Financial Time-Series Forecasting with Deep Learning Models and Social Media Sentiment

Yejin Hwang
Texas A&M University–Corpus Christi
Corpus christi, Texas, USA
yhwang@islander.tamucc.edu

Sreelekha Guggilam
Texas A&M University-Corpus Christi
Corpus christi, Texas, USA
sreelekha.guggilam@tamucc.edu

Abstract

Stock price forecasting remains challenging due to the nonlinear nature of financial markets and the influence of investor sentiment. Recent advances in Transformer-based architectures have achieved strong results in time-series forecasting. However, most prediction models are univariate, limiting their ability to incorporate behavioral signals. This study examines the impact of Reddit-derived investor sentiment on financial forecasting through a multivariate deep learning framework based on the Temporal Fusion Transformer (TFT). The framework integrates FinBERT-based sentiment scores and spike-related behavioral features from r/WallStreetBets with traditional market variables such as price, volume, and calendar effects. Forecasting performance was evaluated for LSTM, PatchTST, TimesFM, and TFT-baseline models against sentiment-informed TFT model (TFT-sentiment). These findings demonstrate that incorporating Reddit-based behavioral sentiment enhances both forecasting accuracy and investment performance, effectively bridging quantitative modeling and behavioral finance.

CCS Concepts

• **Computing methodologies** → **Natural language processing**; **Machine learning**; **Modeling and simulation**; • **Applied computing**;

Keywords

Time series forecasting, financial forecasting, transformers, LLM, sentiment analysis

ACM Reference Format:
Yejin Hwang and Sreelekha Guggilam. 2026. Financial Time-Series Forecasting with Deep Learning Models and Social Media Sentiment. In *Proceedings of The 12th Mining and Learning from Time Series (MILETS), KDD Workshop (MILETS '26).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

MILETS '26, Jeju, Korea

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Stock price forecasting remains a challenging problem due to the complex, nonlinear nature of financial markets and the significant influence of human sentiment on investor behavior [6]. Traditional statistical models or curve fitting algorithms such as ARIMA, SARIMA, or splines rely solely on historical price data, which limits their capacity to capture market dynamics driven by behavioral and contextual factors.

Deep learning has significantly improved financial time-series forecasting by modeling complex nonlinear dependencies, with results that significantly outperform traditional ML counterparts[13, 27, 32, 33]. Recent advances in Transformer-based architectures, including Chronos-T5[1] and TimesFM[10], have utilized self-attention for scalable time-series forecasting with strong zero-shot performance. However, these foundation models are limited by their univariate training on single-channel sequences, preventing the integration of external drivers like sentiment.

The influence of social media on financial markets has been heavily documented in prior literature[7, 17, 35, 37]. Tracking sentiments expressed in news and tweets is becoming one of the most important activities for investors since it influences stock prices[6, 23]. According to herding behavior theory, retail investors often choose to buy stocks that capture their attention and tend to follow the crowd, leading to strong, trend-like movements[3, 29]. While early studies focused on general public mood from platforms such as financial news[11, 14, 34] or Twitter[6, 18, 35, 36] to predict market trends, Reddit has more recently emerged as a unique and powerful environment for observing retail investor behavior[9, 24, 30, 37]. In particular, the r/WallStreetBets(WSB)[1] community gained global attention during the GameStop (GME) short-squeeze event in 2021, when retail investor coordination led to extreme market volatility[5, 20]. Despite this growing influence, effectively quantifying and integrating such decentralized discourse into predictive models remains challenging. Unlike centralized financial news, these 'bottom-up' signals require advanced NLP architectures to filter noise and extract meaningful behavioral intent[26].

There are two main approaches employed in text sentiment analysis: the lexical approach and the machine learning approach. Lexical approaches aim to map words to sentiments by building a lexicon or a "dictionary of sentiment," as utilized by Loughran and McDonald[21]. VADER (Valence Aware Dictionary for sEntiment Reasoning)[16] is a general sentiment analysis tool designed with a focus on social media data. However, the original VADER lexicon lacked finance-specific words (like "Bear" and "Bull") and unique financial social media jargon (like "Diamond Hands") used on r/WallStreetBets. While previous studies attempted to build

[1] https://www.reddit.com/r/wallstreetbets/

Reddit-specific lexicons using LDA modeling[20], these efforts focused specifically on the GameStop rally event and remain limited in temporal scope. Manually creating and validating comprehensive sentiment lexicons is time and labor intensive.

Alternatively, machine learning approaches involve training a model using previously seen text to predict or classify the sentiment of new input text. Pretrained BERT (Bidirectional Encoder Representations from Transformers)[12]-based language models do not require additional training for sentiment analysis. Mishev et al. (2020)[23] demonstrated that lexicon-based approaches can be efficiently replaced by modern NLP transformers in the finance domain, showing superior performance compared to lexicon-based methods, statistical methods, word encoders, and sentence encoders. FinBERT[2] is a BERT-based language model pretrained on financial text (articles, news) that can classify text sentiment as positive, negative, or neutral. While FinBERT handles financial keywords, it lacks emoji capability unlike VADER. Following Mishev et al.'s findings that distilled versions of NLP transformers are highly effective for financial discourse, we also include DistilBERT[31] in our comparative evaluation. By assessing these three tools (VADER, FinBERT, and DistilBERT), we aim to identify the optimal sentiment analysis tool for our study, as detailed in Section 3.1.

To incorporate social media sentiment into forecasting, we explored multivariate time-series models. Among available architectures, we adopt Temporal Fusion Transformer (TFT)[19], which provides interpretable plots (e.g., decoder variable importance) while most current architectures are considered 'black-box' models where forecasts are controlled by complex nonlinear interactions between many parameters. By embedding sentiment and social media activity features into the TFT, we aim to capture how behavioral signals interact with market fundamentals to enhance both predictive performance and explainability.

The main contributions of this paper are as follows. First, we conduct a systematic comparison of three sentiment analysis tools—VADER, FinBERT, and DistilBERT—to identify the most effective approach for extracting sentiment from financial social media discourse. Second, we demonstrate that incorporating social media sentiment and engagement metrics into multivariate forecasting significantly improves predictive performance compared to price-only baselines. Finally, we leverage the explainability mechanisms of the Temporal Fusion Transformer to quantify the relative importance of sentiment features in stock price prediction, providing insights into how behavioral signals influence market dynamics. Our framework offers a practical approach for integrating decentralized investor sentiment into financial trading strategies.

2 Background

2.0.1 *Long Short Term Memory(LSTM).*

Long Short-Term Memory (LSTM)[15] is a specialized RNN architecture designed to remember both short-term and long-term values. It captures long-term dependencies in sequential data by mitigating the vanishing gradient problem. The core mechanism of an LSTM unit is the cell state, C_t regulated by three gates: forget, input, and output gates.

The input gate(i_t) determines how much of the input node's value should be added to the current memory cell's internal state. The forget gate(f_t) determines whether to keep the current value of the memory or forget it. And the output gate(o_t) determines whether the memory cell should influence the output at the current time step. The mathematical operations at each time step t are defined as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{5}$$
$$h_t = o_t \odot \tanh(C_t) \tag{6}$$

Where:

- x_t: Input vector at time t.
- h_{t-1}: Hidden state from the previous time step.
- f_t, i_t, o_t: Forget, input, and output gates, respectively.
- $C_t, \tilde{C}_t$: Cell state and Input node.
- σ, tanh: Sigmoid and tangent activation functions.
- $\odot$: Element-wise (Hadamard) product.

Cell state(C_t) has fixed weight 1 to prevent vanishing or exploding gradient problem. LSTM became popular in time series prediction problem since it works well on long-term dependencies.

2.0.2 *Patch Time Series Transformer(PatchTST).*

Patch Time Series Transformer (PatchTST)[25] is Transformer-based models for multivariate time series forecasting and self-supervised representation learning. It utilize Patching design and channel-independence. While traditional point-wise(single time step) approaches struggle to capture semantic meaning like a word in a sentence, patching allows the model to observe local trend and comprehensive semantic information.

