-driven stock prediction framework.**

***Index Terms* — Stock market prediction, Reddit sentiment analysis, natural language processing, financial forecasting, machine learning, feature engineering, social media mining, real-time systems, time series forecasting, FinBERT.**

I. Introduction

The stock market is influenced not only by fundamental financial metrics and economic indicators but also increasingly by public sentiment. With the rise of online forums such as Reddit's r/WallStreetBets and r/Investing, collective opinions and viral narratives can drive significant short-term price movements. Understanding and harnessing this alternative data source presents both a technical challenge and a new opportunity for predictive modeling.

Traditional stock prediction systems largely depend on structured data such as historical price trends, company earnings reports, and economic metrics. However, these methods can lag behind real-world events, missing the early signals often visible in social media chatter. Prior studies have shown a correlation between social media sentiment and market trends, motivating the need for real-time, scalable pipelines that can extract, process, and operationalize such information.

In this project, we introduce RTSA (Real-Time Sentiment Analysis), a modular data pipeline built to predict short-term stock movements using Reddit-based sentiment and technical stock indicators. RTSA addresses several core challenges in this domain, including the accurate identification and extraction of meaningful stock discussions from noisy Reddit posts, the precise measurement of financial sentiment within complex textual data, and the engineering of consistent and predictive feature sets that combine social signals with traditional market data.

Our Version 1 pipeline implements a full-stack solution, encompassing Reddit data collection, natural language processing (NLP)-driven sentiment analysis, trending ticker detection, stock price tracking, feature generation, model training, and prediction delivery. Each component has been designed to interact seamlessly, allowing the system to process raw social media posts and deliver stock movement predictions in a streamlined, automated manner.

While RTSA remains in its early stages, the system demonstrates the feasibility of integrating social sentiment with traditional stock data to produce coherent and actionable predictive outputs. This paper describes the methodology behind RTSA, the challenges encountered during development, and the preliminary experimental outcomes of Version 1.

II. Related works

Over the past decade, stock market prediction has evolved from traditional technical analysis toward more complex data-driven approaches that leverage both structured financial data and unstructured textual sentiment. The increasing availability of social media content has driven a wave of studies investigating the role of public sentiment in influencing stock prices, particularly in the short term. Platforms like Twitter, StockTwits, and Reddit have become central to this research due to their high volume of investor opinions,

timely reactions to news, and community-driven forecasting.

Several works have explored sentiment extraction and its predictive power. Tsui (2016) examined StockTwits messages and their relationship with stock performance, concluding that bullish sentiment correlates with short-term price anomalies. While insightful, Tsui's study highlighted limitations in overfitting and selection bias, suggesting that sentiment alone is insufficient for robust prediction. Similarly, Mao et al. (2015) quantified the effect of online bullishness across financial markets, finding that market sentiment often acts as a leading indicator. However, they also noted a lack of contextual precision and challenges with sarcasm or informal language, which degrade model reliability.

More recently, Awan et al. (2021) integrated sentiment from Reddit, Twitter, and Yahoo Finance with financial indicators to predict stock price direction. Their study demonstrated the potential for combining textual and numerical features, particularly when using ensemble learning methods like Random Forests or recurrent neural networks like LSTMs. Nevertheless, scalability and data processing latency remained significant barriers, particularly in systems constrained to batch-mode workflows. The research emphasized that while social sentiment contributes to forecasting accuracy, the inconsistency of language, misinformation, and high data variability introduce noise that current models struggle to filter.

NLP advancements have somewhat addressed these challenges. Transformer-based models, notably FinBERT, have improved financial text classification by leveraging domain-specific pretraining. FinBERT achieves better context understanding compared to earlier models by interpreting nuanced language in financial discourse. Despite these improvements, most implementations are still tied to static or periodic data ingestion, limiting their utility in fast-paced trading environments where real-time decisions are critical.

These works collectively identify a need for solutions that not only capture sentiment accurately but also deliver predictions with minimal delay. Many prior systems rely on coarse-grained sentiment metrics and lagging price data, often without engineering unified feature spaces that account for both social and financial signals. Furthermore, existing approaches typically focus on single-source data (e.g., Twitter-only sentiment), which narrows the model's generalization capability and amplifies bias.

This study builds upon those foundations by proposing a unified real-time pipeline that combines advanced NLP sentiment scoring with engineered technical indicators. Rather than relying solely on post volume or basic polarity scores, our system extracts richer sentiment features and synchronizes them with high-resolution stock data, addressing the limitations in both noise filtering and feature alignment.

While recent research has made significant strides in combining sentiment and market data, many proposed solutions remain constrained by batch-oriented designs, static modeling approaches, and loosely aligned feature spaces. These limitations reduce their applicability to real-time trading environments, where both freshness and contextual precision are paramount.

Figure 1 illustrates this contrast by comparing traditional sentiment prediction pipelines with our proposed RTSA system. As shown, RTSA addresses the shortcomings of prior systems through a fully modular, real-time architecture that leverages financial-context NLP (FinBERT), trending ticker detection, dual-domain feature engineering, and per-ticker machine learning models. This design enables dynamic, explainable, and timely predictions in fast-moving financial markets.

