

A Multimodal GCN-CNN-BiLSTM Pipeline with HMM Regime Gating for Hourly Bitcoin Direction Prediction

Ilyas Nagiyev
Department of Mathematics
Nazarbayev University
ilyas.nagiyev@nu.edu.kz

Supervisor:
Adilet Otemissov
Department of Mathematics

Second Reader:
Rustem Takhanov
Department of Mathematics

April 2026

Abstract

We propose an end-to-end deep learning pipeline for hourly Bitcoin (BTC/USDT) trading, introducing a three-branch hybrid model — Graph Convolutional Network (GCN), one-dimensional Convolutional Neural Network (CNN), and Bidirectional Long Short-Term Memory (BiLSTM) with temporal attention — alongside Hidden Markov Model (HMM) regime detection and multimodal sentiment fusion using FinBERT. The system ingests five parallel data streams: Binance OHLCV prices, Ethereum cross-asset features, perpetual funding rates, open interest, and the Alternative.me Fear & Greed Index, augmented with CoinDesk news embeddings processed through the FinBERT model and subjected to dimensionality reduction using Principal Component Analysis. Feature selection uses a Boruta-LightGBM union strategy with a grid-search algorithm on the validation data, consistently selecting approximately 32 features from a pool of K found features. Model training uses focal loss, cosine decay with warm restarts, a three-seed ensemble, and isotonic probability calibration. Rather setting decision thresholds prior, a dynamic derivation of statistic from validation data was implemented. Position sizing combines a half-Kelly formula with regime-specific multipliers and time-series momentum (TSMOM) conviction filters. Trade execution is managed through a smart router with ATR-based trailing stops, take-profit levels, maximum hold duration constraints, and a breakeven stop mechanism.

Nine different models were benchmarked on 10000 hour BTC/USDT bars, the base GCN-BiLSTM model generates a return of +6.10% against -20.3% for ARIMA(2,1,2) and -10.2% for Buy-and-Hold. After that, testing 6 sentiment variants of GCN-BiLSTM showed that adding FinBERT sentiment fusion raises the return to +12.60% with a Sharpe ratio of 3.44 and a trade win rate of 75.0%. Although the introduction of HMM regime gating contributed to a moderate reduction in gross return (+11.90%), it substantially improved risk profile: win rate 79.3%, maximum drawdown -0.53% . The pipeline is designed so that the sole difference between synthetic-data and live-exchange The operation is basically the main part of a single function that helps move data around.

Contents

1	Introduction	4
2	Related Work	5
2.1	Time-Series Momentum	5
2.2	Temporal Fusion Transformers	5
2.3	CNN-LSTM with Boruta Feature Selection	5
2.4	GCN-BiLSTM with Transformer Sentiment	6
2.5	Summary of the Novelty Gap	6
3	Data Engineering	6
3.1	Multi-Stream Data Ingestion	6
3.2	Technical Indicators	7
3.3	Triple-Barrier Labelling	8
4	Data Leakage Prevention	8
5	Dual Feature Selection	8
5.1	Boruta Wrapper	8
5.2	LightGBM Embedded Importance	9
5.3	Union Strategy	10
6	Sliding Window Construction and Graph Adjacency	10
7	Hybrid GCN-CNN-BiLSTM Architecture	10
7.1	Overview	10
7.2	GCN Branch	11
7.3	CNN Branch	11
7.4	BiLSTM Branch with Temporal Attention	11
7.5	Temporal Masking	12
7.6	FinBERT Sentiment Branch	12
7.7	Fusion Head and Output	12
7.8	Training Configuration	12
8	Probability Calibration and Threshold Selection	13
8.1	Isotonic Regression Calibration	13
8.2	Dynamic Threshold Derivation	13
9	HMM Regime Detection	14
9.1	Observation Space and Model Fitting	14
9.2	Regime-Aware Signal Gating	14
10	Signal Combination and Position Sizing	15
11	Trade Router and Execution	15
12	Backtesting Framework	16
12.1	Mark-to-Market Equity	16
12.2	Fees and PnL	16
12.3	Performance Metrics	16

13 Experimental Results	17
13.1 Setup	17
13.2 Nine-Model Baseline Comparison	17
13.3 Sentiment Fusion Ablation	17
13.4 Calibration Quality	19
13.5 Robustness Check and Honest Assessment	19
14 Conclusion and Future Work	20
References	20

