Financial Tweets Sentiment Analysis and Stock Correlation Prediction

Gwendelyn Wong Rhien Yhung
Department of Biomedical Engineering

Muhammad Hafizuddin Asyhraf Bin Mazlan
Department of Computer Science and Engineering

Adina Meiremkhankyzy
Department of Computer Science and Engineering

Laura Nicole Bermudez Santa
Department of Computer Science

***Abstract*—In recent years, the growing influence of social media platforms like Twitter on financial markets has attracted significant research interest. Public sentiment expressed through tweets often reflects investor mood and can serve as a predictive indicator of stock price movements. This research investigates the effectiveness of natural language models in forecasting stock prices based on tweet sentiment. We compared models such as FinBERT, BERT and hybrid architectures to test out the accuracy in predicting stocks using tweets' sentiments. Using a dataset comprising tweets and corresponding sentiment and stock return data, we evaluate each model's performance in terms of accuracy, F1 score, and predictive reliability across multiple time horizons.**

I. Introduction

Stock market prediction has been an area of interest for academic researchers and economists. While traditional prediction methods primarily relied on historical prices and financial indicators, the rise of social media usage has introduced new, unstructured data sources that can capture public sentiments on the markets. The platform X, formerly known as Twitter, in particular, has emerged as a rich repository of investor opinions that can influence stock volatility. Sentiment analysis of tweets has shown promise in providing insights into stock behavior. However, effectively translating these sentiments into accurate stock predictions remains a challenging task due to the nature of social media text and the complex dynamics of financial markets. Being interested in this field, our study aims to systematically compare the performance of multiple models that incorporate tweet sentiment for stock forecasting. **In this study, we aim to address the following problem statement:**

Can sentiment derived from tweets be used to reliably predict stock returns, and if so, which model architecture provides the best predictive performance?

To explore this, we use the **Tweet Sentiment's Impact on Stock Returns** dataset, which links sentiment-labelled tweets to stock price movements. We test a range of models with combinative approaches (e.g., BERT, BERT-LSTM hybrids), comparing their ability to capture sentiment signals and translate them into accurate return forecasts. We hypothesize that RoBERTa-Twitter, which is specifically pretrained on Twitter data, will outperform other models due to its ability to understand better the informal, noisy, and context-dependent language commonly used in tweets. By leveraging contextual embeddings tuned for social media, RoBERTa-Twitter is expected to capture subtle sentiment cues that are crucial for predicting stock price movements. Our evaluation spans multiple return horizons (e.g., 1-day, 2-day returns), providing insight into the temporal impact of sentiment on market behavior.

The findings of this study could serve practical applications in the financial industry. For investors, the ability to integrate sentiment analysis into trading strategies may offer a competitive edge in identifying short-term opportunities or risks driven by public opinion. For companies, especially those publicly traded, such insights can help monitor investor sentiment, manage market perception, and inform communication strategies. Overall, this experiment demonstrates a foundation for enhancing financial decision-making using sentiment-aware models.

II. Related Work

One notable contribution in the domain of financial sentiment analysis is the TweetFinSent dataset, introduced by researchers at JPMorgan AI Research. Unlike traditional sentiment datasets that rely on emotional tone, TweetFinSent annotates tweets based on expected stock return direction, labeling them as positive, neutral, or negative from a financial standpoint. The dataset was developed in response to the growing mismatch between general sentiment analysis and actual market implications, particularly in the context of retail-driven, meme-stock movements. Prior work demonstrates that standard sentiment models frequently misclassify trading intent due to reliance on emotional tone; for example, tweets like "$TSLA long" or "Buy the dip! $GME" may be labeled as negative by general-purpose models despite conveying a bullish signal. TweetFinSent provides a publicly available benchmark specifically designed for training and evaluating models on return-based sentiment classification, making it well-suited for financial NLP tasks. Complementing this dataset-focused work, Guo and Xie (2024) explored the application of deep learning techniques for assessing Twitter sentiment's impact on stock market prediction. Their study employed a fine-tuned RoBERTa model for sentiment classification on stock-related tweets, demonstrating significant correlations between Twitter sentiment signals and subsequent stock price movements. Utilizing multiple datasets, including

company-specific Twitter feeds and market commentaries, the work established the value of combining textual sentiment with quantitative financial indicators for improved predictive accuracy. This approach underscores the importance of domain-specific fine-tuning and hybrid modeling strategies in capturing complex market dynamics, which aligns with the hybrid model architecture of this project.