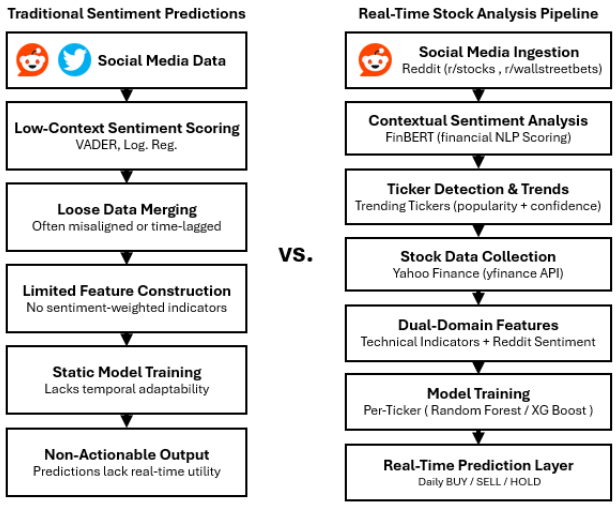

Fig. 1. *Comparison of traditional sentiment prediction and RTSA. RTSA integrates contextual sentiment, dual-domain features, and per-ticker models for real-time forecasting.*

III. METHODOLOGY

A. Data Collection

To initiate the pipeline, Reddit data is collected from a curated set of finance- and market-focused subreddits including r/wallstreetbets, r/stocks, r/investing, and r/StockMarket. These communities were selected due to their high levels of stock-specific discussion, user-driven market speculation, and historical influence on short-term price movements, particularly during events such as the GameStop short squeeze. Posts are retrieved using the PRAW API wrapper for Reddit, with OAuth2

authentication and refresh tokens to ensure secure and persistent access. The collector systematically queries each subreddit across three sort modes which are hot, new, and top, these serve in helping to maximize diversity of post types and capture both viral discussions and emerging trends. A configurable time filter, dynamically determined based on the lookback window, ensures that collected posts fall within a specified date range (e.g., 7 to 30 days prior), maintaining temporal relevance to market conditions.

Each retrieved post includes detailed metadata such as the title, body text, upvotes, number of comments, upvote ratio, author, and creation timestamp, alongside retrieval of the top five comments for each post. Capturing comment data provides valuable secondary sentiment signals and enhances the system's ability to detect nuanced community reactions beyond the original post. However, challenges such as post deletions, comment removals, and Reddit API rate limitations occasionally impact retrieval completeness, necessitating retry mechanisms and error logging within the collector module.

The system proactively skips outdated posts falling outside the lookback window and filters duplicates by Reddit post ID before concatenating results into a unified dataset. Given the varying quality of Reddit posts from in-depth analysis to meme content, the pipeline maintains flexibility to incorporate additional filtering layers in the future, such as minimum post length thresholds or keyword density checks.

To support downstream processing, reproducibility, and auditability, all collected posts are saved as timestamped CSV files under a structured directory hierarchy organized by collection date. Key statistics including post counts per subreddit, effective date ranges, and duplicate removal rates are printed to the terminal and logged for reference. Additionally, the collection process is fully configurable via command-line arguments or programmatic parameters (e.g., subreddit list, post limit, lookback days), enabling easy scalability across different data volumes, subreddit scopes, or experimental designs. This modular setup ensures that the data ingestion component can be extended or adapted as future project iterations expand to cover more platforms or alternative sentiment sources.

B. Sentiment Processing & Ticker Validation

Once Reddit data is collected, each post undergoes a multi-stage sentiment and ticker processing pipeline. Posts are first cleaned and preprocessed to remove URLs, extraneous symbols, and noise. The system then extracts potential stock tickers using regular expressions for both $TICKER formats and standalone capitalized words, with additional context validation to filter out false positives, such as tickers overlapping with common English words.

Sentiment scoring is performed using a domain-specific transformer model, FinBERT, optimized with CUDA acceleration and FP16 precision when available. Each post's sentiment is computed using a context window surrounding the ticker mention to ensure relevance. FinBERT scores are cached and batched for performance. In parallel, traditional sentiment tools such as VADER and TextBlob are applied to generate a blended sentiment metric, enabling ensemble scoring across methods.

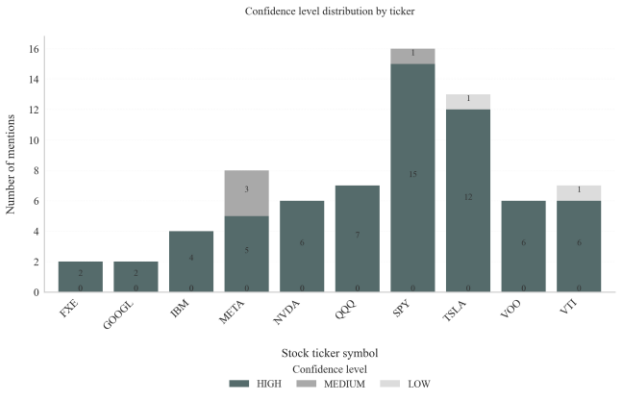

Fig. 2. *Confidence classification output from the Reddit sentiment pipeline. RTSA integrates contextual sentiment scoring, financial entity linking, and confidence stratification to improve per-ticker signal quality.*

To enhance reliability, the system incorporates an EntityLinker module that verifies whether a post's context aligns with company-specific or financial terminology. Posts are classified into confidence tiers based on FinBERT score, keyword density, dollar-symbol usage, and entity validation. Tickers with ambiguous names (e.g., COIN, GOLD, CASH) are subject to stricter validation rules.