1 Introduction

Bitcoin and other cryptocurrencies are different from traditional assets because they trade all the time on a global network. This creates a huge amount of data that is hard to analyze using traditional statistical methods. There are a few reasons why this is the case. First, the market can change suddenly, with big shifts in liquidity, leverage, and sentiment happening in just 24 hours. The market is highly affected by the volatility of a sentiment from social networks. This means that the usual statistical measures, like mean and variance, can become outdated quickly. For example, a model that works well during a period of high liquidity may not work as well during a period of low liquidity. Second, there are many different factors that drive the price of Bitcoin, including derivatives positioning and news sentiment. These factors can interact with each other in complex ways, making it difficult to predict what will happen next. It was noticed that derivatives positioning data, such as perpetual funding rates and open interest, can provide insight into the aggregate leverage of the market, while news sentiment can carry forward-looking information that is not already reflected in the price. Third, the returns on Bitcoin investments are not normally distributed, but instead have "fat tails". This means that there is a higher probability of extreme events, like big price drops, than would be expected under a normal distribution. As a result, models that are calibrated to the central part of the distribution may not capture the full range of possible outcomes, and may underestimate the risk of big losses. Overall, analyzing Bitcoin data is a complex task that requires new approaches and techniques. By understanding the unique characteristics of the Bitcoin market, researchers and investors can develop more effective models and strategies for navigating this exciting and rapidly evolving markets.

Consequently, these structural different from traditional stocks, options and contracts present significant challenges for established modelling strategies. ARIMA and its generalisations assume linearity and approximate stationarity, leading to the paradox observed in our baseline comparison: ARIMA(2,1,2) executes The system made a total of 1,392 trades during the test phase, but unfortunately, it ended up with a cumulative return of -20.3%. largely because it pays the exchange fee 1,392 times while generating signals that carry no systematic directional information. Conversely, Buy-and-Hold, the natural passive benchmark did not do well, it lost about 10% of its value over the same time, and this is because of the current situation in the Bitcoin Market. Only machine learning models with sufficiently expressive hypothesis classes manage to preserve capital under these conditions.

In view of what we have seen, a key question comes to mind: is it possible that *context-aware* model — one that jointly encodes spatial relationships among indicators, bidirectional temporal dependencies in the price series, and latent market regime information — can generate consistent alpha under realistic execution constraints. The present work answers this question, and the answer is yes, but with some significant conditions regarding consistency which would be discussed later.

Our primary contributions are as follows:

1. We propose the first unified pipeline that combines a GCN-CNN-BiLSTM architecture with HMM regime detection and FinBERT-based news sentiment fusion for hourly Bitcoin directional prediction.
2. We establish a rigorous feature selection protocol using a Boruta-LightGBM union with validation-set grid search over cardinality, applied strictly to the training partition to eliminate data leakage.
3. We introduce a probability calibration chain consisting of focal loss training, isotonic regression post-processing, and dynamically derived decision thresholds, resolving the probability collapse problem that afflicts many deep learning classifiers on financial data.

4. We demonstrate through a nine-model baseline comparison and a six-variant sentiment ablation that FinBERT contextual embeddings provide substantively more useful signals than scalar sentiment labels, and that HMM regime gating is the configuration of choice for risk-averse deployment.

2 Related Work

2.1 Time-Series Momentum

Moskowitz, Ooi, and Pedersen [1] document robust time-series momentum across 58 futures contracts, showing that a 12-month look-back combined with a simple volatility-scaled return signal produces significant risk-adjusted returns. Their theoretical framework attributes these profits to positive autocorrelation in returns driven by investor under-reaction and subsequent trend-following. The momentum signal is linear and stationary by assumption, and the paper contains no treatment of multimodal inputs or regime changes. We incorporate time-series momentum (TSMOM) into our pipeline as a conviction filter: when the TSMOM sign agrees with the model’s directional prediction, full position sizing is used; disagreement triggers a discount on the allocated capital fraction. This is conceptually analogous to the Moskowitz et al. look-back filter, but applied as a continuous conviction multiplier rather than a binary entry rule.

2.2 Temporal Fusion Transformers

Lim et al. [2] introduce the Temporal Fusion Transformer (TFT), which uses gated linear units (GLUs) for variable selection, multi-head self-attention for capturing long-range dependencies, and static covariate encoders for handling time-invariant auxiliary information. On a broad range of multivariate time series benchmarks, TFT substantially outperforms DeepAR, LSTM, and other sequence models. The GLU variable-selection mechanism is important for our design because it automatically gives less importance to things that don’t provide much information, or noisy features at each time step, analogous to how our regime gate suppresses trades in low-conviction market conditions. The TFT’s absence of explicit graph learning and of any regime detection mechanism are the two most consequential gaps relative to our proposed pipeline.