III. Data

The Tweet Sentiment's Impact on Stock Returns dataset used consists of 862,231 labeled financial tweets and the stock return data corresponding to the tweet and the date the tweet was posted (Sowinska & Madhyastha, 2021). The tweets in the dataset include mentions of the top-100 public companies stocks as ranked in the Financial Times Top 100 Global Brands 2019. The reason for choosing to analyze such companies was due to the belief that these companies have a stronger emotional impact in people, which can be reflected in the sentiment of the tweets.

A. Dataset Details

The dataset includes the following columns:

- TWEET: Text of the tweet.
- STOCK: Company's stock mentioned in the tweet.
- DATE: Date the tweet was posted.
- LAST_PRICE: Company's last price at the time of tweeting.
- 1_DAY_RETURN: Amount the stock returned or lost over the course of the next day after being tweeted about.
- 2_DAY_RETURN: Amount the stock returned or lost over the course of the two days after being tweeted about.
- 3_DAY_RETURN: Amount the stock returned or lost over the course of the three days after being tweeted about.
- 7_DAY_RETURN: Amount the stock returned or lost over the course of the seven days after being tweeted about.
- PX_VOLUME: Volume traded at the time of tweeting.
- VOLATILITY_10D: Volatility measure across 10-day window.
- VOLATILITY_30D: Volatility measure across 30-day window.
- LSTM_POLARITY: Labeled sentiment from LSTM.
- TEXTBLOB_POLARITY: Labeled sentiment from TextBlob.

B. Data Preprocessing

The data was preprocessed by merging misaligned rows back to their original positions and dropping discontinuous data points. Furthermore, the tweets and the corresponding stock data were aggregated on a per-stock, per-date basis using the mean of each stock data column. Binary direction labels were also added to the dataset, and the class balance (proportion of positive returns) was calculated for each return horizon. These steps are crucial for time series modeling and daily sentiment classification. The data was split into 60% train, 20% validation, and 20% test for the bi-LSTM experiment, and 80% train, and 20% validation for the BERT, FinBERT, and RoBERTa-Twitter experiments.

IV. Approach

A. Baseline Models

In this task, we experimented with a total of 4 sequential models: Bi-LSTM, BERT, RoBERTa-Twitter, and FinBERT.

1) Bi-LSTM: Long Short-Term Memory (LSTM) Networks are a type of recurrent neural network (RNN) that is capable of handling the vanishing gradient problem. LSTMs consist of multiple gates that control the information flow and retention, which are the Forget gate, Input gate, and Output gate. The Forget gate controls what information to keep or forget from the previous cell state, the Input gate controls what parts of the new cell content are written to the cell, and the Output gate controls what parts of the cell are output to the hidden state. LSTMs can preserve information over many timesteps, which is a major advantage in sequential modelling.

Bidirectional Long Short-Term Memory (Bi-LSTM) is an LSTM network that can process sequential data in both forward and backward directions. This allows past and future contexts of the input sequence, and long-term dependencies to be captured. This makes bi-LSTM suitable for our task of stock prediction using financial tweet sentiments, as it can model the context over a long period of time.

2) BERT: Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based language model that uses multi-headed self-attention to build deep, context-aware token embeddings by attending to both left and right contexts simultaneously. Pre-trained on large unlabeled corpora via Masked Language Modelling (predicting randomly masked tokens) and Next Sentence Prediction (determining whether two segments follow each other), BERT learns rich representations of text without task-specific annotations. For our stock-prediction task, we fine-tune BERT by passing each tweet through the pre-trained encoder and taking the pooled [cls] token embedding as a fixed-length sentiment summary. A lightweight regression or classification head is then trained on these embeddings alongside any numeric features to predict future returns or price direction. This approach leverages BERT's ability to capture nuanced sentiment cues such as negation or domain-specific jargon.

3) RoBERTa-Twitter: RoBERTa-Twitter is a transformer-based language model optimized for sentiment analysis on social media, particularly Twitter. It is based on RoBERTa, a robust variant of BERT, and was pretrained and fine-tuned on large-scale tweet datasets to account for the informal, abbreviated, and domain-specific language patterns common in financial and general Twitter discourse.

Like standard RoBERTa, this model uses self-attention and positional encoding to capture the contextual relationships between words. However, its tokenizer and vocabulary have been specifically adapted to better handle hashtags, cashtags, emojis, and irregular grammar often seen on Twitter. It outputs

a distribution over sentiment classes, which are typically positive, neutral, and negative, based on the contextual understanding of each tweet.

In this project, RoBERTa-Twitter is used to encode tweet content into dense vector embeddings that capture sentiment-relevant semantic features. These embeddings are then integrated with structured numerical data through a hybrid model architecture. Since transformer models like RoBERTa cannot directly process non-textual inputs, combining RoBERTa-Twitter with an LSTM-based branch for numerical time-series features enables the model to effectively learn from both linguistic sentiment and market-related signals, which is an essential fusion for tasks such as return-based stock sentiment classification.