Beyond temporal patching, PatchTST also adopts a 'Channel-independent' architecture, which differs significantly from the conventional channel-mixing approach. In channel mixing approach, input token takes the vector of all time series features and projects it to the embedding space to mix information. All the series share the same attention patterns, which may be harmful if the underlying multivariate time series carries series of different behavior.

On the other hand, Channel-independence means that each input token only contains information from a single channel. Since each time series is passed through the Transformer separately, it generates its own attention maps. That means different series can learn different attention patterns for their prediction. Channel-independent models are less likely to overfit data during training.

2.0.3 *Time-series Foundation Model(TimesFM).*

Time-series Foundation Model(TimesFM)[10] is time-series foundation model for forecasting whose out-of-the-box zero-shot performance on a variety of public datasets developed by Google Research. TimesFM is based on pretraining a decoder style attention model with input patching, using a large time-series corpus comprising both real-world and synthetic datasets. Unlike PatchTST, which focuses on encoder-based representation learning, TimesFM is trained in decoder-only setting. In other words, given a sequence of input patches, the model is optimized to predict the next patch as a function of all past patches. Similar to LLMs this can be done in parallel over the entire context window, and automatically enables the model to predict the future after having seen varying number of input patches. In addition to its decoder-only backbone, TimesFM distinguishes itself through output patching, which allows the model to predict a sequence of future values simultaneously rather than point-by-point.

2.0.4 Temporal Fusion Transformer (TFT). Temporal Fusion Transformer (TFT) [19] is an attention-based deep learning architecture designed for high-performance, interpretable multi-horizon forecasting. While traditional RNN-based or Transformer-based models often considered as 'black-boxes' lacking interpretability, TFT can provide variable importance plot through specialized components.

The multi-horizon forecasting model is defined as shown in Equation (7), which predicts the q-th sample quantile for a τ-step-ahead horizon. It generates forecasts by integrating historical targets (y), observed inputs ($\mathbf{z}$), known future inputs ($\mathbf{x}$) within a finite look-back window k, alongside static metadata ($\mathbf{s}$).

$$\hat{y}_i(q,t,\tau) = f_q(\tau, y_{i,t-k:t}, \mathbf{z}_{i,t-k:t}, \mathbf{x}_{i,t-k:t+\tau}, \mathbf{s}_i) \tag{7}$$

To effectively process these diverse inputs, TFT incorporates several key architectural innovations. First, **Variable Selection Networks (VSN)** are utilized to perform instance-wise feature selection, allowing the model to prioritize relevant signals while filtering out noise from both static and time-varying variables. Second, a **Locality Enhancement Layer** based on an LSTM encoder-decoder structure is applied to capture local temporal patterns and handle the differing dimensions of past and future inputs naturally. Lastly, **Gated Residual Networks (GRN)** enable the model to adaptively skip unnecessary components, preventing overfitting in simpler datasets while maintaining capacity for complex relationships.

In this study, TFT_baseline serves as an ablation control to measure effect of sentiment integration. This baseline utilizes standard quantitative variables, including price ratio, trading volume, and temporal features such as days since the most recent earnings announcement. TFT_sentiment extends the baseline by incorporating Reddit-derived sentiment signal and activity-spike features.

This comparative framework systematically evaluates how the integration of social media sentiment improves forecast accuracy and investment performance.

3 Methodology

We define the prediction task as a supervised learning problem where the objective is to forecast the 5-day forward return of a stock using a multi-modal feature set. This set includes normalized OHLCV ratios, calendar data, market volatility, and social media sentiment features.

3.0.1 Feature Normalization. To ensure scale-invariance across different assets and timeframes, we normalize raw price data into relative ratios. For each price component $P \in \{\text{Open, High, Low, Close}\}$, the ratio $R_{P,t}$ is calculated relative to the previous day's closing price C_{t-1}: $R_{P,t} = \frac{P_t - C_{t-1}}{C_{t-1}}$

where P_t is the specific price metric at time t and C_{t-1} is the closing price of the preceding trading period. Similarly, we calculate the volume ratio $R_{V,t}$ as: $R_{V,t} = \frac{V_t - V_{t-1}}{V_{t-1}}$

3.0.2 Target Variable. Following previous study in stock market analysis[28], we focus on predicting price changes rather than absolute values. The target variable $y_{t,5}$ represents the 5-day ahead return, calculated as the percentage change between the current closing price C_t and the closing price five trading days into the future C_{t+5}: $y_{t,5} = \frac{C_{t+5} - C_t}{C_t}$

3.1 Sentiment Analysis Tool Accuracy Validation

To determine the most effective sentiment analysis model for processing Reddit-sourced financial text, we conducted a comparative evaluation of three architectures: VADER (a lexicon-based model) [16], FinBERT (a finance-domain transformer) [2], and DistilBERT (a distilled transformer baseline) [31].

3.1.1 Benchmark Dataset. The models were evaluated using the Financial PhraseBank (FPB) dataset [22], which contains financial statements labeled by multiple annotators. To assess model adaptability and consistency, we measured classification accuracy across four distinct subsets categorized by annotator agreement levels: 50%, 66%, 75%, and 100%.

3.1.2 Model Selection. Performance was quantified by the proportion of correctly classified sentences across all agreement subsets (see Table 1). FinBERT was selected as the primary scoring model for this research due to its performance.

Dataset	**Vader**	**Finbert**	**Distilbert**
Sentences_AllAgree.txt	0.5707	**0.9717**	0.2584
Sentences_75Agree.txt	0.5627	**0.9473**	0.2667
Sentences_66Agree.txt	0.5563	**0.9182**	0.2912
Sentences_50Agree.txt	0.5429	**0.8896**	0.2992

Table 1: Model Accuracy Comparison by Dataset

3.2 Sentiment Feature Engineering

3.2.1 Sentiment Signal Construction. To construct a reliable sentiment signal from Reddit posts, we implemented a FinBERT-based classification and aggregation pipeline. Following the Attention-Induced Trading Hypothesis [4], we treat neutral posts as weakly positive signals ($0 \le s_i \le 1$). Any mention of a ticker of social discourse on Reddit increases the stock's visibility and expands investors' "choice set".

For sentiment score setting, posts classified as Positive or Negative were assigned values of +1 and −1, respectively. To capture subtle investor tone in Neutral posts, we assigned them a weight based on their raw FinBERT[2] softmax probability[2]. This effectively treated neutrality as a weakly positive signal (0 to 1).

$$s_i = \begin{cases} +1 & \text{if } y_i = \text{positive} \\ -1 & \text{if } y_i = \text{negative} \\ P_{neu} & \text{if } y_i = \text{neutral} \end{cases} \tag{8}$$

The daily sentiment indicator S_t is calculated as the mean of s_i across all posts for a given ticker on day t.

[2] https://github.com/ProsusAI/finBERT

Ticker	Days	Mentions	Avg Ment.	Spike Ct	Rate (%)	Max Inten.	Avg Sent.
AAPL	1000	967	0.967	59	5.9	7.119	-0.019
AMZN	1000	1123	1.123	57	5.7	8.395	0.005
GOOGL	1000	874	0.874	44	4.4	12.664	0.021
PLTR	1000	1327	1.327	55	5.5	10.391	0.059
WMT	1000	961	0.961	52	5.2	11.127	-0.038
V	1000	786	0.786	50	5.0	5.524	-0.043
NFLX	1000	1052	1.052	51	5.1	9.511	0.016
TSLA	1000	1197	1.197	55	5.5	4.749	-0.002
MA	1000	864	0.864	50	5.0	9.308	0.002
NVDA	1000	780	0.780	49	4.9	7.726	0.081

Table 2: Summary Statistics of Reddit Activity across Selected Tickers (1,000 Days). Mentions represents the total post count, with Avg Ment. showing the daily mean. Spike Ct and Rate (%) indicate the frequency and proportion of days with abnormal posting surges, respectively. Max Inten. denotes the peak magnitude of these surges, and Avg Sent represents mean sentiment score.