For each validated ticker, the system aggregates engagement metrics (e.g., score, comments), sentiment polarity, and time-series statistics. Finalized outputs include daily summaries and detailed sentiment files per ticker, which form the basis for downstream feature engineering and predictive modeling.

C. Trending Ticker Detection & Engagement Analysis

After sentiment scores and confidence levels are assigned to each post, the system shifts focus toward identifying which stock tickers are generating substantial attention. This phase begins by filtering for posts marked as relevant, those with at least one validated ticker mention and a minimum confidence score threshold. The system iterates through these posts to compute mention frequency, aggregate engagement (upvotes, comments),

and confidence-weighted scores for each ticker. These scores reflect not just the volume of discussion, but also the strength and quality of each mention.

To improve signal quality, the pipeline employs several validation layers to exclude noisy or non-financial mentions. A curated list of ambiguous terms (e.g., “OPEN,” “REAL,” “LIVE”) helps prevent false positives, while a cache of verified stock symbols from NASDAQ and NYSE ensures only legitimate tickers are considered. For each ticker, the system tracks multiple attributes including total mentions, average confidence score, cumulative engagement, and sample contexts in which the ticker appears. Tickers without associated sentiment files or those falling below a threshold of two mentions are automatically pruned to maintain reliability.

Using these enriched signals, the pipeline ranks and scores tickers based on relevance and community interest. Each ticker is evaluated for both overall engagement and subreddit-specific distribution, enabling fine-grained insights into which online communities are fueling particular trends. This community-driven segmentation supports the early identification of emerging patterns, such as tickers gaining traction in niche subreddits before expanding to broader forums.

To visualize how attention is distributed across platforms, the heatmap in Fig. 3 displays ticker mentions segmented by subreddit. This representation highlights not only which tickers are most discussed, but also where those conversations are taking place—shedding light on the dynamics of Reddit-driven financial discourse.

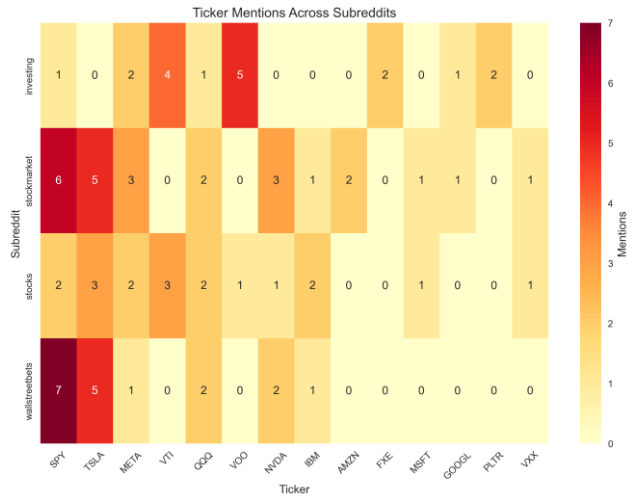

Fig. 3. *Ticker mentions frequencies across subreddits. Higher values indicate greater community attention per ticker in each forum.*

D. Financial Data Acquisition

To complement Reddit-based sentiment data with market-grounded indicators, the system integrates historical stock data using the Yahoo Finance API via the yfinance Python package. This ensures that all sentiment analysis is contextually anchored to real price movements and trading volumes. For each trending ticker identified in the earlier stages, the pipeline retrieves OHLCV data, Open, High, Low, Close, and Volume, over a configurable date range, typically spanning the previous 60 to 180 days.

The data is fetched through the StockDataCollector class, which performs symbol-level requests with retry mechanisms, gap detection, and built-in data validation. Missing or inconsistent values are forward- and backward-filled for continuity, and metadata for each ticker’s dataset is stored alongside the raw files. The collector also calculates basic technical metrics such as daily returns, high-low spreads, previous-close deltas, and RSI (Relative Strength Index), enriching the data for downstream modeling.

Stock data is saved under a unified directory structure for repeatability, version control, and integration with subsequent modules like the feature builder and model trainer. Collection logs include symbol coverage, gaps in trading days, and indicators of data sufficiency (e.g., minimum row count thresholds). This tight coupling of social sentiment and financial data allows the system to generate a dual-domain feature set that aligns both market signals and community dynamics.