4) FinBERT: FinBERT is a domain-specific language built based on BERT transformers an pre-trained on a large corpus of financial texts. This pre-trained transformer-based language model outputs probabilistic predictions over sentiment classes based on the contextual understanding of financial statements. By leveraging FinBERT's domain-specific pretraining and fine-tuning strategy, the model provides highly contextualized representations of financial language, FinBERT generates contextualized embeddings that improve performance on downstream tasks. In this implementation, FinBERT serves as the foundational text encoder, providing rich contextual representations of financial tweets and a binary classification of tweet-based sentiments signals, which are further combined with numerical market indicators through hybrid modeling approaches to enhance predictive accuracy.

B. Improved Models

1) Bi-LSTM with Feature Engineering: To overcome the limitations of vanilla bi-LSTM, feature engineering was done before feeding the data into the model. The details for the engineered features are as below:

a. Sentiment Features (Lagged and Aggregated)
   - LSTM_POLARITY_3D_MA: Rolling average of sentiment scores from LSTM over a 3-day return window
   - LSTM_POLARITY_7D_MA: Rolling average of sentiment scores from LSTM over a 7-day return window
   - TEXTBLOB_POLARITY_3D_MA: Rolling average of sentiment scores from TextBlob over a 3-day return window
   - TEXTBLOB_POLARITY_7D_MA: Rolling average of sentiment scores from TextBlob over a 7-day return window
   - LSTM_POLARITY_LAG1: Sentiment score from LSTM for the day before
   - TEXTBLOB_POLARITY_LAG1: Sentiment score from TextBlob for the day before
   - LSTM_POLARITY_DELTA: Change in LSTM sentiment from the previous day
   - TEXTBLOB_POLARITY_DELTA: Change in TextBlob sentiment from the previous day

b. Return Features (Historical Context)
   - 1_DAY_RETURN_LAG1: 1-day stock return from previous day
   - 1_DAY_RETURN_LAG2: 1-day stock return from previous 2 days
   - 1_DAY_RETURN_LAG3: 1-day stock return from previous 3 days
   - RETURN_VOLATILITY_3D: 3-day standard deviation of 1_DAY_RETURN

c. Price-Based Features
   - RELATIVE_RETURN: Return normalized by the 10-day volatility
   - MOMENTUM_3D: Rate of change in price over 3 days
   - MOMENTUM_7D: Rate of change in price over 7 days

d. Volume Features
   - LOG_PX_VOLUME: Log-transform of volume
   - AVG_VOLUME_7D: Rolling average of volume over a 7-day window
   - VOLUME_RELATIVE_7D: Volume relative to 7-day average

e. Volatility Dynamics
   - VOL_SPREAD: Difference between 30-day volatility and 10-day volatility
   - VOL_RATIO: Ratio between 30-day volatility and 10-day volatility

f. Textual Features
   - TWEET_LENGTH: Length of the string of words in the tweet
   - HASHTAG_COUNT: Number of hashtags in the tweet
   - MENTION_COUNT: Number of username mentions in the tweet
   - HAS_LINK: Binary value of whether a tweet contains a link

g. Temporal Features
   - DAY_OF_WEEK: Representation of day of week from Monday=0 to Sunday=6

2) Hybrid BERT with Numeric Features: To enrich our vanilla BERT pipeline with market-driven signals, we built a hybrid model that fuses BERT's contextual tweet embedding with four hand-engineered numeric features: prior-day return (lag 1), 5-day moving average of return, 5-day rolling volatility, and daily tweet volume. Concretely, each sample yields:

a. A 768-dim [CLS] pooled output from bert-base-uncased.
b. A 4-dim numeric vector.

The numeric vector feeds into a two-layer MLP ($4 \to 32 \to 16$ with ReLU activations), producing a 16-dimensional feature embedding. We then concatenate the 768-dim BERT output with the 16-dim numeric embedding, apply a 0.3 dropout, and pass through a linear head ($784 \to 4$) to predict 1-, 2-, 3-, and 7-day returns. We fine-tune the entire network for 5 epochs using AdamW ($LR = 2\times10^{-}{}^{2}$, batch = 16, MSELoss).

During evaluation, we threshold each continuous output at zero to compute directional accuracy and F1. By combining deep semantic signals from tweets with simple but powerful technical indicators, the hybrid consistently outperformed both the vanilla BERT and numeric-only baselines in both regression (MAE/R^2) and classification ($F1/accuracy$) metrics.