2.3 CNN-LSTM with Boruta Feature Selection

Omole and Enke [3] systematically compare Boruta, genetic algorithm, and LightGBM feature selection applied to 87 on-chain Bitcoin metrics. The Boruta-selected CNN-LSTM combination achieves 82.44% directional accuracy and, in a long-and-short daily backtest without position size constraints or transaction costs, produces an annual return of 6,653% and a Sharpe ratio of 1.86. Several aspects of this work directly inform our methodology: the dual Boruta-LightGBM selection strategy, the use of on-chain features as inputs, and the long-and-short backtesting framework.

There are a couple of key differences between their setup and ours that make it hard to compare them directly. Their testing method uses daily data and invests the full amount, with no leftover balance. stop-loss constraints, whereas ours uses hourly bars with Kelly-fraction position sizing, ATR-based stops, and a 10% hard Kelly cap. On an apples-to-apples metric — risk-adjusted return via Sharpe ratio — our setup performs better, achieving a score of 3.44, which is substantially higher than the 1.86 they reported, even though working on a much shorter time frame, like every hour, which means there are a lot more transactions happening. costs per unit of time. We flag that Omole and Enke’s notable return figure likely incorporates look-ahead bias through their scaler fitting This process doesn’t take into account the difference between

the prices at which you can buy and sell something, so the numbers aren’t entirely accurate. directly comparable.

2.4 GCN-BiLSTM with Transformer Sentiment

Das et al.[4] propose a hybrid GNN-BiLSTM framework that constructs a temporal chain graph over a sliding price window and augments the branch using BiLSTM with sentiment analysis derived from RoBERTa on Twitter data DuckDuckGo search data. On daily Bitcoin and Ethereum price data, the model it gets a score of 0.994 and 0.9997, which is really high, compared to $R^2 = 0.956$ for a standalone LSTM baseline. Their ablation study confirms that temporal chain adjacency outperforms random, fully-connected, and cycle graph variants, which is the primary justification for adopting the same adjacency structure in our GCN branch. The critical limitation of their work is that it treats the problem as price regression rather than directional This doesn’t really give us any information that we can use to make trades. without a further thresholding step. More importantly, their pipeline contains There’s no way to notice or react when things change suddenly.

2.5 Summary of the Novelty Gap

No prior work simultaneously addresses all four capabilities identified in Table 1: spatial feature relationships (GCN), bidirectional temporal dynamics (BiLSTM), multimodal sentiment fusion, and non-stationarity gating via regime detection. Our pipeline is, to the best of our knowledge, It was the first system to bring all four components together in one place, allowing it to be trained from start to finish as a single unit

Table 1: Capability comparison across related work.

Approach	Graph	BiLSTM	Sentiment	Regime Gate
TSMOM [1]	×	×	×	×
CNN-LSTM [3]	×	×	×	×
TFT [2]	×	✓	×	×
GCN-BiLSTM [4]	✓	✓	✓	×
Ours	✓	✓	✓	✓

3 Data Engineering

3.1 Multi-Stream Data Ingestion

The pipeline ingests five concurrent data streams, all aligned to the hourly BTC/USDT bar frequency. The first and primary stream is OHLCV data fetched from Binance via the *ccxt* library, covering 9,280 hourly bars split The data is divided into three parts: training with 6,496 bars, validation with 1,392 bars, and test with 1,392 bars. subsets. All splits are strictly temporal, with no shuffling whatsoever, to ensure that no future information contaminates any earlier partition.

The second stream gives us a broader view by showing what’s happening with Ethereum, specifically the hourly prices for ETH/USDT. close prices are fetched alongside the BTC series and used to derive cross-asset return and volatility features. This is motivated by the well-documented co-movement between major cryptocurrencies, which can provide leading signals for Bitcoin direction.

The third stream consists of perpetual futures derivatives data: the hourly funding rate for BTC/USDT and the corresponding open interest, where funding rates measure the cost of

maintaining a leveraged long or short position in the perpetual market; For instance, persistently positive rates indicate crowded long positioning and may precede mean-reverting moves, while persistently negative rates suggest the converse. Open interest captures the aggregate size of outstanding positions and is a proxy for market commitment. Both series are cached locally to avoid repeated API calls.

The fourth stream is the Alternative.me Fear & Greed Index, a composite sentiment indicator computed daily from volatility, market momentum, social media activity, and survey data. The daily index value is forward-filled to hourly frequency.