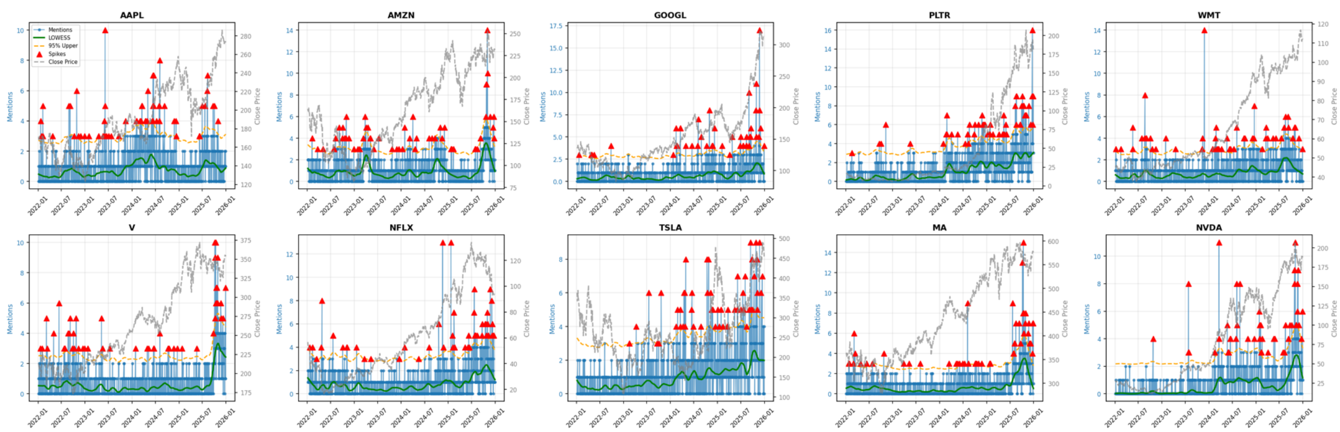


Figure 1: Historical Reddit mention volume and spike detection for the 10 selected tickers (2022–2025). The blue bars represent daily mention counts, while the green line indicates the dynamic baseline estimated via LOESS smoothing. Red triangles denote detected spikes where the activity exceeds the 95% confidence interval upper bound (orange dashed line). Normalized closing prices are shown as grey dashed lines to illustrate the temporal relationship between retail attention and market movement.

3.2.2 Temporal Alignment and Synchronization. All post timestamps are synchronized with the NYSE trading calendar to ensure temporal consistency between behavioral signals and market activity:

- **Intra-day alignment**: Posts created before 4:00 PM ET are assigned to the current trading day.
- **Off-market adjustment**: Posts created after market close or during non-trading days are shifted to the next trading session.

3.2.3 Multi-scale Sentiment Features. To capture varied temporal impacts, we derive lagged and rolling features:

- Lagged Sentiment: We define $S_{t,\text{lag } k} = S_{t-k}$ for $k \in \{1, 2, 3, 4, 5, 7, 14\}$ to capture delayed impacts while ensuring causality.
- Sentiment Mean: To quantify medium-term trends, the rolling mean for window w is calculated as[3]:

$$\bar{S}_{t,w} = \frac{1}{w} \sum_{i=t-w}^{t-1} S_i, \quad w \in \{3, 14\} \tag{9}$$

3.3 Spike and Event Features

To capture abnormal surges in investor attention that may influence short-term price dynamics, we introduce spike-based features derived from Reddit posting activity. A spike represents a statistically significant increase in posting frequency relative to the expected baseline, signaling heightened market attention or event-driven discourse.

[3] $Sentiment\_mean$, $spike\_presence\_sum$ and $spike\_intensity\_max$ for 7-days was excluded due to high-correlation with 3 and 14-days (Refer appendix).

3.3.1 LOESS-based Spike Detection. We utilize a LOESS (Locally Estimated Scatterplot Smoothing)[8] model to estimate a dynamic baseline for Reddit activity. LOESS is a non-parametric regression technique that fits local polynomial functions to data subsets, effectively smoothing non-linear and noisy time-series patterns. By constructing a 95% confidence interval around the smoothed curve, the upper bound serves as an adaptive, time-varying threshold ($C_{t,95\%}$) that accounts for local trends and seasonal patterns.

A spike is identified on any day t where the actual post count c_t exceeds this upper bound. We define two primary variables to capture the occurrence and magnitude of these anomalies:

$$\text{SpikePresence}_t = \begin{cases} 1 & \text{if } c_t > C_{t,95\%} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$\text{SpikeIntensity}_t = \max(c_t - C_{t,95\%}, 0) \tag{11}$$

This adaptive thresholding approach mitigates the impact of non-stationarity and noise, providing a robust alternative to static cutoffs. Figure 1 represents reddit mention volume and spike detection for the 10 selected tickers (2022–2025).

3.3.2 Aggregated Spike Features. To maintain strict causality and prevent data leakage, all spike-based variables are computed using historical lookback windows. We derive two aggregated indicators to summarize the frequency and extremity of retail attention:

- Spike Presence Sum ($w \in \{3, 14\}$): The total number of spikes within each lookback window[3].

$$\text{SpikeSum}_{t,w} = \sum_{i=t-w}^{t-1} \text{SpikePresence}_i \tag{12}$$

- Spike Intensity Max ($w \in \{3, 14\}$): The magnitude of the most intense spike within the window[3].

$$\text{SpikeMax}_{t,w} = \max_{i \in [t-w, t-1]} (\text{SpikeIntensity}_i) \tag{13}$$

These metrics aggregate sustained retail interest and extremity, which serve as potential precursors to short-term market volatility.

3.4 Baseline models

We employ the Temporal Fusion Transformer (TFT) as a multivariate deep learning framework capable of integrating both financial and behavioral covariates. For performance benchmarking and explainability analysis, the experimental models are compared against five baselines:

- **LSTM**: A standard Recurrent Neural Network (RNN) architecture used for capturing long-term dependencies in time-series data.
- **PatchTST**: A transformer-based model that utilizes patching to extract local semantic information from time-series.
- **TimesFM (Zero-shot/Hyperparameter-tuned)**: A foundation model for time-series forecasting, tested in both its original zero-shot state and after hyperparameter tuning.
- **TFT Baseline**: A standard implementation of the Temporal Fusion Transformer without the proposed sentiment enhancements for direct comparison.

4 Experimental Setup

This section describes data sample construction, evaluation metrics, and implementation details.

4.1 Datasets

4.1.1 Stock Data. We evaluate our approach on the US stock market. We collected historic price data from Yahoo Finance, covering Jan 5, 2022, to December 29, 2025 (1000 trading days). We defined our initial stocks-of-interest as the top 20 US stocks by market capitalization to ensure high liquidity and data availability, from which we selected 10 representative tickers (AAPL, AMZN, GOOGL, NVDA, TSLA, NFLX, PLTR, WMT, V, and MA) based on their high social media mention volume and sentiment density. Other major tickers were excluded to ensure the model's focus on sentiment-driven price action. All trading dates were aligned with the official U.S. market calendar, with weekends and public holidays excluded to ensure consistency with actual trading sessions.

4.1.2 Social Media Data. Social sentiment data were collected from Reddit's r/WallStreetBets (WSB) using the Python PRAW API. This community, with over 20 million members, represents one of the most active forums for retail investor discussions. Stock-specific keywords were used to filter relevant posts (e.g., "tesla," "tsla," "elon," "cybertruck," "robotaxi" for Tesla)(see appendix). Both post titles and contents were extracted, and log value of Reddit score (upvotes minus downvotes) was used to indicate community interest and engagement for each post. To ensure linguistic consistency, we only extracted english post. All posts were timestamped in UTC and subsequently realigned with the NYSE trading calendar (posts after 4:00 PM ET assigned to the next trading day). Table 2 is summary statistics of reddit posts we collected.