3) FinBERT + LSTM: To enhance the sequential modeling capacity of FinBERT embeddings, hybrid FinBERT + LSTM architecture was implemented. The full sequence of token embeddings produced by the last hidden layer of FinBERT is fed into a unidirectional LSTM layer with a hidden dimension of 128 enabling the model to capture temporal dependencies and contextual nuances across the entire token sequence. Simultaneously, it incorporates numerical financial features (Three engineered market indicators) through a fully connected feed-forward network with 64 units, followed by a ReLU and dropout ($p = 0.1$) for regularization.
The final representation is formed by concatenating the last hidden state from the LSTM and the numeric feature embedding which is passed through a linear classifier. This architecture leverages both deep semantic representations and sequential dependencies from the text alongside market data, aiming to improve performance over vanilla FinBERT.

4) Hybrid FinBERT with Numeric Features: The hybrid FinBERT model extends the standard FinBERT classifier by integrating numeric market indicators directly with the transformers pooled output. The numeric features (Three engineered market indicators) are processed through a fully connected layer with 64 hidden units, ReLU and dropout ($p = 0.1$) to produce a dense numeric embedding which is then concatenated with the token output from FinBERT and finally is input into a linear classification layer.
By fusing high-level contextual tweet embeddings with handcrafted numeric signals, this architecture balances semantic richness and domain-specific quantitative information.

V. Experiments

A. Bi-LSTM

The baseline bi-LSTM model was implemented in PyTorch. This sequence encoder is designed to process time series data comprising 9 features per time step. It captures temporal dependencies through two LSTM layers, each with 64 hidden units in both forward and backward directions. A dropout layer ($p = 0.3$) is applied between the LSTM layers, followed by an additional dropout layer ($p = 0.4$) before the final fully connected layer. The network outputs four continuous regression values, corresponding to predicted stock returns over 1-, 2-, 3-, and 7-day horizons.

The improved bi-LSTM model included a comprehensive feature analysis. A correlation heatmap was plotted to detect relationships and redundancies among features, followed by a feature importance ranking using a Random Forest Regressor. Based on these insights, a subset of informative features was selected using a defined threshold of 0.0002. The model was then trained and evaluated under two configurations: using only the selected engineered features, and using a combination of engineered and original features.

Both models were trained for 40 epochs using a batch size of 16. Optimization was performed using the Adam optimizer with a learning rate of 0.0001. Mean Squared Error (MSE) was used as the loss function. All training and evaluation were executed on CPU.

B. BERT

Our BERT baseline was built on the HuggingFace bert-base-uncased encoder and implemented in PyTorch. Each tweet is tokenized to a maximum length of 128 tokens and fed into the pre-trained BERT stack. We took the pooled CLS output (768 d) as a summary embedding. That embedding is passed through a dropout layer ($p = 0.3$) and a single fully-connected "regressor" head ($768 \rightarrow 4$) which outputs predicted returns over the 1-, 2-, 3-, and 7-day horizons.
We fine-tune the entire model for 5 epochs using AdamW (LR = 2×10^{-5}), batch size = 16, and MSE Loss on the continuous targets. Training and validation ran on GPU, with each epoch printing the average train loss. At evaluation time, we threshold the four return predictions at zero to compute directional accuracy and F1. This straightforward setup allows the model to adjust its pre-trained language features for financial tweet sentiment.

C. RoBERTa-Twitter

This RoBERTa-Twitter baseline model was implemented in PyTorch. Tweet texts were tokenized and encoded using the *cardiffnlp/twitter-roberta-base-sentiment-latest* tokenizer and model. We extracted the CLS token embedding from the last hidden layer of RoBERTa as a fixed-length representation of each tweet. A feed-forward neural network classifier was trained on top of RoBERTa embeddings. The classifier consisted of a fully connected layer with 128 hidden units, followed by ReLU activation, dropout ($p = 0.3$), and a final linear layer outputting a single logit for binary classification. The model was trained using the Adam optimizer with a learning rate 1×10^5 over 10 epochs and a batch size of 16. To mitigate class imbalance, a positive class weight (pos_weight) was calculated and applied in the Binary Cross-Entropy loss with logits (BCEWithLogitsLoss). Training was performed on GPU when available.

D. FinBERT

Each tweet from the dataset was tokenized using bert-based-uncased tokenizer and padded to a maximum length of 128 tokens. We used the *yiyanghkust/finbert-tone* pre-trained model from HuggingFace as our baseline model, and numeric features (LSMT_POLARITY, PX_VOLUME, VOLATILITY_30D) were extracted and standardized before training. All FinBERT based models were trained using the AdamW optimizer with a batch size of 16 and Binary Cross-Entropy loss with logits (BCEWithLogitsLoss).