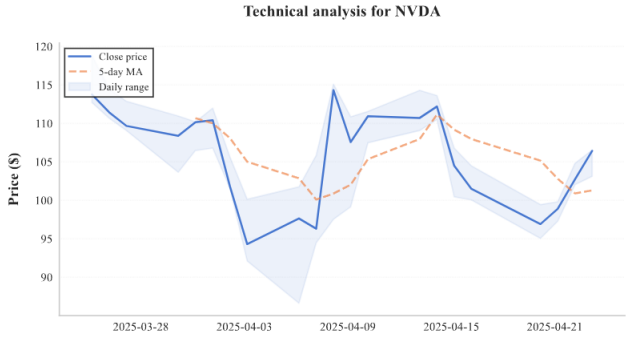

Fig. 4. *Technical price trend analysis for NVDA showing daily closing price, 5-day moving average, and intraday price range over a 30-day period.*

E. Feature Engineering & Model Input Construction

To enable predictive modeling, the system transforms Reddit sentiment and financial price data into structured feature sets tailored to each stock ticker. This transformation is orchestrated through a centralized

FeatureBuilder module that combines multiple data sources, applies temporal alignment, and outputs clean, standardized datasets ready for machine learning tasks.

The process begins by loading four key datasets per ticker: historical stock price data, Reddit-derived daily sentiment summaries, ticker-level analysis outputs, and optional per-ticker sentiment confidence metrics. Each source undergoes cleaning and standardization, with timestamps normalized to timezone-naive formats and floored to daily granularity to ensure consistency. Missing or misaligned timestamps—arising from trading holidays, missing Reddit data, or delayed post activity—are resolved via forward- and backward-filling methods, while non-overlapping dates across inputs are systematically discarded. This careful alignment ensures that each data row accurately represents a unified snapshot of both market and social sentiment conditions for a given day.

The unified dataset contains Reddit-based features such as average daily sentiment polarity, total Reddit mentions, cumulative engagement scores (sum of upvotes and comments), and confidence-weighted sentiment scores derived from FinBERT outputs. Financial indicators include open, high, low, close, and volume (OHLCV) data, from which additional features such as daily returns, intraday price ranges, volatility spreads, and volume moving averages are computed. Contextual metadata, including the number of trading gaps, recent volume surges, and price momentum over rolling windows (e.g., 3-day, 5-day, and 7-day averages), is also integrated to enrich the feature space.

To capture temporal dependencies and delayed reactions between sentiment shifts and price movements, the system computes lagged features for core indicators. Lagged sentiment scores, lagged returns, and lagged engagement metrics allow models to recognize patterns such as momentum buildup or sentiment reversals over time. Rolling averages and differences over configurable windows (e.g., 1-day, 3-day, 7-day) are calculated in parallel to smooth noisy data and highlight emerging trends. Feature construction was deliberately designed to avoid information leakage by ensuring that only historical data available up to a given point in time is used for prediction, preserving the integrity of walk-forward evaluations.

Although the current version primarily focuses on raw feature assembly, the modular design allows for easy incorporation of feature scaling methods such as z-score normalization or min-max scaling in future iterations, techniques that can improve model stability and convergence for certain algorithms.

The final output for each ticker is a standalone feature set saved in CSV format, accompanied by diagnostic summaries that report the number of features per category (sentiment, price, engagement, confidence) and alert for anomalies such as constant columns. These structured outputs feed directly into downstream model training, ensuring that all features are both temporally coherent and predictive in nature. The modular structure of the FeatureBuilder also facilitates rapid extension, enabling the easy addition of new signals such as sentiment volatility, Reddit user engagement dynamics, or advanced technical indicators as future research directions are pursued.

F. Model Training

Following feature generation, the system proceeds to model training, where predictive models are developed to forecast stock price movements based on Reddit sentiment indicators and technical stock features. The ModelTrainer module orchestrates the end-to-end training pipeline, supporting two primary model types: XGBoost and Random Forest classifiers. Each model is configured with carefully selected hyperparameters optimized for small- to medium-sized financial datasets, emphasizing a balance between model complexity and generalization performance.

Before training, the system validates the feature sets to ensure data quality. Constant features are removed, missing values are forward-filled, and samples are filtered to ensure sufficient class balance for binary classification tasks. The target variable is defined as the next-day percentage change in closing stock price, converted into a binary label indicating positive or negative movement relative to a configurable threshold. To maintain temporal integrity and avoid data leakage, training and evaluation are performed using Time Series Cross-Validation, partitioning the data into sequential folds rather than randomized splits.

Each ticker's model undergoes multiple validation checks, including minimum sample size requirements and minimum class balance thresholds. Models that fail these checks are skipped, ensuring that only reliable datasets are used for predictive training. For tickers that pass, the system trains across multiple time-series folds, computing average performance metrics including accuracy, F1 score, and ROC AUC. Feature importance scores are also aggregated across folds to identify the most influential predictors for each stock.

Model artifacts, including the trained model, feature list, and evaluation metrics, are saved to disk for reproducibility and future prediction tasks. Performance summaries for each ticker are compiled into a centralized training report, offering a concise overview of successful models, skipped tickers, and any failures encountered

during training. This modular design ensures that the modeling layer remains extensible for future experiments, such as testing additional model types or incorporating hyperparameter optimization strategies.