4.2 Metrics

We evaluate model performance using a comprehensive set of metrics that capture both prediction accuracy and trading performance. All metrics are computed on 5-day-ahead predictions during the period *November 3–December 11, 2025.*

The evaluation metrics are categorized into predictive performance and portfolio performance. For predictive performance, we use Information Coefficient (IC), Rank Information Coefficient (Rank IC), Precision@K(Prec@K), and Directional Accuracy. Each matrix is used to evaluate the alignment (between predicted and actual returns), ranking accuracy, and top-K precision of predictions, and direction of change respectively. Portfolio performance metrics include Return, Sharpe Ratio (SR), and Maximum Drawdown (MDD), which access profitability, risk-adjusted return, and maximum loss during the investment period.

4.3 Implementation Details

All forecasting models were implemented in Python and trained on a consistent dataset spanning 1,000 trading days. To ensure a rigorous and fair comparison across different architectures, we standardized the input context length to 96 days and the prediction horizon to a 5-day forward return.

The selection of a 96-day context length was driven by both domain-specific and architectural considerations (see Figure 2).

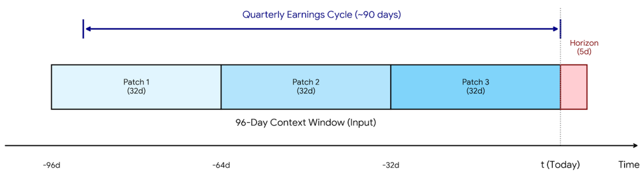


Figure 2: TimesFM input architecture: 96-Day Context Alignment

From a financial perspective, this window effectively encompasses a full quarterly earnings cycle, which typically occurs every three months (approximately 90 days). Capturing this period allows the models to incorporate the volatility and price dynamics associated with the most recent earnings announcements.

Technically, this choice aligns with the structural requirements of the **TimesFM** model, which processes sequences in discrete patches. Since the default input patch length for TimesFM is 32, a context length of 96 (32×3) ensures optimal alignment during tokenization, thereby preventing unnecessary padding and potential information loss.

4.3.1 *Training Configuration:Data Split.*

- Training: October 1, 2022–September 30, 2025 (752 trading days)
- Validation: October 1-31, 2025 (23 trading days)
- Evaluation : November 1–December 11, 2025 (28 trading days)

LSTM, Patch TST, TimesFM are trained with loss $\mathcal{L}$, Mean Squared Error (MSE) and Adam optimizer.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (\hat{r}_i - r_i)^2 \tag{14}$$

4.3.2 TimesFM Hyperparameter Optimization. To optimize the performance of the TimesFM-2.0-500M foundation model for our specific financial forecasting task, we conducted a systematic grid search. While architectural parameters such as the number of layers (50), model dimensions (1,280), and input/output patch lengths (32/128) remained fixed to maintain compatibility with pre-trained weights [4], three critical hyperparameter dimensions(Learning Rate, Batch Size, and Context Length) were explored.

4.3.3 Experimental Design and Results. We evaluated 18 total combinations across the following search space: Learning Rate ($LR \in \{10^{-6}, 10^{-5}, 10^{-4}\}$), Batch Size ($B \in \{32, 64, 128\}$), and Context Length ($C \in \{96, 128\}$). The model was fine-tuned for 50 epochs on an NVIDIA A100 GPU. Table 3 summarizes the performance of the top-performing configurations based on validation loss.

The optimal configuration ($LR = 10^{-5}, B = 128, C = 96$) was selected for final evaluation on the held-out test set, achieving a final validation MSE of 0.4106. Table 4 summarizes hyperparameters for all models evaluated in this study.

[4]https://huggingface.co/google/timesfm-2.0-500m-pytorch

Rank	LR	Batch	Context	Train Loss	Val Loss
1	$\mathbf{10^{-5}}$	**128**	**96**	**0.0208**	**0.4106**
2	10^{-5}	32	96	0.0310	0.4141
3	10^{-5}	64	96	0.0222	0.4172
4	10^{-6}	64	96	0.0558	0.5064
5	10^{-4}	128	96	0.3775	0.5558

Table 3: TimesFM Fine-tuning: Top 5 Grid Search Results

Model	Hyperparameter	Value
Common	Context / Prediction Length	96 / 5
	Batch Size	128
LSTM	Hidden Size	32
	Dropout Rate	0.1
PatchTST	Patch Length / Stride	16 / 8
	d_model / n_heads	32 / 4
	Encoder Layers	2
	Dropout Rate	0.1
TimesFM	Learning Rate	1e-5
TFT	Hidden Size / Attention Heads	32 / 4
	Hidden Continuous Size	16
	Dropout Rate	0.1
	Max Encoder Length	96

Table 4: Hyperparameters for the benchmarked models.

5 Results

Table 5 and Figure 3 present a comparative prediction analysis of the six models and the prediction plots across all six models, respectively. Compared with the all baseline models, our new TFT sentiment approach significantly outperforms across both predictive and portfolio metrics. The proposed model achieved a Directional Accuracy (DA) of 0.586 and an Information Coefficient (IC) of 0.178 (Table 5). From a portfolio perspective, the integration of sentiment signals resulted in a Sharpe Ratio of 2.740, representing a substantial improvement over other baseline models.

5.1 Predictive Effect of Social Media Sentiment

Table 6 presents a comparative analysis of the Information Coefficient (IC) between the baseline and the sentiment-enhanced model across ten major tickers. The results demonstrate a substantial performance gain, with the average IC improving from −0.0776 to 0.0882, representing an overall enhancement of 213.57%. Notably, stocks such as NFLX (4335.11%), AMZN (130.42%) and TSLA (80.83%) exhibited the most significant improvements in alignment, suggesting that the integration of social media sentiment effectively corrects the predictive bias found in the baseline model.

5.2 Investment Performance

We adopt the most common investment strategy named Long/Short Strategy to demonstrate our profitability. Our approach significantly outperforms the baselines in terms of average cumulative long-short return and Sharpe ratio in Figure 4. The TFT sentiment model outperformed all baselines, achieving a 20.19% cumulative