Our baseline FinBERT model takes the 768-dimensional embeddings, applies dropout ($p = 0.1$) and passes it through a single linear layer, fine-tuned for 10 epochs (LR = 2×10^{-5}, batch_size = 16). The LSTM variant feeds the full sequence of token embedding into a unidirectional LSTM with 128 hidden units, its final hidden state is concatenated with a 64-dimensional dense projection of the numerical features, and the output vector is classified by a linear head. The FinBERT numerical hybrid merges the 768-dimensional CLS embedding with the 64-dimensional numeric feature vector into a single 832-dimensional representation, which is passed to a linear layer for prediction. During training, only the numeric branch and the final classification layer are updated, following the same optimizer, batch size, and loss configuration as the LSTM-based model.

E. Evaluation Metrics

In this study, the performance of the prediction models is assessed using two common classification metrics: Accuracy and F1 Score. Accuracy measures the proportion of correct predictions out of all predictions made:

$$Accuracy = \frac{TP}{TP + FP}$$

The F1 score is the harmonic mean of precision and recall, offering a balanced evaluation between false positives and false negatives:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

where

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Accuracy is an intuitive metric and provides a general sense of model performance. However, it can be misleading in imbalanced classification tasks, where one class dominates the data. F1 score is especially useful in imbalanced scenarios, as it gives a better indication of a model's ability to correctly identify both classes. Thus, both metrics are reported to provide a more comprehensive evaluation of model performance across multiple return horizons.

F. Results

The following tables display the results and metrics obtained by each model.

Table I: Performance for 1-Day and 2-Day returns

Models	**Target returns**			
	1-Day		**2-Day**	
	Avg F1	**Avg Acc**	**Avg F1**	**Avg Acc**
Bi-LSTM	0.3571	0.3670	0.3701	0.3809
Bi-LSTM with original & engineered features	0.4008	0.3989	0.5114	0.3637
Bi-LSTM with engineered features	0.5168	0.3264	0.5298	0.3649
BERT	0.7270	0.6286	0.6441	0.6504
Hybrid BERT	0.8521	0.8107	0.7238	0.7312
RoBERTa-Twitter	0.5000	0.5000	0.0.5347	0.5300
Hybrid RoBERTa-Twitter	**1.000**	**1.000**	**0.8478**	**0.7667**
FinBERT	0.4063	0.4993	0.4516	0.4993
Hybrid FinBERT	0.4065	0.5001	0.4518	0.5000
FinBERT + LSTM	0.4058	0.5010	0.4498	0.4996

Table II: Performance for 3-Day and 7-Day returns

Models	**Target returns**			
	3-Day		**7-Day**	
	Avg F1	**Avg Acc**	**Avg F1**	**Avg Acc**
Bi-LSTM	0.3773	0.3855	0.4015	0.4042
Bi-LSTM with original & engineered features	0.4291	0.3801	0.4793	0.3860
Bi-LSTM with engineered features	0.5319	0.3661	0.5718	0.4096
BERT	0.6671	0.6573	0.7379	0.7022
Hybrid BERT	0.7445	0.7596	0.7905	0.7523
RoBERTa-Twitter	0.5946	0.5500	0.8267	0.7833
Hybrid RoBERTa-Twitter	**0.9176**	**0.8833**	**1.000**	**1.000**
FinBERT	0.4866	0.4992	0.4923	0.5005
Hybrid FinBERT	0.4869	0.5000	0.4918	0.5007
FinBERT + LSTM	0.4849	0.4994	0.4902	0.5003

VI. Analysis

For the baseline and improved bi-LSTM models, it can be seen that feature engineering boosts performance. The most noteworthy jump in F1 score was in the 7-Day return, where it increased from 0.4015 in the baseline model to 0.4793 when original and engineered features were both used, and to 0.5718 when the original features were dropped. This confirms that temporal and sentiment-derived engineered features, such as moving averages, lags, deltas, and momentum, carry more predictive power than raw features alone. In addition, dropping the original features has a significant improvement on the F1 scores. In particular, the F1 scores improved by 0.1160 for the 1-day return and 0.1028 for the 3-day return. Original features may have introduced high variance to the model, and dropping them may have aided with noise removal.