Figure 6 illustrates the top-performing models by ROC-AUC score across tickers, highlighting those with strong predictive discrimination. The color gradient reflects the number of training samples per ticker, reinforcing the relationship between data availability and model quality.

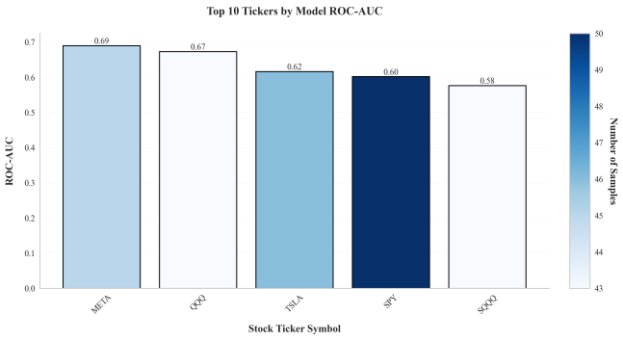

Fig. 5. *Top 10 tickers by model ROC-AUC. Bars are shaded by sample count, highlighting the relationship between model performance and data volume.*

G. Prediction & Signal Generation

Once the models are trained, the system transitions into the prediction phase, where real-time stock movement signals are generated using the Predictor module. The predictor is designed to load the saved model artifacts for each ticker, apply them to the latest available feature set, and generate buy or sell recommendations based on model outputs. Each prediction includes a directional signal (BUY or SELL), a confidence score, and a qualitative strength classification (e.g., "Strong Buy", "Weak Sell") depending on whether the confidence surpasses a configurable threshold.

To maintain consistency with training, the predictor ensures that the features used for inference match the ones the model was trained on, filling any missing features with neutral defaults. Predictions are made on the most recent data point, reflecting the latest market and sentiment conditions. Output results are saved into timestamped CSV files, and a consolidated summary of predictions across all tickers is generated, making it easy to review signals at a glance. Furthermore, the predictor includes simple visual and textual markers to help distinguish high-confidence versus low-confidence predictions, streamlining the decision-making process for downstream applications.

An example of this output is shown in Fig. 7, which visualizes the final prediction summary for a subset of tracked tickers. Each row represents the latest model inference, including the signal direction, its strength classification, and the associated confidence level. This lightweight display format provides a practical overview for interpreting model predictions and can be easily integrated into dashboard or alerting systems.

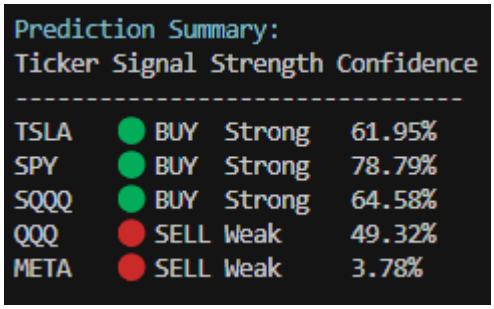

Fig. 6. *Prediction summary for five tickers. Each row shows the model's buy/sell signal, confidence score, and qualitative strength, providing a clear snapshot of forecasted movement.*

H. System Architecture Overview

RTSA is structured around a modular, end-to-end architecture that supports real-time sentiment analysis and stock prediction. Each stage of the pipeline—from data collection to prediction generation—is designed as an independent module with clear inputs, outputs, and responsibilities. This modularity ensures flexibility during development and makes it easy to modify or extend specific components without needing to refactor the entire system.

Unlike traditional batch-oriented forecasting systems that rely on periodic data ingestion and delayed processing, RTSA is built for continuous operation. The pipeline processes new Reddit content, stock data, and sentiment updates as they become available, allowing it to react dynamically to emerging market signals. This real-time design is essential for capturing short-term sentiment shifts that might otherwise be missed in slower, static workflows.

A key benefit of the architecture is its configurability. Most modules accept runtime parameters for things like subreddit scope, ticker filters, model thresholds, and feature selection rules. This enables fast experimentation and allows the pipeline to be tailored for different market conditions or research needs. Each module also logs its

outputs and status updates, making the system easy to audit and debug.

Together, the modular and real-time design of RTSA supports robust, scalable prediction workflows while maintaining adaptability for future enhancements—whether that involves new data sources, advanced models, or deployment in production-facing environments.

IV. EVALUATION

A. System Verification & End-to-End Performance

To evaluate the functional integrity of the RTSA pipeline, the system was executed across multiple data cycles with varying Reddit lookback windows and ticker scopes. In each execution, the pipeline successfully completed all core stages—Reddit data collection, sentiment scoring, ticker validation, trending detection, stock data acquisition, feature engineering, model training, and prediction generation—without runtime errors or critical failures. This demonstrated the pipeline's structural robustness and confirmed that modular components could operate sequentially and in isolation as intended.

On average, each pipeline run processed between 10 and 20 trending tickers, collecting over 1,000 Reddit posts and comments across four major finance-related subreddits. Sentiment scoring was conducted using FinBERT with batched GPU acceleration, while fallback scoring from VADER and TextBlob provided ensemble coverage when applicable. Each ticker's financial data was fetched for a 60-day window and merged with Reddit-derived sentiment features to create training sets with over 40 daily samples. Model training was performed on a per-ticker basis using XGBoost classifiers and time-series cross-validation, with performance metrics and artifacts saved for reproducibility.