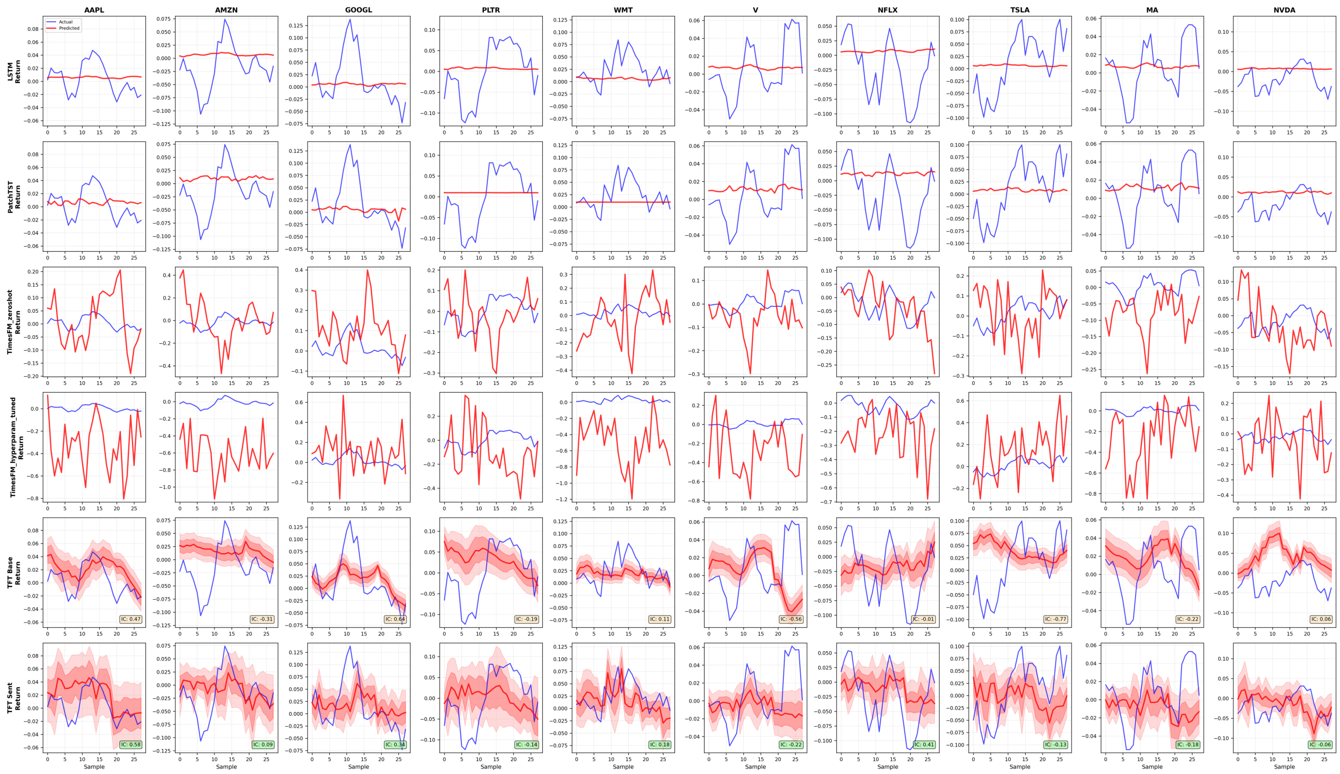


Figure 3: Time-Series Prediction Results for Selected High-Sentiment Stocks. The grid displays the actual (blue) versus predicted (red) returns over the test sample period. Rows represent different deep learning architectures: LSTM, PatchTST, TimesFM(Zeroshot), TimesFM(Hyperparameter-Tuned) TFT (Base), and TFT (Sentiment-augmented). Y-axis is uniformed across all plots besides TimesFM due to high variance of prediction values. Columns represent the 10 selected tickers. TFT baseline and sentiment model include the Information Coefficient (IC) to compare effect of Reddit-sentiment.

return and a Sharpe Ratio of 2.740, which significantly exceeds the baseline TFT (16.63% and 1.670).

5.3 Explainability

To investigate the decision-making mechanism of the TFT-Sentiment model, we analyze the variable importance weights derived from the Temporal Fusion Transformer (TFT) decoder. This analysis provides insights into how the model prioritizes different input features when forecasting financial returns.

As illustrated in Figure 5, the model exhibits a clear hierarchy in feature utilization. We categorize the key findings from this importance distribution as follows:

Notably, sentiment-related features, such as `sentiment_lag3` and `sentiment_mean_14d`, rank highly in the importance scale. The prominence of `sentiment_lag3` suggests a delayed reaction of market prices to retail investor sentiment, confirming that information from platforms like Reddit (e.g., r/wallstreetbets) provides a predictive lead time for the model.

The high ranking of `rolling_volatility` and `spike_intensity` measures demonstrates the model's ability to adjust its forecasts during periods of market turbulence. By prioritizing these features, the TFT-Sentiment model can effectively distinguish between stable trends and high-risk regimes.

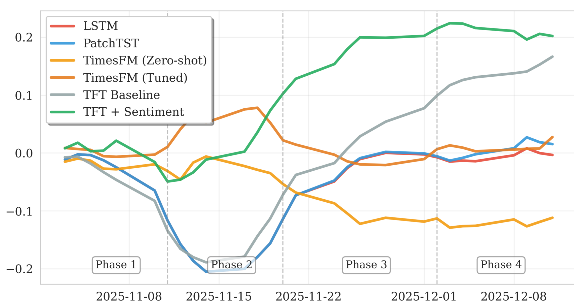
These findings validate that the integration of sentiment analysis into the TFT architecture allows the model to capture non-linear relationships that traditional financial models might overlook.

6 Conclusion

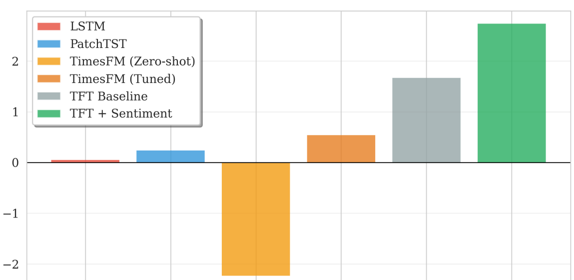
This paper investigates whether Reddit sentiment and social media activity can enhance stock price prediction when integrated into the Temporal Fusion Transformer (TFT) framework. The primary contributions include a systematic comparison of three sentiment analysis tools i.e. VADER, FinBERT, and DistilBERT, for financial social media sentiment analysis. Empirical results demonstrate that incorporating lagged sentiment features and activity metrics like spike features significantly improves both predictive accuracy and investment performance compared to baseline models relying solely on traditional financial data like stock closing price. Through systematic explainability analysis using TFT's attention mechanisms, we quantify the relative importance of sentiment signals alongside conventional factors such as earnings cycles and seasonal patterns.

Model	DA	IC	Rank IC	Prec@3	Prec@5	Ret (%)	Sharpe	MDD (%)
LSTM	0.486	0.055	0.064	0.262	0.464	-0.347	0.052	-20.304
PatchTST	0.500	-0.080	-0.061	0.357	0.493	1.530	0.239	-20.304
TimesFM_Zeroshot	0.446	-0.165	-0.161	0.238	0.464	-11.185	-2.231	-12.362
TimesFM_Tuned	0.500	-0.016	-0.034	0.333	0.450	2.751	0.539	-9.176
TFT_Baseline	0.536	-0.024	0.024	0.238	0.471	16.631	1.670	-18.355
TFT_Sentiment	**0.586**	**0.178**	**0.168**	**0.405**	**0.643**	**20.194**	**2.740**	**-6.902**

Table 5: Comprehensive performance comparison across all models. Best results are highlighted in bold. For all reported metrics except MDD, higher values indicate superior performance, whereas lower MDD is better.



(a) Cumulative Returns



(b) Sharpe Ratios

Figure 4: Investment Performance: (a) Cumulative returns and (b) Sharpe ratios within a single column.

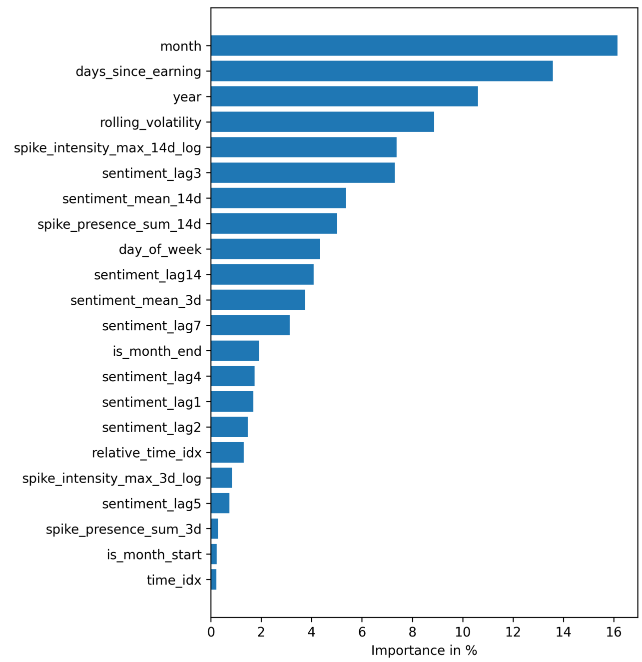


Figure 5: Decoder variable importance for the TFT-Sentiment model, illustrating the relative contribution of each feature to the final prediction.

