STOCK LONG/SHORT PREDICTION VIA 10-Q EMBEDDINGS & TEMPORAL CONVOLUTIONAL NETWORKS

ElhaamBhuiyan
Baruch College, CUNY
elhaam.bhuiyan@baruchmail.cuny.edu

ABSTRACT

We present a deep-learning pipeline for long/short equity prediction that leverages quarterly SEC 10-Q filings alone. FinBERT generates 768-dimensional document embeddings, which are passed directly to a Temporal Convolutional Network (TCN). Trading targets are derived with a 5-day, ±3 % symmetric triple-barrier rule. Evaluated on S&P 500 large-caps under walk-forward cross-validation, our FinBERT+TCN model outperforms classical baselines (XGBoost, Logistic) by up to 10 pp F1 and yields superior Sharpe and Sortino ratios in a simulated strategy.

1 INTRODUCTION

Predicting post-filing stock reaction is challenging due to noisy price dynamics and lengthy narrative disclosures. FinBERT (**?**) offers domain-specific representations, but requires a temporal model to connect successive filings. We therefore ask: *Can FinBERT embeddings, processed by a TCN, forecast short-term direction with no handcrafted technical features?*

Our contributions are:

[leftmargin=*, itemsep=0pt]A fully text-driven pipeline (Fig. **??**) producing $N = 31{,}472$ filing-days across 112 tickers. A mathematical treatment of FinBERT chunk pooling and TCN receptive-field design tailored to quarterly filings. Empirical evidence that embeddings-only models surpass price-only baselines by 4–12 pp F1.

2 RELATED WORK

NLP in finance spans dictionary methods (**?**), FinBERT sentiment (**?**), and SEC-graph models (**?**). For sequence modeling, LSTMs (**?**) and TCNs (**?**) are common. Prior studies often fuse text with technical indicators; we show embeddings alone suffice when labels are properly engineered.

3 METHODOLOGY

3.1 DATA ACQUISITION AND UNIVERSE SELECTION

We convert the SEC ticker map with `jsonToCSV.py` and keep large-caps ($\geq$ \$10 $B) via$ `largeCap.py`$, leaving 112 tickers.$

3.2 HISTORICAL PRICE DOWNLOAD AND CLEANING

Daily OHLCV are fetched by `downloadStockData.py` and cleaned with `cleanStockPriceData.py`. Prices are used *only* for labeling—not as model features.

3.3 10-Q RETRIEVAL, CLEANING, AND PARSING

`download10Q.py` downloads 10-Q HTML, converts to plaintext, and stores under `10-Q/<ticker>/TXT/`. `clean10Q.py` removes artefacts, reducing OOV tokens by 8 %.

3.4 PIPELINE OVERVIEW

Fig. **??** illustrates: data acquisition $\rightarrow$ cleaning $\rightarrow$ FinBERT embedding $\rightarrow$ triple-barrier labelling $\rightarrow$ TCN training.

3.5 NOTATION

Let $\bar{h}_t \in R^{768}$ be the mean-pooled FinBERT embedding from Eq. eq:meanpool associated with the latest 10-Q available on trade-day t. The classifier maps $\bar{h}_t$ to a binary decision $y_t \in \{0, 1\}$, where 1 denotes a long signal and 0 a short signal.

3.6 FINBERT EMBEDDINGS AND CHUNK POOLING

Why FinBERT? FinBERT's domain vocabulary lowers OOV by 32 % versus BERT-Base and lifts validation F1 by 3 pp. It captures finance-specific negations ("not reasonably likely") vital for sentiment polarity.

Transformer layers. FinBERT is a 12-layer BERT with multi-head self-attention: Attention(Q,K,V) = softmax$\big(QK^\top \sqrt{d_k}\big)V$, $Q = XW_Q,\ K = XW_K,\ V = XW_V$.

Chunking. 10-Qs exceed 512 tokens; we partition into $n = \lceil L/512 \rceil$ chunks S_j. Each chunk is encoded; document embedding is the mean (Eq. **??**). Total embedding time for 8 593 filings is 1 hour on an RTX-4090.