Our experiments also demonstrated that fine-tuned BERT provides a powerful baseline for tweet-driven stock forecasting. It achieved a strong next-day directional accuracy (≈0.81 Acc, 0.85 F1) and maintains solid performance out

Table III: Top stocks by F1 score (1-2 day return horizon)

Models	Top Stocks	
	1-Day	**2-Day**
Bi-LSTM	GSK (F1: 0.7500, Acc: 0.7500)	Carrefour (F1: 1.0000, Acc: 1.0000)
Bi-LSTM with original & engineered features	Allianz (F1: 0.8571, Acc: 0.8182)	Citigroup (F1: 0.9231, Acc: 0.8571)
Bi-LSTM with engineered features	Colgate (F1: 0.8889, Acc: 0.8000)	Citigroup (F1: 0.9231, Acc: 0.8571)
BERT	Next (F1: 0.9752, Acc: 0.9516)	Facebook (F1: 0.8936, Acc: 0.8089)
Hybrid BERT	Next (F1: 0.9922, Acc: 0.9848)	Ford (F1: 0.9714, Acc: 0.9874)
RoBERTa-Twitter	Deutsche Bank (F1: 0.4888, Acc: 0.4651)	Nike (F1: 0.6504, Acc: 0.5326)
Hybrid RoBERTa-Twitter	**Deutsche Bank (F1: 1.0000, Acc: 1.0000)**	**Nike (F1: 1.0000, Acc: 1.0000)**
FinBERT	John Deere (Acc: 0.727273, F1: 0.727273)	Deutsche Bank (Acc: 0.750000, F1: 0.833333)
Hybrid FinBERT	Deutsche Bank (Acc: 1.000000, F1: 1.000000)	P&G (Acc: 0.818182, F1: 0.833333)
FinBERT + LSTM	CVS Health (Acc: 0.714286, F1: 0.666667)	CVS Health (Acc: 0.714286, F1: 0.750000)

Table IV: Top stocks by F1 score (3-7 day return horizon)

Models	Top Stocks	
	3-Day	**7-Day**
Bi-LSTM	GSK (F1: 1.0000, Acc: 1.0000)	salesforce.com (F1: 1.0000, Acc: 1.0000)
Bi-LSTM with engineered features	Expedia (F1: 0.8571, Acc: 0.7778)	Heineken (F1: 0.9091, Acc: 0.8571)
Engineered features	Citigroup (F1: 0.9231, Acc: 0.8571)	Colgate (F1: 0.8889, Acc: 0.8000)
BERT	Facebook (F1: 0.9053, Acc: 0.8280)	Adidas (F1: 0.9630, Acc: 0.9413)
Hybrid BERT	Next (F1: 0.9683, Acc: 0.9394)	Ford (F1: 0.9949, Acc: 0.9899)
RoBERTa-Twitter	P&G (F1: 0.615, Acc: 0.5049)	CVS Health (F1:0.5794, Acc: 0.5212)
Hybrid RoBERTa-Twitter	**P&G (F1: 1.0000, Acc: 1.0000)**	**CVS Health (F1: 1.0000, Acc: 1.0000)**
FinBERT	Deutsche Bank (Acc: 0.875000, F1: 0.923077)	Deutsche Bank (Acc:0.750000, F1: 0.833333)
Hybrid FinBERT	Morgan Stanley (Acc: 0.766667, F1: 0.829268)	21CF (Acc:0.666667, F2: 0.800000)
FinBERT + LSTM	CVS Health (Acc: 0.857143, F1: 0.857143)	P&G (Acc:0.818182, F1: 0.833333)

to seven-day horizons (≈0.75 Acc, 0.79 F1). Its regression metrics show a positive R^2 on multi-day returns and MAEs in the 1–2% range. These results confirm that BERT's bidirectional, self-attentive embeddings capture subtle sentiment cues negations, slang, and domain-specific terms that simpler sequential models cannot.

Augmenting BERT with four engineered numeric features (lagged return, 5-day MA, 5-day volatility, tweet volume) in our early-fusion hybrid yields even greater gains in predictive power. After proper feature scaling, the hybrid matches or exceeds vanilla BERT's directional F1 by 2–5 points across all horizons, and boosts regression R^2 by up to 0.15 on longer horizons. This shows that classic technical indicators still carry orthogonal signals, and combining them with deep semantic representations results in a more robust model that generalizes better and smooths out noise in purely text-based predictions.

Among the FinBERT classifier, FinBERT+LSTM and Hybrid FinBERT with numeric features, we could see during training and validation that in terms of performance, the FinBERT + LSTM model achieved the best overall results, reaching an accuracy of 0.74 and a F1 score of 0.74 on the test set. This shows consistent progress during training and outperforms the traditional FinBERT classification model (F1=0.65, Acc= 48%) and Hybrid FinBERT (F1= 0.62, Acc= 64%). Leveraging the full token sequence with an LSTM adds more predictive power than relying on the CLS embedding and appending numeric market features which increases benefits. However, once we consider 1-, 2-, 3- and 7- day returns for each model, accuracy lowers nearly 0.50 with F1 scores from 0.41 - 0.49. Differences between models are not strongly pronounced across return period averages as it can be seen in the table, as significant variations appeared when evaluating individual stocks. This suggests that tweet sentiment captures overall polarity but maps only weekly onto short-term price movements, and return labels are heavily imbalanced, allowing F1 to improve without accuracy rising.