The decoder variable importance plots reveal that the model effectively learns to balance behavioral signals from r/WallStreetBets with fundamental market drivers, providing transparency into the decision-making process and bridging the gap between 'black-box' deep learning architectures and interpretable financial domain expertise.

Several promising directions remain for future investigation. Extending the analysis to incorporate additional social media platforms beyond Reddit (e.g., Twitter/X, StockTwits) could provide a more comprehensive view of retail investor sentiment and improve model generalization. Alternately, investigating causal mechanisms through Granger causality tests or interventional analysis would help distinguish genuine predictive signals from spurious correlations in sentiment-price relationships. Additionally, developing ensemble approaches that combine multiple sentiment extraction methods could leverage the complementary strengths of lexicon-based and transformer-based tools. Finally, real-time deployment and backtesting on live trading scenarios would validate the practical applicability of our framework under realistic market conditions, including transaction costs and liquidity constraints. These extensions would further strengthen the integration of alternative data sources into rigorous, production-ready financial forecasting systems.

References

[1] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815* (2024).

[2] Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063* (2019).

[3] Asad Ayoub and Ayman Balawi. 2022. Herd behavior and its effect on the stock market: An economic perspective. *Calitatea* 23, 188 (2022), 285–289.

Ticker	Baseline IC	Sentiment IC	Improv. (%)
AAPL	0.4744	0.5774	21.72
AMZN	−0.3098	0.0942	130.42
GOOGL	0.6380	0.3443	−46.04
MA	−0.2229	−0.1778	20.25
NFLX	−0.0097	0.4102	4335.11
NVDA	0.0597	−0.0620	−203.94
PLTR	−0.1884	−0.1438	23.67
TSLA	−0.7720	−0.1273	83.50
V	−0.5607	−0.2153	61.59
WMT	0.1149	0.1820	58.31
Average	−0.0776	0.0882	213.57

Table 6: Comparison of Information Coefficient (IC): Baseline vs. Sentiment-Enhanced Model

[4] Brad M Barber, Xing Huang, Terrance Odean, and Christopher Schwarz. 2022. Attention-induced trading and returns: Evidence from Robinhood users. *The Journal of Finance* 77, 6 (2022), 3141–3190.

[5] André Betzer and Jan Philipp Harries. 2022. How online discussion board activity affects stock trading: the case of GameStop. *Financial markets and portfolio management* 36, 4 (2022), 443–472.

[6] Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science* 2, 1 (2011), 1–8.

[7] Hailiang Chen, Prabuddha De, Yu Hu, and Byoung-Hyoun Hwang. 2014. Wisdom of crowds: The value of stock opinions transmitted through social media. *The review of financial studies* 27, 5 (2014), 1367–1403.

[8] William S Cleveland and Susan J Devlin. 1988. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association* 83, 403 (1988), 596–610.

[9] Shaen Corbet, Yang Hou, Yang Hu, and Les Oxley. 2022. We Reddit in a forum: The influence of message boards on firm stability. *Review of Corporate Finance* 2, 01 (2022), 151–190.

[10] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*.

[11] Min-Yuh Day and Chia-Chou Lee. 2016. Deep learning for financial sentiment analysis on finance news providers. In *2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*. IEEE, 1127–1134.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[13] Eva Dezsi and Ioan Alin Nistor. 2016. Can deep machine learning outsmart the market? a comparison between econometric modelling and long-short term memory. *Romanian Economic and Business Review* (2016).

[14] Lodi Dodevska, Viktor Petreski, Kostadin Mishev, Ana Gjorgjevikj, Irena Vodenska, Lou Chitkushev, and Dimitar Trajanov. 2019. Predicting companies stock price direction by using sentiment analysis of news articles. In *Proceedings of the 15th Annual International Conference on Computer Science and Education in Computer Science*. 37–42.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[16] Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, Vol. 8. 216–225.

[17] Peiran Jiao, Andre Veiga, and Ansgar Walther. 2020. Social media, news media and the stock market. *Journal of Economic Behavior & Organization* 176 (2020), 63–90.

[18] Wasiat Khan, Mustansar Ali Ghazanfar, Muhammad Awais Azam, Amin Karami, Khaled H Alyoubi, and Ahmed S Alfakeeh. 2022. Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing* 13, 7 (2022), 3433–3456.

[19] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting* 37, 4 (2021), 1748–1764.

[20] Suwan Long, Brian Lucey, Ying Xie, and Larisa Yarovaya. 2023. "I just like the stock": The role of Reddit sentiment in the GameStop share rally. *Financial Review* 58, 1 (2023), 19–37.

[21] Tim Loughran and Bill McDonald. 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of finance* 66, 1 (2011), 35–65.

[22] Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology* 65, 4 (2014), 782–796.

[23] Kostadin Mishev, Ana Gjorgjevikj, Irena Vodenska, Lubomir T Chitkushev, and Dimitar Trajanov. 2020. Evaluation of sentiment analysis in finance: from lexicons to transformers. *IEEE access* 8 (2020), 131662–131682.

[24] Markus Münster, Felix Reichenbach, and Martin Walther. 2024. Robinhood, Reddit, and the news: The impact of traditional and social media on retail investor trading. *Journal of Financial Markets* 71 (2024), 100929.

[25] Y Nie. 2022. A Time Series is Worth 64Words: Long-term Forecasting with Transformers. *arXiv preprint arXiv:2211.14730* (2022).

[26] Nuno Oliveira, Paulo Cortez, and Nelson Areal. 2017. The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with applications* 73 (2017), 125–144.

[27] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. 2020. Deep learning for financial applications: A survey. *Applied soft computing* 93 (2020), 106384.

[28] Kijeong Park, Sungchul Hong, and Jong-June Jeon. 2025. Dynamic higher-order relations and event-driven temporal modeling for stock price forecasting. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*. 6048–6056.

[29] Adik Duwi Rahayu, Aditya Putra, Chiata Oktaverina, and Regina Aulia Ningtyas. 2021. Herding behavior in the stock market: A literature review. *International Journal of Social Sciences Review* 1, 2 (2021), 08–25.

[30] Felix Reichenbach and Martin Walther. 2023. Financial recommendations on Reddit, stock returns and cumulative prospect theory. *Digital Finance* 5, 2 (2023), 421–448.

[31] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[32] Georgios Sermpinis, Charalampos Stasinakis, and Christian Dunis. 2014. Stochastic and genetic neural network combinations in trading and hybrid time-varying leverage effects. *Journal of International Financial Markets, Institutions and Money* 30 (2014), 21–54.

[33] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing* 90 (2020), 106181.

[34] Wataru Souma, Irena Vodenska, and Hideaki Aoyama. 2019. Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science* 2, 1 (2019), 33–46.

[35] Andrew Sun, Michael Lachanski, and Frank J Fabozzi. 2016. Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction. *International Review of Financial Analysis* 48 (2016), 272–281.

[36] Emanuele Teti, Maurizio Dallocchio, and Alberto Aniasi. 2019. The relationship between twitter and stock prices. Evidence from the US technology industry. *Technological Forecasting and Social Change* 149 (2019), 119747.

[37] Sonja Warkulat and Matthias Pelster. 2024. Social media attention and retail investor behavior: Evidence from r/wallstreetbets. *International Review of Financial Analysis* 96 (2024), 103721.

A Research Pipeline

Figure 6 demonstrates our work flow chart.

B Feature Engineering

To select features for TFT model, we visualize correlation plots as following. The log value of Reddit score (upvotes-downvotes per post) variable was selected to reflect post's popularity and total comments and mentions were excluded due to high correlation with reddit score. For sentiment mean, spike presence, and spike intensity, 7-day was excluded due to high correlation with 3 and 14-day.