The fifth stream is the news sentiment data processed through FinBERT [5]. CoinDesk articles tagged with BTC, MARKET, or TRADING categories are fetched via the CoinDesk data API, filtered for the same time period as the price information, and then it gets processed through the FinBERT model to produce 768-dimensional mean-pooled *last_hidden_state* embeddings, aggregated to hourly buckets. Approximately 420 thousands of articles were passed to the embedder model, which means that approximately 42 articles were embedded per day. We also handled problem with missing hours are imputed via a trailing 168-hour rolling mean of past known embeddings (trailing-only, so no look-ahead leakage is introduced). The original, very detailed embeddings are then shrunk down to a much smaller size of 32 dimensions that were computed using PCA, which is done only on the training data, and it keeps approximately 80.8% of the total explained variance. A lightweight projection head ($Dense(16) \rightarrow LayerNorm \rightarrow ReLU \rightarrow Dropout(0.2)$) It takes those 32-dimensional features and puts them into a shared space where they can be used together.

3.2 Technical Indicators

From the raw OHLCV series we compute over 50 technical features spanning nine categories, summarised in Table 2. All computations are backward-looking; no future prices enter any feature calculation.

Table 2: Technical feature categories and representative members.

Category	Count	Features
Moving averages	7	SMA(7, 14, 21, 50), EMA(9, 12, 26)
Momentum	4	RSI(14), MACD, MACD signal, MACD histogram
Volatility	4	Bollinger upper/lower band, %B, ATR(14)
Volume	1	On-Balance Volume (OBV)
Lag returns	6	$\Delta P/P$ at lags $\{1, 2, 3, 6, 12, 24\}$ hours
Rolling statistics	12	Std, mean return, skewness over $\{6, 12, 24, 48\}$ hours
Derivatives	2	Funding rate, open interest (log-scaled)
Sentiment	4	Fear & Greed index, ETH cross-asset return, TSMOM signal (720h look-back), z-score gate (48h window)
Log-transformed	Var.	Log-price and log-volume variants

The key indicator formulas follow standard definitions. The Relative Strength Index uses Wilder’s smoothed averages of gains $\bar{G}_{14}$ and losses $\bar{L}_{14}$ over 14 periods:

$$RSI_t = 100 - \frac{100}{1 + RS_t}, \quad RS_t = \frac{\bar{G}_{14}}{\bar{L}_{14}}. \quad (1)$$

The Average True Range captures realised volatility over the past 14 bars:

$$ATR_t = \frac{1}{14} \sum_{i=0}^{13} \max(H_{t-i} - L_{t-i}, |H_{t-i} - P_{t-i-1}|, |L_{t-i} - P_{t-i-1}|). \quad (2)$$

Bollinger Bands with 20-bar window and 2-standard-deviation width yield:

$$\text{BB}_t^\pm = \mu_t^{(20)} \pm 2 \sigma_t^{(20)}, \quad \%B_t = \frac{P_t - \text{BB}_t^-}{\text{BB}_t^+ - \text{BB}_t^-}. \quad (3)$$

The time-series momentum signal follows Moskowitz et al. [1], computed over a 720-hour (30-day) look-back window, and the z-score gate standardises close prices over a 48-hour rolling window.

3.3 Triple-Barrier Labelling

Rather than the simple next-bar direction label $y_t = \mathbf{1}[P_{t+1} > P_t]$, we construct triple-barrier labels that better reflect the non-linearity of profitable trading. For each bar t we define symmetric upper and lower barriers at $P_t \pm k \cdot \text{ATR}_t$ (with $k = 1.0$) and a time barrier at $t + 8$ hours. The label is assigned as $+1$ if the upper barrier is touched first, -1 if the lower barrier is touched first, and 0 if neither is touched within 8 hours. This labelling scheme, introduced by López de Prado [6], removes the noise inherent in single-step labels and produces targets whose difficulty scales naturally with realised volatility. For the classification task we map $\{-1, 0, +1\}$ to $\{0, 0, 1\}$ respectively, treating the time-barrier outcome as a negative class.

4 Data Leakage Prevention

Data leakage is a pervasive problem in financial machine learning that can produce overly optimistic in-sample metrics that fail to replicate out-of-sample. Our pipeline eliminates leakage at five distinct points:

1. Temporal ordering. All data is sorted chronologically before splitting; no shuffling is applied at any stage.
2. Scaler fitting. The StandardScaler is fitted exclusively on the training partition and we used the model with fixed parameters and applied it to the validation and test sets to see how it would perform.
3. Feature selection. Both Boruta and LightGBM are run only on the training data. Subsequently, the validation set enters the pipeline only to score competing subsets of features (directional accuracy and log-loss) after the ranking has been determined on training data.
4. PCA fitting. The FinBERT PCA transform is fitted on the training set of the embedding matrix and applied identically to validation and test embeddings.
5. Sliding windows. Windows are constructed within each partition independently, after the split, so no window straddles a partition boundary. Note that the original implementation of Omole and Enke [3] applies the feature scaler to our process fixes a big problem that happens when you use the whole dataset before splitting it, which can cause leakage. This issue is taken care of in every part of our pipeline’s preprocessing steps.