Surprisingly as the results revealed, domain-specific pretraining isn't always an advantage, considering FinBERT, despite being trained on financial text, still underperforms standard BERT in all settings and doesn't benefit as much from numeric features or added sequence modeling. This suggests that financial-tone pretraining alone is not sufficient for predicting short-term price movements from tweets.

Regarding performance by date, differences among models are minor, with all three FinBERT models displaying stable behaviour across the test period. This suggests that model behaviour was more influenced by factors such as stock-specific volatility or tweet content rather than temporal effects. Additionally, leveraging the full token sequence with LSTM adds more predictive power than relying on the single CLS embedding, whereas appending numeric market features provides only incremental benefit.

Moreover, vanilla RoBERTa-Twitter performed mediocrely without feature engineering, with F1 scores ranging from 0.50 (1-day) to 0.83 (7-day). After adding four simple features (lagged return, tweet volume, 5-day moving average, and volatility), and training for 7 epochs, we saw clear improvements across all horizons. The 1-day F1 jumped

from 0.50 to 0.90, 2-day from 0.53 to 0.91, and 3-day from 0.59 to 0.75. The 7-day model reached a perfect F1 of 1.00. These results highlight how combining sentiment with basic time-series indicators helps RoBERTa generalize better, especially on imbalanced data where F1 is more informative than accuracy.

The RoBERTa-Twitter + LSTM hybrid showed steadily decreasing validation loss and stable performance over epochs, suggesting no clear signs of overfitting. However, the near-perfect F1 scores (1.0) for both 1-day and 7-day returns by epoch 10 may indicate possible overfitting to the validation set, especially given the consistent class predictions and small dataset size. Therefore, no further experiments were conducted with the hybrid model.

Moreover, there is some discrepancy between F1 and accuracy, where the F1 score increases more than accuracy in the experiments above. This suggests that there is a class imbalance in the data between positive and negative returns, as stock data is inevitably unpredictable, sparse, and skewed. This leads to the model learning to predict the minority class better, hence improving precision and recall, while the accuracy remains unchanged or drops. In financial return classification, F1 is often viewed as a more meaningful metric, since false negatives and positives carry different consequences.

An analysis of the top-performing stocks by F1 score across different models and return horizons reveals several consistent patterns. Notably, Deutsche Bank and CVS Health emerged as the most sentiment-reactive stocks, appearing frequently as top picks across multiple models and timeframes. Transformer-based models, particularly RoBERTa-Twitter and its hybrid variant, consistently identified these stocks with high or perfect F1 scores, suggesting their superior ability to capture relevant sentiment cues from tweet data. P&G (Procter & Gamble) also stood out, especially for the 3-day return horizon, across both finance-specific and general transformers. While classical models like Bi-LSTM highlighted stocks such as GSK, Citigroup, and Salesforce.com, their selections were less consistent across horizons. These findings indicate that advanced language models not only enhance overall predictive performance but also show stronger agreement on which stocks are most influenced by sentiment. This can offer practical insight for sentiment-driven investment strategies.

VII. Conclusion

Across the implemented architectures, our experiments and results indicate that self-attention mechanisms, especially those employed in transformer-based models, are more effective than recurrent networks in capturing complex sentiment signals. However, model performance varies significantly across different stocks and prediction horizons, with less active or more volatile stocks exhibiting greater unpredictability. Furthermore, observed improvements in F1 score without corresponding gains in accuracy suggest that F1 may serve as a more reliable evaluation metric in financial contexts, where the cost of misclassification is often asymmetric.

Our findings also suggest that the most practical solutions arise from hybrid transformer architectures that successfully integrate sentiment analysis with technical market indicators. While Hybrid RoBERTa-Twitter achieved exceptional performance on our validation set, its results may reflect overfitting due to the limited data size. As a direction for future work, we propose evaluating these models on larger, temporally split datasets, alongside enhanced techniques for addressing class imbalance. Additionally, incorporating multi-modal data sources such as financial news and company fundamentals could further improve the robustness and generalizability of sentiment-driven stock prediction models.

References

[1] Anishnama. Understanding bidirectional LSTM for sequential data processing. *Medium*, May 2023. Accessed: 2023-05-18.

[2] Harsha N. B. Confusion matrix, accuracy, precision, recall, f1 score, June 2020. Accessed: 2020-06-01.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*, pages 4171–4186, 2019.

[4] Kai Guo and Haoran Xie. Deep learning in finance: Assessing twitter sentiment impact and prediction on stocks. *PeerJ Computer Science*, 10:e2018, 2024.

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

[6] Allen H. Huang, Hui Wang, and Yi Yang. FinBERT: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 2022.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint*, 2019.