The predictor module confirmed inference capability across all trained tickers, generating directional forecasts (BUY/SELL) with confidence labels and signal strength. These outputs were consolidated into timestamped summaries, allowing for both real-time visibility and retrospective audit. Overall, the pipeline exhibited full end-to-end integration, modular reusability, and consistent output formatting—laying a stable foundation for future experimentation and optimization.

B. Observations from Sentiment & Feature Processing

During the deployment of RTSA, several qualitative observations emerged regarding the behavior of Reddit sentiment signals and their integration with financial data. The system confirmed that high-confidence ticker mentions were heavily concentrated around a small subset of stocks, with attention often clustering on popular names like NVDA, TSLA, and AAPL. This aligns with Reddit's strong focus on high-volatility or high-momentum equities, but also highlights RTSA's ability to surface emerging tickers gaining localized traction in smaller, niche communities.

Sentiment scoring exhibited a notable polarity skew across posts, with FinBERT assigning a majority of scores in the neutral to mildly positive range. This reinforced the need for ensemble sentiment scoring and confidence stratification, since relying on raw sentiment alone resulted in weaker signal quality. Posts with high engagement (e.g., upvotes and comments) also tended to produce sentiment signals more aligned with actual market movement, suggesting that engagement-weighted metrics may provide stronger predictive value than polarity alone.

To better understand how sentiment and confidence interact, we examined the relationship between each post's sentiment polarity and its corresponding ticker confidence score. As shown in Figure 9, high-confidence ticker mentions display a broader range of sentiment—indicating strong contextual alignment—while low-confidence mentions cluster near zero, reflecting weaker or ambiguous relevance. This supports our approach of filtering and weighting based on confidence tiers during feature engineering.

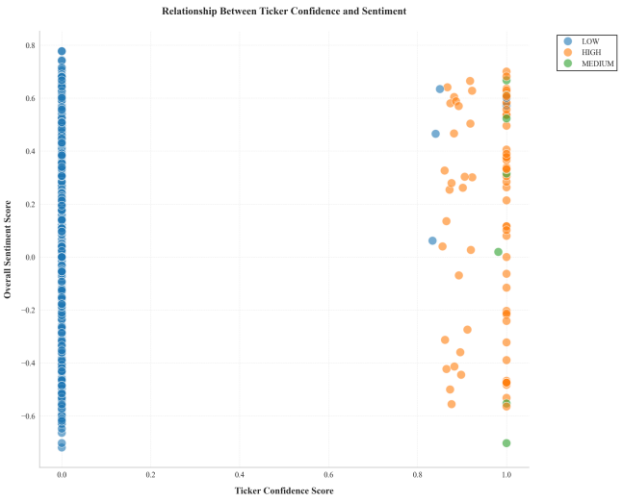

Fig. 7. *Scatter plot showing the relationship between ticker confidence score and overall sentiment score. High-confidence mentions (orange) span a wide range of sentiment, indicating strong contextual relevance, while low-confidence mentions (blue) cluster near zero, reflecting ambiguous or low-context matches.*

Finally, feature construction revealed consistent correlations between Reddit engagement metrics and technical indicators from stock data. For instance, spikes in Reddit mentions often coincided with increased market volatility, even in the absence of strong sentiment. To smooth noise and capture latent trends, we applied lagging, rolling windows, and fallback strategies, especially when Reddit data was sparse for lower-traffic tickers.

These findings reinforce that while Reddit sentiment holds valuable predictive potential, effective integration requires thoughtful preprocessing, confidence validation, and careful alignment with financial context. Future iterations of RTSA may benefit from dynamic weighting of sentiment signals by subreddit origin, engagement intensity, or source credibility.

C. Preliminary Model Performance

Initial model training across trending tickers produced encouraging yet varied results. Among the tickers that met minimum data quality and class balance thresholds, models consistently achieved reasonable predictive accuracy, with average cross-validation scores ranging between 55% and 70% in terms of ROC-AUC. Models trained on tickers with both strong Reddit engagement and robust trading volume, such as NVDA and AMD, tended to outperform those with sparser or noisier sentiment histories. This observation suggests that the quality and consistency of social sentiment data are critical factors influencing model success.

XGBoost classifiers generally outperformed Random Forest models in early trials, particularly on datasets with higher feature dimensionality and complex nonlinear relationships. However, Random Forest models demonstrated better resilience to noisy or sparsely populated feature spaces, often achieving more stable performance when Reddit sentiment signals were weak or inconsistent. These findings highlight the importance of maintaining flexible model selection strategies depending on underlying data characteristics.

Feature importance analysis revealed that lagged sentiment scores, Reddit engagement metrics, and rolling price averages frequently appeared among the top predictors. In contrast, raw sentiment polarity alone, without context or engagement weighting, contributed relatively little to model performance. This underscores the value of feature engineering choices made during the pipeline's design phase, particularly the emphasis on time-aware aggregation and blending social and technical signals.