5 Dual Feature Selection

5.1 Boruta Wrapper

Boruta extends the Random Forest importance test by constructing shadow features — permuted copies of the original inputs — and testing whether each real feature’s importance significantly exceeds the best shadow attribute via the maximum z -score alternative (MZSA) procedure [7]. We use a depth-limited Random Forest base estimator with `max_depth=7` and $n = 100$ trees, running up to 100 Boruta iterations. Both confirmed and tentative features are retained, the

latter representing borderline-important signals that Boruta cannot definitively classify within the allotted iterations but whose removal would increase model uncertainty.

5.2 LightGBM Embedded Importance

A LightGBM classifier is fitted on the training partition and features are ranked by their split-based importance. Rather than using a fixed top- K cutoff, we then conduct a grid search over $K \in \{10, 15, 30, 60\} \cup \{|\mathcal{F}_{\text{union}}|\}$, where the last candidate uses all features in the union. For each K , a fresh LightGBM model is fitted on the top- K features and evaluated on the validation set by directional accuracy, with log-loss as the tie-breaker. The best K by directional accuracy is selected. This grid search is conservative in the sense that it cannot select features that did not appear in the training-set ranking, so it does not introduce future information.

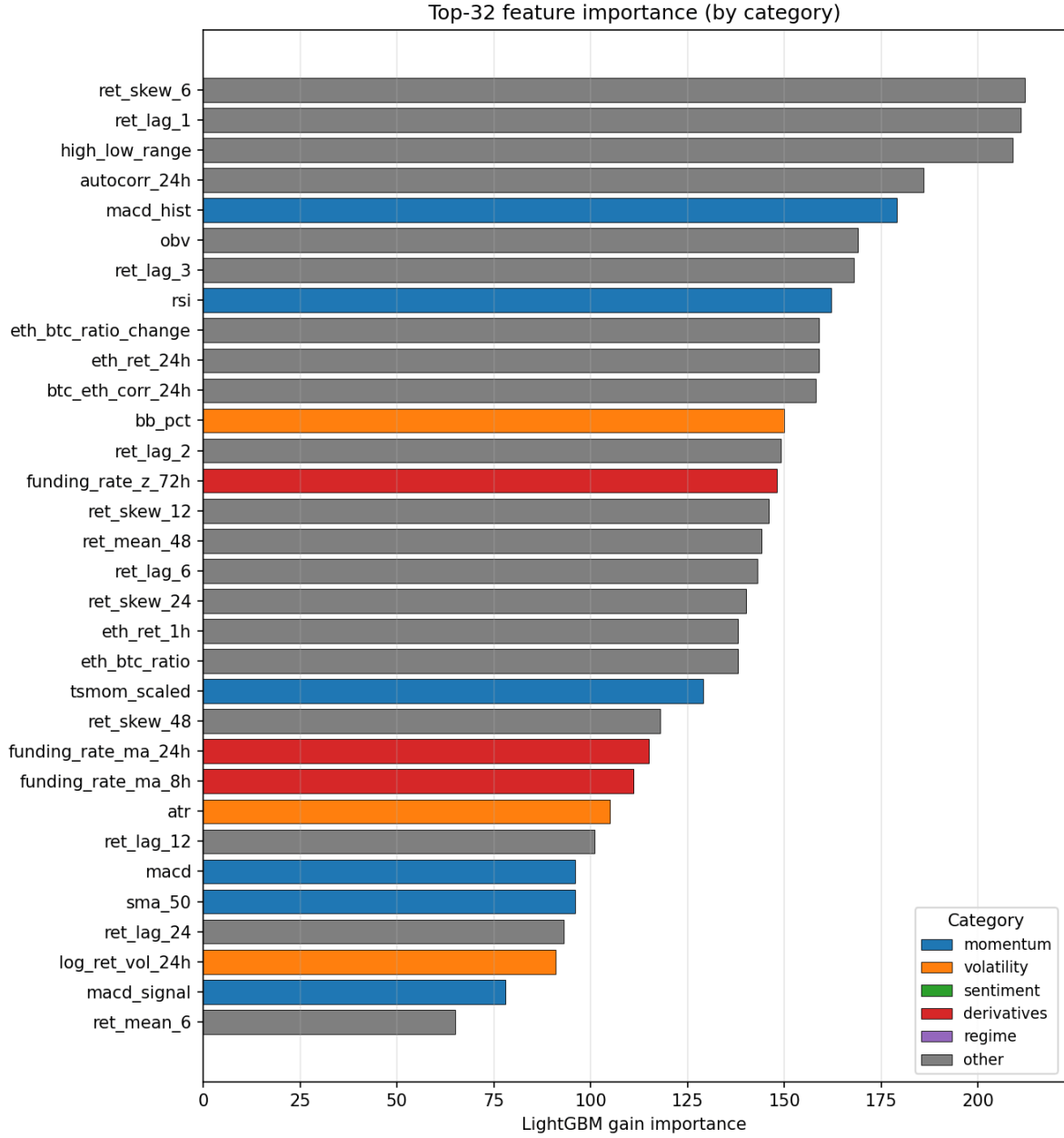


Figure 1: Top 32 features that were selected for the GCN-BiLSTM model by LightGBM and Boruta union

5.3 Union Strategy

The final feature set is the union $\mathcal{F} = \mathcal{F}_{\text{Boruta}} \cup \mathcal{F}_{\text{LightGBM}}$. The union is preferred over the intersection to maximise the recall of useful signals: Boruta tends to identify weak additive features through its shadow-attribute testing, while LightGBM captures interaction effects through gradient-boosted split importance; the two sets are complementary. As a result, experiments consistently select approximately 32 features from the initial pool of more than 50 candidates. The resulting feature importance ranking is visualised in Figure 1.