Heatmap for feature selection for TFT framework.

Correlation plot.

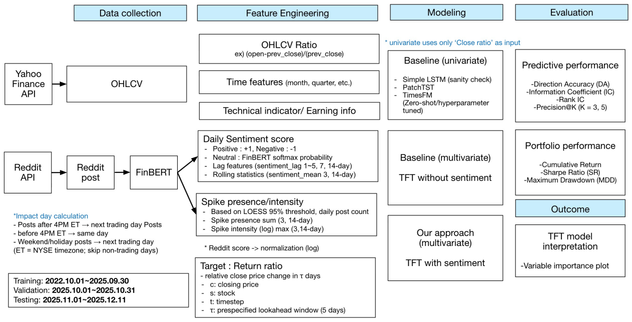


Figure 6: Flow chart of the steps in our proposed framework for stock market forecasting using DL models and social media sentiment

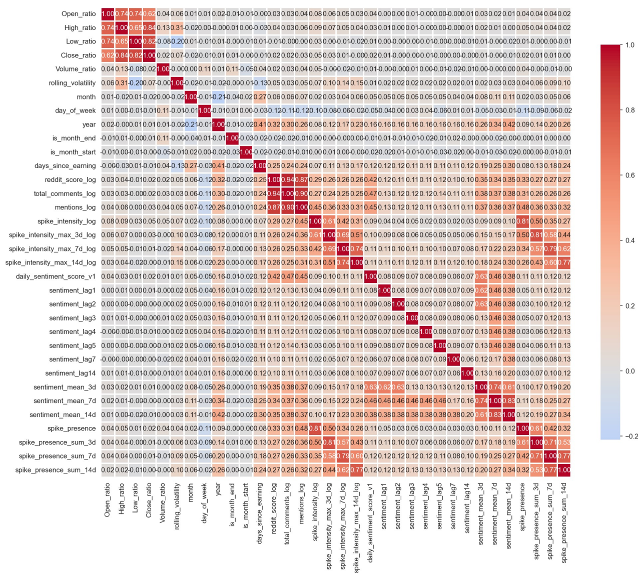


Figure 7: Heatmap for feature selection for TFT framework

B.0.1 Log Transformation of Spike Intensity. We applied logarithmic transformation to the spike intensity features for the following reasons:

As illustrated in Figure 9, the original spike intensity features exhibited highly right-skewed distributions. Most values were concentrated near zero, while extreme outliers extended up to 12-13, creating a heavy-tailed distribution that could negatively impact model training stability.

Mathematically, we applied: $log\_feature = log(1 + feature)$

Log transformation (bottom row, Figure 9) compressed the value range from [0, 13] to approximately [0, 2.5], resulting in more stable gradients during temporal fusion transformer training. The transformed distributions show reduced variance and more symmetric spread across tickers.

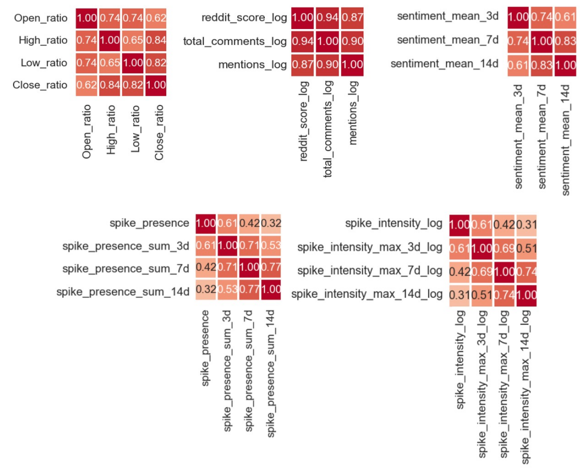


Figure 8: Heatmap for feature selection for TFT framework

C Evaluation Metrics Formulation

C.0.1 Prediction Accuracy Metrics. **Information Coefficient (IC):** The Pearson correlation coefficient between predicted returns and realized returns, defined as:

$$\text{IC} = \text{corr}(\hat{r}, r) = \frac{\text{Cov}(\hat{r}, r)}{\sigma_{\hat{r}} \sigma_r} \tag{15}$$

where $\hat{r}$ denotes predicted returns and r denotes actual returns. IC is the standard metric in quantitative finance for evaluating return prediction quality. Higher values indicate stronger predictive power, with IC > 0.05 considered meaningful in practice.

Rank Information Coefficient (Rank IC): The Spearman rank correlation coefficient, which measures the monotonic relationship between predicted and actual returns:

$$\text{Rank IC} = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{16}$$

where d_i is the difference between the ranks of $\hat{r}_i$ and r_i. Rank IC is more robust to outliers than IC and better captures ordinal prediction quality.

Directional Accuracy (DA): The proportion of predictions where the sign of the predicted return matches the sign of the actual return:

$$\text{DA} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(\text{sign}(\hat{r}_i) = \text{sign}(r_i)) \tag{17}$$

DA measures the model's ability to predict the direction of price movement, which is critical for trading decisions.

Precision@K (Prec@K): A cross-sectional metric measuring stock selection accuracy. For each trading day t with M stocks, we define:

$$\text{Prec@K}_t = \frac{|\text{Top-K}_{\hat{r}}^{(t)} \cap \text{Top-K}_r^{(t)}|}{K} \tag{18}$$

where Top-K$_{\hat{r}}^{(t)}$ denotes the set of K stocks with highest predicted returns on day t, and Top-K$_r^{(t)}$ denotes the set of K stocks with