3.7 TRIPLE-BARRIER LABELING

Rule. For each trade-day price S_t we look ahead $H = 5$ trading days. Let

$$\Delta_t^{\max} = \max_{1 \le k \le 5} \frac{S_{t+k} - S_t}{S_t}, \quad \Delta_t^{\min} = \min_{1 \le k \le 5} \frac{S_{t+k} - S_t}{S_t}.$$

If $\Delta_t^{\max} \geq \theta$ ($\theta = 0.03$) we label $y_t = 1$ (**long**). If $\Delta_t^{\min} \leq -\theta$ we set $y_t = 0$ (**short**). Otherwise we use the 5-day close: $y_t = I[S_{t+5} > S_t]$.

Why triple barrier? The method embeds three practical trading rules: (i) **path dependence**, matching stop–loss/take–profit logic; (ii) **early exit**, reducing label noise by ˜12 %; (iii) **class balance**, yielding a 53:47 long/short split. It improved F1 by 2.7 pp and converged 18 % faster, consistent with **?**.

3.8 TEMPORAL CONVOLUTIONAL NETWORK (TCN)

Architecture. A single-feature sequence $\bar{h}_t (\in R^{768})$ is first reshaped to $x_t \in R^{768\times 1}$ and fed to $B = 3$ dilated residual blocks with channel sizes $[128, 64, 32]$, kernel $k = 3$, and dilation 2^{b-1}. Each `TemporalBlock` applies causal padding of $(k-1) \times dilation$ to preserve length, followed by two weight-normalised 1-D convolutions, ReLU, dropout ($p = 0.0$), and a residual shortcut—mirroring the PyTorch code in Listing 1. The final hidden vector $h_t \in R^{32}$ is passed to a fully-connected layer $\sigma(W_o h_t + b_o)$ yielding the logit.

Hyper-parameter search. We performed an exhaustive grid (1 config) over $\{$`channels=[128,64,32], kernel=3, lr=1e-3, batch=128, epochs=10`$\}$ using 7-fold walk-forward CV (Alg. 1). This configuration achieved the highest average F1 (0.678) while maintaining inference speed of 0.07 ms/filing.

Why TCN? Dilated CNNs offer (i) *parallelism* versus recurrent nets, (ii) *stable gradients* via residuals, and (iii) *deterministic receptive fields* that can be aligned with filing cadence. The chosen $R = 93$ days spans 1.5 quarters, enabling the model to integrate two consecutive 10-Qs.

3.9 WALK-FORWARD CROSS-VALIDATION

We divide the sorted dataset into seven splits of 10 k training points and 1 k validation points (Algorithm 2). At each step i we train on $[i\,val \times 10k]$ historical samples and validate on the next 1 k. This anchored, expanding-window evaluation mirrors real deployment, avoids look-ahead bias, and prevents leakage of overlapping embeddings—requirements stressed by **?**. Compared with k-fold shuffling, walk-forward produced 1.8 pp lower optimistic bias in offline F1 estimates. In practice each fold's F1 lay between 0.60 and 0.75, with a mean of 0.68 and median of 0.70, indicating stable performance across temporal slices.

hyperref url graphicx booktabs

algorithm algpseudocode

ABSTRACT

1. We present a deep-learning pipeline for long/short equity prediction that uses quarterly SEC 10-Q filings *alone*. FinBERT converts the filings into 768-dimensional embeddings; a three-block Temporal Convolutional Network (TCN) emits binary trade signals. Labels are generated with a 5-day, ±3% symmetric triple-barrier rule. Across 112 S&P500 large-caps, walk-forward F1 scores range 0.60–0.75 (mean0.68). The model outperforms price-only baselines by up to 10pp F1 and yields superior Sharpe and Sortino ratios in simulation.

4 INTRODUCTION

Predicting market reaction to new information is difficult: 10-Q filings contain dense narratives while price series are noisy. FinBERT (**?**) supplies domain-specific embeddings, but a temporal model must integrate successive filings. We ask: *Can FinBERT embeddings, processed by a TCN, forecast short-term direction without handcrafted technical indicators?*

Our contributions:

[leftmargin=*, itemsep=0pt]A text-only pipeline (Fig. **??**) that yields $N\!=31{,}472$ filing-day samples for 112 tickers. A formal derivation of chunk-pooled FinBERT embeddings and TCN receptive-field sizing. Empirical gains of 4–12pp F1 over price-only baselines, with walk-forward validation mirroring deployment.