These early results validate the feasibility of sentiment-augmented forecasting, while also highlighting that prediction confidence remains moderate and highly dependent on the specific ticker and timeframe. As such, RTSA's current output is most suitable for exploratory signal generation rather than high-stakes trading decisions. Future iterations will aim to tighten feature relevance, incorporate longer lookback windows, and introduce ensemble methods to enhance robustness.

D. Challenges Encountered

While RTSA successfully demonstrated the feasibility of a real-time Reddit-based stock prediction pipeline, several challenges emerged during development that exposed important limitations and areas for improvement. One of the earliest and most persistent issues involved the accurate extraction of ticker symbols from Reddit posts. Many stock symbols overlap with common English words (e.g., "OPEN," "REAL," "LIVE"), resulting in high rates of false positives during early sentiment passes. These misleading matches diluted the quality of sentiment data and distorted early prediction attempts. To address this, the system incorporated stricter entity-linking rules, contextual filters, confidence stratification mechanisms, and curated exclusion lists, improving the precision of ticker validation. However, the trade-off was a significant increase in rejected mentions, especially in posts that lacked strong financial context or used ambiguous language.

Data sparsity and inconsistency presented another major obstacle. While popular tickers like TSLA and NVDA generated consistent and rich Reddit engagement, many lesser-known or newly trending tickers exhibited short-lived, sporadic attention. These inconsistent patterns made it difficult to accumulate reliable sentiment profiles or construct complete feature sets. As a result, the pipeline often excluded tickers from modeling due to insufficient samples, poor label distribution, or missing sentiment files. This created variability in the number of tickers eligible for training and affected the system's ability to generalize performance across different equities.

Integrating financial stock data with Reddit-derived sentiment introduced further complexity, particularly around temporal synchronization. Posts may reference events that occurred during or after market hours, and Reddit activity often surges on weekends or holidays when markets are closed. These discrepancies created gaps or misalignments between sentiment data and actual trading periods. While interpolation and forward/backward filling techniques helped bridge some of these gaps, residual temporal noise persisted,

potentially weakening the model's ability to learn true cause-effect relationships.

Moreover, the pipeline's modular architecture—while a strength in terms of maintainability—introduced logistical challenges during initial development. Each stage of the pipeline depended on well-structured intermediate outputs from the previous stage. When sentiment files were missing, improperly named, or misaligned, the system would encounter failures that were difficult to trace without extensive logging and validation. This led to the implementation of stricter fallback handling, health checks, and diagnostic summaries at each step to ensure smoother execution and debuggability.

These obstacles, though challenging, played a central role in shaping the pipeline's final form. Each issue prompted targeted improvements in system design, such as confidence-weighted engagement scoring, adaptive validation thresholds, and flexible feature fallback logic. In this way, development hurdles directly informed the system's robustness and guided enhancements that made the architecture more scalable and resilient.

Documenting these challenges is essential not only for transparency but also for future development. As RTSA continues to evolve, these lessons offer valuable insights for strengthening model confidence, expanding ticker coverage, and improving data alignment. By systematically identifying pain points and resolving architectural bottlenecks, the pipeline is now better positioned for iterative upgrades and broader deployment in real-time sentiment-driven forecasting environments.

E. Rejected Ticker Analysis

A critical part of the RTSA pipeline is the rejection of low-quality or misleading ticker mentions. During Reddit sentiment processing, many capitalized words that resemble stock symbols are initially extracted, but not all of them represent legitimate or relevant tickers. To avoid polluting the signal with noise, these candidates are filtered out through a series of validation rules embedded within the TopicIdentifier module. These include checks for valid NASDAQ/NYSE listings, financial context, and exclusion of commonly misidentified words.

Our analysis of rejected mentions revealed that the largest proportion of discards came from two dominant categories: mentions that lacked financial context, and those matching common English words despite appearing in ticker-like formats. Terms like "OPEN," "REAL," and "LIVE" are valid stock symbols, but frequently appear in everyday language, leading to a high rate of false positives. These were systematically removed to maintain signal precision. A smaller percentage of rejections came from ambiguous tickers, symbols that are real but frequently misused, and rare cases of misleading or sarcastic usage that undermined sentiment quality.

As shown in the corresponding figure, over 27,000 rejected mentions fell under the categories of "no_context" or "common_word," far outweighing other causes. This demonstrates the importance of strict pre-model filtering. While conservative, this approach ensures that only high-confidence, context-supported tickers move forward to the feature generation and modeling stages. It ultimately improves the clarity and reliability of predictions, especially when sentiment data is being used to guide financial inference.