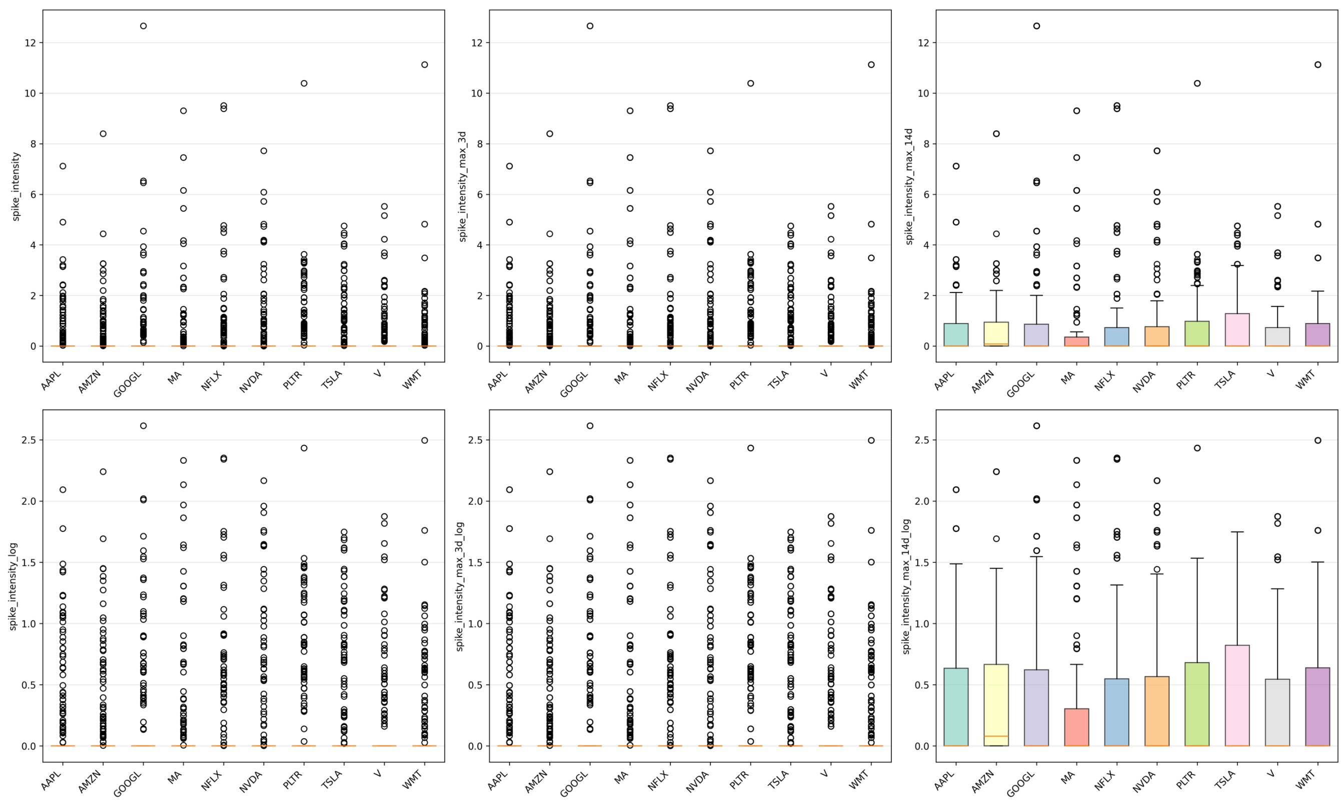


Figure 9: Distribution of spike intensity features before (top) and after (bottom) log transformation. Left to right: base feature, 3-day maximum, and 14-day maximum. Log transformation reduces right skewness and compresses value range from [0, 13] to [0,2.5], improving numerical stability for model training.

highest actual returns. The final metric is the average across all days:

$$\text{Prec@K} = \frac{1}{T}\sum_{t=1}^{T}\text{Prec@K}_t \tag{19}$$

We report Prec@3 and Prec@5 to evaluate top-stock selection ability, which directly reflects portfolio construction performance . This cross-sectional formulation is standard in stock ranking literature.

C.0.2 Trading Performance Metrics. To evaluate practical trading performance, we simulate a simple long-short strategy based on predicted returns. The strategy is defined as:

$$\text{position}_i = \begin{cases} +1 & \text{if } \hat{r}_i > 0 \text{ (long)} \\ -1 & \text{if } \hat{r}_i < 0 \text{ (short)} \\ 0 & \text{if } \hat{r}_i = 0 \text{ (no position)} \end{cases} \tag{20}$$

The strategy return at each time step is:

$$r_{\text{strategy},i} = \text{position}_i \times r_i \tag{21}$$

Cumulative Return: The total portfolio return over the evaluation period, computed using compound returns to reflect realistic capital growth with reinvestment:

$$\text{Return} = \prod_{i=1}^{N}(1 + r_{\text{strategy},i}) - 1 \tag{22}$$

This formulation is standard in finance literature and accounts for the compounding effect of sequential gains and losses.

Sharpe Ratio (SR): The risk-adjusted return, measuring excess return per unit of volatility:

$$\text{SR}_{\text{period}} = \frac{\mathbb{E}[r_{\text{strategy}}] - r_f}{\text{std}(r_{\text{strategy}})} \tag{23}$$

where r_f is the risk-free rate (set to 0 in our experiments). Since our predictions are 5-day-ahead, we annualize the Sharpe ratio as:

$$\text{SR}_{\text{annual}} = \text{SR}_{\text{period}} \times \sqrt{\frac{252}{5}} \approx \text{SR}_{\text{period}} \times 7.1 \tag{24}$$

where 252 is the number of trading days per year, and $252/5 \approx 50.4$ represents the number of 5-day periods annually.

Maximum Drawdown (MDD): The largest peak-to-trough decline in cumulative portfolio value, measuring downside risk:

$$\text{MDD} = \min_{t\in[1,T]} \frac{C_t - \max_{\tau\le t} C_\tau}{\max_{\tau\le t} C_\tau} \tag{25}$$

where $C_t = \sum_{i=1}^{t} r_{\text{strategy},i}$ is the cumulative return at time t. Lower (less negative) MDD indicates better risk control.

D Keywords for extracting posts

Ticker	Keywords
AAPL	aapl, apple, tim cook, iphone, ipad, macbook, ios, vision pro, app store, apple watch, m4 chip, apple ai, siri, wwdc, apple intelligence, macos, tvos, watchos, airpods, homepod, icloud, apple music, apple tv, apple pay, apple card, apple silicon, m1, m2, m3, pro display xdr, magic keyboard, face id, touch id, imessage, facetime, apple arcade, apple fitness
AMZN	amzn, amazon, jeff bezos, andy jassy, aws, prime video, alexa, kindle, amazon delivery, twitch, ecommerce, marketplace, fire stick, blue origin, amazon music, whole foods, prime day, amazon web services, ec2, s3, lambda, redshift, dynamodb, amazon go, amazon pharmacy, ring, eero, fire tablet, audible, zoox, kuiper, bedrock, amazon advertising, fba
GOOGL	googl, goog, google, alphabet, sundar pichai, gemini, youtube, android, google cloud, deepmind, waymo, search engine, tpu, google ai, vertex ai, bard, pixel, chrome, chromebook, gcp, workspace, gmail, google maps, google drive, google photos, nest, fitbit, waze, adsense, google play, tensor, project astra, google i/o, ruth porat
MA	mastercard, priceless, mastercard network, mastercard stock, mastercard visa, credit card, digital payment, michael miebach, fintech, crypto, contactless, transaction, interchange, b2b, cross-border, mastercard send, mastercard identity, vocalink, finicity, open banking, cyber security, ciphertrace, ekata, dynamic yield, sessionm, consumer spending, travel recovery, payment rails, virtual card
NFLX	nflx, netflix, reed hastings, ted sarandos, greg peters, streaming, original, squid game, stranger things, netflix ads, password sharing, subscribers, gaming, binge, tudum, netflix games, netflix shop, bridgerton, crown, witcher, paid sharing, ad tier, churn, content spend, free cash flow, upfronts, nielsen gauge
NVDA	nvda, nvidia, nvidia ai, jensen huang, gpu, h100, b100, blackwell, rtx, cuda, hopper, a100, tensor core, semiconductor, chips, ai hardware, ada lovelace, geforce, g-sync, nvlink, nvswitch, omniverse, dgx, inference engine, l40s, poseidon, nvsmi, jetson, dali, v100, grace hopper, gh200, nvidia drive, dlss, rtx 4090, rtx 5090, tegra, mellanox, infiniband, spectrum-x
PLTR	pltr, palantir, alex karp, karp, peter thiel, gotham, foundry, apollo, palantir aip, palantir ai, government, military, contract, defense, boot camp, artificial intelligence, big data, analytics, decision making, nhs, titan, vantage, maven, skykit, ontology, commercial revenue, government revenue, customer count, net dollar retention, gaap profitability
TSLA	tsla, tesla, elon musk, elon, musk, cybertruck, spacex, neuralink, fsd, full self driving, autopilot, supercharger, electric vehicle, gigafactory, ev, megapack, optimus, robotaxi, battery day, model s, model 3, model x, model y, powerwall, solar roof, giga texas, giga berlin, giga shanghai, 4680 battery, dojo, master plan, starlink, tesla energy, tesla bot
V	visa, visa card, v, visa payment, visa network, visa stock, mastercard visa, ma, credit card, digital payment, ryan mcinerney, fintech, visa crypto, contactless, transaction, interchange, cross-border, visa direct, visa checkout, visa infinite, visa signature, cybersource, plaid, stablecoin, cbdc, payment processing, payment gateway, merchant fees, consumer spending, travel spending, fraud prevention
WMT	wmt, walmart, doug mcmillon, walmart plus, walmart grocery, sams club, retail, walmart stock, ecommerce, walmart pharmacy, walmart dividend, walmart logistics, walmart target, flipkart, black friday, rollback, great value, walmart connect, retail media, supply chain, omnichannel, curbside pickup, walmart health, walmart go, inhome delivery, phonepe, massmart, walmart international, consumer staples, inflation impact

Table 7: Reddit Post Extraction Keywords by Ticker