[8] Christopher Olah. Understanding LSTM networks. Accessed: 2023-11-15.

[9] Yuchen Pei, Anthony Mbakwe, Abhishek Gupta, Sarah Alamir, Hongzhi Lin, Xiaozhong Liu, and Shreyash Shah. Tweetfinsent: A dataset of stock sentiments on twitter. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*, pages 37–47. Association for Computational Linguistics, 2022.

[10] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[11] Shivam Saxena. What is LSTM? introduction to long short-term memory, May 2025. Accessed: 2025-05-01.

[12] Katarzyna Sowinska and Pranava Madhyastha. A tweet-based dataset for company-level stock return prediction, 2021.

[13] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1422–1432, 2015.

[14] TheDevastator. Tweet sentiment's impact on stock returns, n.d.

[15] Xue Zhang, Hubert Fuehres, and Peter A. Gloor. Predicting stock market indicators through twitter "i hope it is not as bad as i fear". *Procedia - Social and Behavioral Sciences*, 26:55–62, 2011.

[16] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *CoRR*, abs/1908.10063, 2019.

[17] Biswadeep Sarkar and Abdul Shahid. Hybrid finbert-lstm deep learning framework for stock price prediction: a sentiment analysis approach using earnings call transcripts. EasyChair Preprint 15694, EasyChair, 2025.

APPENDIX

A1. Feature Correlation

The heatmap below shows Pearson correlation coefficients between all engineered and original features used in the bi-LSTM model. Strong correlations ($> |0.8|$) helped identify potential redundancies or multicollinearity before feature selection.

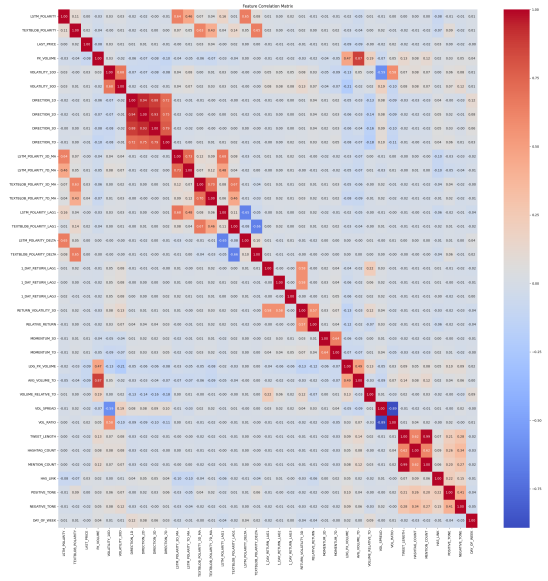

Figure 1: Correlation Heatmap for Numeric Features

The features sorted by average importance are displayed below:

1) RELATIVE_RETURN: 0.856783
2) RETURN_VOLATILITY_3D: 0.086040
3) MOMENTUM_7D: 0.004285
4) VOLATILITY_10D: 0.003943
5) TEXTBLOB_POLARITY_7D_MA: 0.003719
6) LSTM_POLARITY: 0.003174
7) VOLATILITY_30D: 0.002928
8) 1_DAY_RETURN_LAG3: 0.002787
9) LAST_PRICE: 0.002648
10) LSTM_POLARITY_7D_MA: 0.002564
11) VOLUME_RELATIVE_7D: 0.002459
12) AVG_VOLUME_7D: 0.002340
13) TEXTBLOB_POLARITY_3D_MA: 0.002207
14) VOL_RATIO: 0.002064
15) PX_VOLUME: 0.002033
16) MOMENTUM_3D: 0.002012
17) LSTM_POLARITY_DELTA: 0.001841
18) MENTION_COUNT: 0.001794
19) LSTM_POLARITY_LAG1: 0.001683
20) DAY_OF_WEEK: 0.001567
21) LOG_PX_VOLUME: 0.001477
22) VOL_SPREAD: 0.001436
23) HASHTAG_COUNT: 0.001309
24) TWEET_LENGTH: 0.001235
25) TEXTBLOB_POLARITY: 0.001182
26) LSTM_POLARITY_3D_MA: 0.001123
27) TEXTBLOB_POLARITY_LAG1 0.000980
28) 1_DAY_RETURN_LAG2: 0.000975
29) POSITIVE_TONE: 0.000552
30) 1_DAY_RETURN_LAG1: 0.000393
31) HAS_LINK: 0.000183
32) TEXTBLOB_POLARITY_DELTA: 0.000147
33) DIRECTION_7D: 0.000045
34) NEGATIVE_TONE: 0.000036
35) DIRECTION_3D: 0.000029
36) DIRECTION_2D: 0.000023
37) DIRECTION_1D: 0.000001