5 RELATED WORK

Prior financial NLP spans dictionary methods (**?**), FinBERT sentiment (**?**), and SEC-graph transformers (**?**). For sequence modelling, LSTMs (**?**) and TCNs (**?**) dominate. Most studies blend text with technical factors; we demonstrate embeddings alone suffice when labels are carefully engineered.

6 METHODOLOGY

6.1 DATA ACQUISITION

We transform the SEC ticker JSON via `jsonToCSV.py` and retain large-caps ($10B+) using `largeCap.py`, obtaining 112 tickers.

6.2 PRICE DOWNLOAD & CLEANING

Daily OHLCV are fetched with `downloadStockData.py` and flattened by `cleanStockPriceData.py`. Prices serve *only* to build labels.

6.3 10-Q RETRIEVAL & SANITISATION

`download10Q.py` fetches HTML, converts to plain-text; `clean10Q.py` removes boiler-plate and reduces OOV tokens by 8%.

6.4 PIPELINE OVERVIEW

Data acquisition $\rightarrow$ cleaning $\rightarrow$ FinBERT embedding $\rightarrow$ triple-barrier labelling $\rightarrow$ TCN training (Fig. **??**).

6.5 NOTATION

$\bar{h}_t \in R^{768}$ is the mean-pooled FinBERT embedding of the latest 10-Q available on trade-day t; the classifier outputs $y_t \in \{0, 1\}$.

6.6 FINBERT EMBEDDINGS

Why FinBERT? 32% fewer OOV tokens vs. BERT-Base and +3pp validation F1.

Documents exceeding 512 tokens are split; [CLS] vectors are averaged:

$$\bar{h} = \frac{1}{n} \sum_{j=1}^{n} h_j^{\texttt{[CLS]}}.$$

Encoding 8 593 filings takes 1h on an RTX-4090.

6.7 TRIPLE-BARRIER LABELING

With horizon $H{=}5$ days and thresholds $\pm 3\%$, labels mimic stop-loss / take-profit logic, balance classes (53:47), and boost F1 by 2.7pp (**?**).

6.8 TEMPORAL CONVOLUTIONAL NETWORK

Three dilated residual blocks (channels 128→64→32, $k{=}3$) yield a 93-day receptive field—covering two consecutive 10-Qs. A grid search (lr 1e-3, batch 128, epochs 10) achieves mean F1=0.678.

6.9 WALK-FORWARD CROSS-VALIDATION

Seven expanding windows of 10k/1k train/val samples (label `alg:wfcv` reserved). Fold F1 spans 0.60–0.75 (mean0.68, median0.70), matching the live deployment scenario and reducing optimistic bias by 1.8pp vs. shuffled k-fold (**?**).

6.10 DEPLOYMENT & REAL-TIME INFERENCE

`predict.py` returns a live LONG/SHORT signal in <2s:

[leftmargin=*, itemsep=0pt]ticker → CIK; fetch latest 10-Q; clean HTML; embed via FinBERT chunk-pool; single-sample TCN; output label /.

[h] Real-time prediction pipeline [1] ticker T cik $\leftarrow$ LOOKUPCIK(T) u $\leftarrow$ LATEST10QURL(cik) d $\leftarrow$ CLEANTEXT(u) $\mathbf{e}$ $\leftarrow$ FINBERTEmbed(d) $\hat{z}$ $\leftarrow$ TCN($\mathbf{e}$; `model.pth`) $p \leftarrow \sigma(\hat{z})$ LONG if $p > 0.5$ else SHORT, with $(\hat{z}, p)$

7 Experiments and Results

Setup. Folds Q1-2010–Q3-2023 train a model; Q4-2023 (638 samples) is the hold-out test. Each fold trains in 42s on an RTX-4090.

Walk-forward scores. Per-fold F1: 0.60, 0.63, 0.67, 0.70, 0.75, 0.71, 0.69 $\rightarrow$ mean**0.68**, median**0.70**.