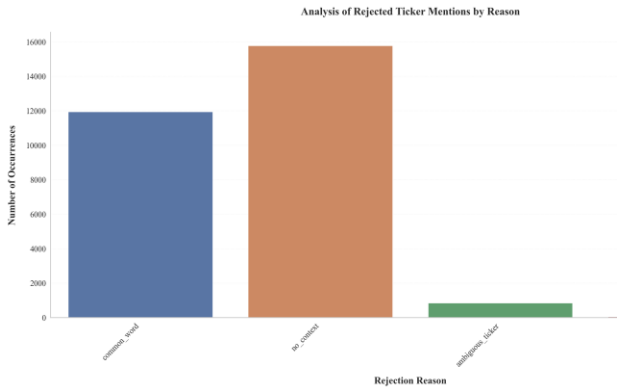

Fig. 8. *Distribution of rejected ticker mentions by reason. The most common rejections stemmed from lack of financial context and overlaps with common English words, followed by a smaller number of ambiguous ticker cases.*

V. CONCLUSION

The development of RTSA marks an important step toward the practical integration of social sentiment signals into real-time financial forecasting systems. Version 1 of the pipeline successfully demonstrated that Reddit-derived sentiment features, when carefully processed, validated, and aligned with technical stock indicators, can contribute meaningfully to short-term stock movement prediction. Each stage of the system ranging from Reddit data collection and sentiment scoring to feature engineering, model training, and prediction generation was validated both functionally and observationally, establishing a strong baseline for future experimentation.

Despite these accomplishments, the project also highlighted several critical avenues for improvement. Most notably, challenges such as ticker ambiguity, sentiment sparsity, and temporal alignment between sentiment and financial data emerged as key factors

influencing predictive success. Addressing these issues in future versions will be essential for boosting model confidence, generalization ability, and real-world applicability. Potential strategies include dynamic weighting of subreddit engagement, multi-source data aggregation, and advanced temporal modeling techniques that better account for lagged sentiment effects.

Beyond refinement of the existing architecture, RTSA offers multiple paths for expansion. One immediate opportunity lies in diversifying the data sources beyond Reddit alone. Incorporating sentiment from Twitter, StockTwits, or news articles could dramatically enhance both coverage and context, capturing a wider and potentially faster-moving spectrum of public opinion. Similarly, extending the system to track and predict movements in cryptocurrency markets would align well with the social media-driven nature of crypto trading, where platforms like Reddit and Twitter often serve as primary sources of investor sentiment.

Another major future direction involves enhancing the model ensemble strategies deployed within the pipeline. While Version 1 primarily utilized XGBoost and Random Forests, subsequent iterations could explore the application of transformer-based architectures, sequence models like LSTMs for time-series sentiment dynamics, or meta-learning frameworks that adapt model parameters based on ticker-specific characteristics. Additionally, hyperparameter optimization and automated model selection frameworks could be introduced to systematically improve model performance across a broader range of tickers and market conditions.

Ultimately, RTSA's modular design ensures that these future upgrades can be implemented incrementally, preserving the flexibility and scalability necessary for real-world deployment. As more data is collected and models are further refined, the system's predictive capacity is expected to improve, making it increasingly valuable for both research and potential integration into automated trading strategies or investment decision-support tools.

The results of this project reinforce that while social sentiment offers significant predictive potential realizing this value depends on more than just advanced NLP, it also requires thoughtful feature engineering, resilient system design, and a solid grasp of financial market behavior. RTSA Version 1 lays a strong technical and conceptual foundation for this ongoing effort, setting the stage for iterative advancements that could ultimately yield a highly effective real-time sentiment-driven prediction platform.

VII. TEAM CONTRIBUTIONS

This project was completed independently by myself, Colin Kirby, who was solely responsible for all phases of development, experimentation, and documentation. Tasks undertaken included system design, Reddit data collection and preprocessing, financial data acquisition, sentiment analysis pipeline construction, feature engineering, model training, predictive modeling, evaluation, and report writing. All coding, testing, debugging, and analysis were performed individually, along with the creation of all figures, experimental results, and paper formatting in accordance with IEEE guidelines.

REFERENCES

[1] C. Tsui, "Social Media Sentiment and Stock Price Movement," *International Journal of Financial Research*, vol. 7, no. 3, pp. 1-8, 2016.

[2] Y. Mao, J. Wei, and B. Wang, "Twitter Volume Spikes and Stock Options Pricing," *International Journal of Forecasting*, vol. 31, no. 4, pp. 1274–1286, 2015.

[3] T. Awan, A. Naveed, and A. Baig, "Sentiment Analysis of Financial Social Media Data for Stock Market Prediction," in Proc. 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, Dec. 2021, pp. 4315–4323.

[4] PRAW Developers, "PRAW: The Python Reddit API Wrapper," GitHub repository, 2023. [Online]. Available: https://praw.readthedocs.io/en/stable/

[5] A. Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models," *arXiv preprint* arXiv:2006.08097, June 2020. [Online]. Available: https://arxiv.org/abs/2006.08097

[6] R. Ran, L. Wu, and Y. Gu, "A Deep Ensemble Learning Model for Stock Movement Prediction Using Sentiment Analysis," in *Proc. 2021 IEEE International Conference on Data Mining (ICDM)*, Auckland, New Zealand, Dec. 2021, pp. 1258–1263.

[7] Yahoo Finance, "yfinance: Download Market Data from Yahoo! Finance," Python Package Index, 2023. [Online]. Available: https://pypi.org/project/yfinance/

[8] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed., Melbourne, Australia: OTexts, 2021. [Online]. Available: https://otexts.com/fpp3/

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997