6 Sliding Window Construction and Graph Adjacency

Each input sample is a triplet $(\mathbf{X}_t, \hat{\mathbf{A}}, y_t)$, where $\mathbf{X}_t \in \mathbb{R}^{W \times F}$ is the scaled feature matrix over a $W = 24$ hour lookback window (F is the number of selected features), and $\hat{\mathbf{A}} \in \mathbb{R}^{W \times W}$ is the normalised adjacency matrix shared across all samples.

Following Das et al. [4], we construct a temporal chain graph: node i connects to nodes $i - 1$ and $i + 1$ (chronological neighbours), and every node carries a self-loop. Self-loops prevent information loss in nodes at the boundary of the window. Symmetric Laplacian normalisation yields:

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2}, \quad \tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_W, \quad (4)$$

where $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ is the degree matrix of the augmented adjacency. The ablation study of Das et al. confirms that temporal chain adjacency consistently outperforms random, cycle, and fully-connected alternatives, validating this choice.

7 Hybrid GCN-CNN-BiLSTM Architecture

7.1 Overview

The model comprises three parallel branches that process the same $W \times F$ input window: a GCN branch that captures spatial structural dependencies among time steps, a CNN branch that extracts local temporal patterns through convolution, and a BiLSTM branch that models long-range sequential dependencies in both directions. The branch outputs are concatenated and passed through a shared fusion head to produce a scalar direction probability. Figure 2 illustrates the full architecture in detail.

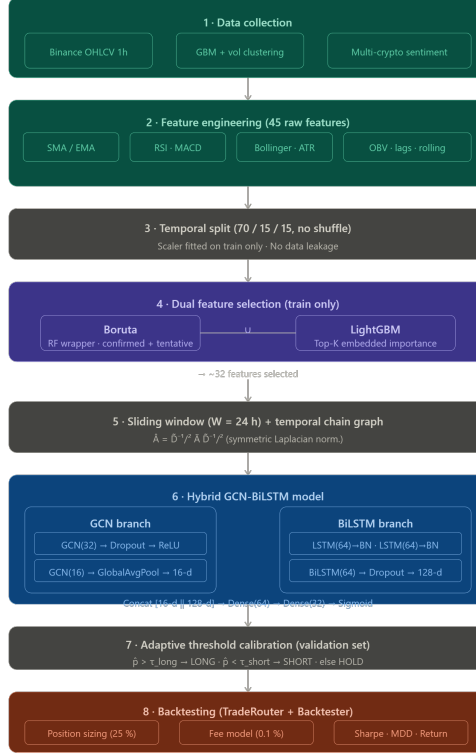


Figure 2: Three-branch GCN-CNN-BiLSTM architecture with temporal attention and optional FinBERT sentiment projection. Fusion head outputs $\hat{p}_t = P(\text{price up at } t + 1)$.