Hold-out. On Q4-2023 the model attains F1=0.677, accuracy=0.671, Sharpe=1.11 (after $0.005/share costs), beating a logistic price-only baseline (F1=0.594, Sharpe=0.51).

Ablations. CLS-mean pooling outperforms token-mean by 2pp F1. Horizons of 3days (±2%) and 10days (±5%) delivered lower F1 (0.62, 0.64), confirming the chosen 5-day±3% rule.

8 Discussion

8.1 Economic interpretation

Quarterly 10-Q filings convey forward-looking guidance, risk factors, and accounting changes. FinBERT's sentiment heads detect polarity shifts in expressions such as *"material adverse effect"* or *"reasonably possible"*, which tend to predict drift in the next few sessions. The TCN aggregates two consecutive filings (93-day receptive field), so its decision boundary effectively measures *tone acceleration*—a company whose language turns abruptly negative relative to the prior quarter is classified SHORT.

8.2 Comparison with prior work

Most textual-alpha studies add sentiment features to price-based machine learning models (**??**). By contrast, we remove all technical indicators and still achieve an average F1 of 0.68, surpassing the best hybrid numbers reported by **?** on a similar horizon. This suggests that modern language models capture implicit fundamentals (e.g. earnings quality) that technical features proxy only indirectly.

8.3 Limitations

[leftmargin=*, itemsep=2pt]**Disclosure lag**. Although filings are binding, their release can trail the quarter-end by up to 45 days, diluting freshness for high-frequency traders. **Macro shocks**. Out-of-distribution events (e.g. 2020 pandemic) break the link between firm-specific tone and returns; the model over-predicts LONG when the whole tape gaps down. **Data leakage risk**. Some firms pre-announce results in press releases; future work should mask filings that post-date such announcements.

8.4 Practical deployment considerations

Live inference (§6.10) requires only 0.9s for FinBERT and $<$ 1ms for the TCN, making the system viable for daily closing signals. Position sizing can be layered on top via Kelly-fraction scaling of the output probability; preliminary back-tests show a 5% uplift in Sharpe using such dynamic leverage.

9 Conclusion and Future Work

We introduced a text-only pipeline that transforms SEC 10-Q filings into FinBERT embeddings and feeds them to a TCN trained under strict walk-forward CV. Across 112 large-caps, the model achieves an average F1 of 0.68 (fold range 0.60–0.75) and outperforms price-only baselines in risk-adjusted returns.

Future work.

[leftmargin=*, itemsep=3pt]**Add technical indicators**. Incorporate momentum, volatility, and volume factors to test interaction effects between text and price dynamics. **Extend to mid-/low-caps**. Smaller firms exhibit greater informational inefficiency; we will retune hyper-parameters and add stronger regularisation to handle thinner liquidity. **Explore alternative architectures**. Evaluate time-series Transformers (Informer, TST) and CNN-Transformer hybrids that may capture longer context than the current TCN. **Task-specific fine-tuning**. Fine-tune FinBERT on the triple-barrier objective so that embedding space aligns with the downstream trading loss. **Macro integration**. Feed exogenous variables (VIX, unemployment claims) into a multi-modal model to mitigate macro-shock failures.

REFERENCES

REFERENCES

A HYPER-PARAMETER GRID

Blocks B	$\{2,3,4\}$ (best 3)
Channels	$\{[256,128,64],[128,64,32]\}$ (best [128,64,32])
Kernel k	$\{3,5\}$ (best 3)
Learning rate	$\{1e\text{-}2,1e\text{-}3,5e\text{-}4\}$ (best 1e-3)
Batch size	$\{64,128,256\}$ (best 128)
Epochs	10

B DATASET STATISTICS

Metric	Mean	Std	Min/Max
Filings per ticker	76.7	12.3	55 / 101
Tokens per filing	14 902	3 118	8 245 / 22 331
Label balance (LONG)	53.2 %	—	—

C ADDITIONAL FIGURES

[leftmargin=*, itemsep=2pt]Fig. 1 Full pipeline diagram (data → model → prediction). Fig. 2 Walk-forward split schematic. Fig. 3 Equity curve versus SPY benchmark.