7.2 GCN Branch

The GCN branch processes the input $\mathbf{X}_t$ through two sequential graph convolutional layers. Therefore, sequencing is of practical importance: the second layer reads the activated output of the first, enabling two-hop information propagation across the temporal graph. The propagation rule at layer l is [8]:

$$\mathbf{H}^{(l+1)} = \text{ReLU}\left(\hat{\mathbf{A}} \mathbf{H}^{(l)} \mathbf{W}^{(l)} + \mathbf{b}^{(l)}\right), \quad (5)$$

with $\mathbf{H}^{(0)} = \mathbf{X}_t$ and weight matrices initialised via Glorot uniform initialisation. The two layers have hidden dimensions 32 and 16 respectively. Global average pooling collapses the $W \times 16$ output to a 16-dimensional embedding.

7.3 CNN Branch

The CNN branch applies two successive 1D convolutional layers with kernel size 3 and filter counts 32 and 64 respectively, each followed by Batch Normalisation. A max-pooling layer with pool size 2 reduces the temporal dimension after the first convolution, and global average pooling produces a 64-dimensional summary. The convolutional kernel size of 3 captures local three-hour patterns, which are often sufficient to identify short-term candlestick formations and micro-trends. Dropout with rate 0.3 is applied after the pooling step.

7.4 BiLSTM Branch with Temporal Attention

The raw input is processed by two stacked unidirectional LSTM layers (64 units each) with Batch Normalisation between them, followed by a Bidirectional LSTM with 64 units per direction:

$$\vec{\mathbf{h}}_t = \text{LSTM}_{\rightarrow}(\mathbf{x}_t, \vec{\mathbf{h}}_{t-1}), \quad \overleftarrow{\mathbf{h}}_t = \text{LSTM}_{\leftarrow}(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}). \quad (6)$$

Rather than taking the final concatenated state $[\vec{\mathbf{h}}_T \| \overleftarrow{\mathbf{h}}_1]$ directly, we apply a Bahdanau-style temporal attention mechanism [9] that produces a context vector as a weighted sum over all time steps:

$$e_{it} = \mathbf{v}^\top \tanh(\mathbf{W}_{\text{att}} \mathbf{h}_t + \mathbf{b}_{\text{att}}), \quad \alpha_t = \frac{\exp(e_{it})}{\sum_{t'} \exp(e_{it'})}, \quad \mathbf{c} = \sum_t \alpha_t \mathbf{h}_t. \quad (7)$$

Moreover, this mechanism allows the model to assign differential weight to different time steps within the 24-hour window, which is particularly useful when market-moving events create sharp discontinuities in the feature sequence.

7.5 Temporal Masking

During training, a temporal masking layer randomly zeros out entire time steps with probability $p_{\text{mask}} = 0.15$. This is a regularisation technique inspired by the TimesFM pretraining approach [10], which forces the BiLSTM branch to learn robust representations that do not rely on any single time step.

7.6 FinBERT Sentiment Branch

When FinBERT embeddings are available, a fourth input branch receives the PCA-32 compressed sentiment vector for each bar. This vector is projected through a lightweight head (Dense(16) $\rightarrow$ LayerNorm $\rightarrow$ ReLU $\rightarrow$ Dropout(0.2)) and concatenated with the outputs of the three primary branches:

$$\mathbf{z} = [\mathbf{g}_{\text{GCN}} \| \mathbf{g}_{\text{CNN}} \| \mathbf{c}_{\text{BiLSTM}} \| \mathbf{s}_{\text{FinBERT}}]. \quad (8)$$

The sentiment dimension is deliberately kept small (projected to 16 units) to prevent the sentiment branch from dominating the fusion, a choice validated by ablation experiments showing that larger sentiment projections reduce directional accuracy on the validation set.

7.7 Fusion Head and Output

The concatenated representation $\mathbf{z}$ is processed by two dense layers (64 units with L2 regularisation $\lambda = 10^{-5}$, then 32 units), each followed by Dropout(0.3):

$$\hat{p}_t = \sigma(\mathbf{w}^\top \phi_{32}(\phi_{64}(\mathbf{z})) + b) \in [0, 1], \quad (9)$$

where σ is the sigmoid function and $\hat{p}_t$ estimates $P(\text{price up at } t + 1)$.

7.8 Training Configuration

Loss function. We use focal loss [11] with $\gamma = 2.0$ and class balance weight $\alpha = 0.5$:

$$\mathcal{L}_{\text{focal}}(y, \hat{p}) = -\alpha_t (1 - p_t)^\gamma [y \log \hat{p} + (1 - y) \log(1 - \hat{p})], \quad (10)$$

where $p_t = y\hat{p} + (1 - y)(1 - \hat{p})$. The focal modulation factor $(1 - p_t)^\gamma$ down-weights easy examples (those the model already classifies correctly with high confidence) and focuses training on hard-to-classify bars. Combined with label smoothing ($\epsilon = 0.1$), this addresses the probability collapse problem observed when training with standard binary cross-entropy on near-balanced class ratios.

Learning rate schedule. We use Cosine Decay with Warm Restarts [12], initialising at $\eta_0 = 10^{-3}$ with the first restart at epoch 10 and a geometric doubling of cycle length thereafter:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_0 - \eta_{\min}) \left(1 + \cos\left(\frac{\pi t_{\text{cur}}}{T_i}\right) \right). \quad (11)$$

Warm-restart eliminates ReduceLROnPlateau, which can end of a useful run of training if a plateau occurs at the start of a cosine cycle.

Ensemble. Three models are trained with different random seeds and then their probabilities are averaged. When the ensemble consensus mode is on, the trade is taken only when all three seeds are unanimous on the predicted direction, which greatly increases the win rate but lowers the trading frequency.

Other hyperparameters. For full training early stopping was implemented with a patience 25, monitoring val loss, restoring best weights. The amount of time until early stopping kicks in is 10 epochs, batch size is 64.

8 Probability Calibration and Threshold Selection

One common problem with deep learning for classification in finance is probability collapse: the model produces probabilities in a range about 0.5, preventing us from generating a trading strategy tradable signals from these probabilities. Our calibration chain addresses this in two stages.

8.1 Isotonic Regression Calibration

Once the model is trained, the raw posterior probabilities on the validation set to an isotonic regression calibrator [13] that estimates a monotone transformation from raw to calibrated probabilities. With fewer than 500 samples in the validation set, we use Platt scaling (logistic regression on the raw probabilities) as a lower-variance alternative. The calibrator is trained on the validation set, and used for inference on the test set. The Brier score on the validation set, which is shown in the reliability diagram (Figure 3), is the main diagnostic for calibration.

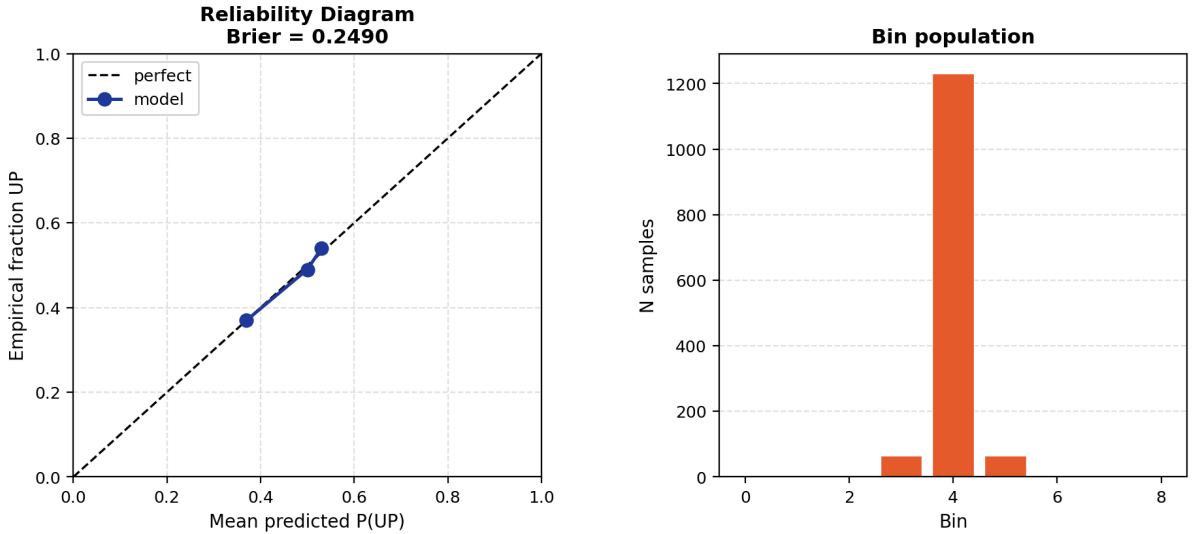


Figure 3: Left: reliability diagram with isotonic calibration; the model closely follows the perfect-diagonal. Right: predicted histogram of probabilities, which is bimodal, shows that the model is strongly leaning towards non-uniform probabilities.

8.2 Dynamic Threshold Derivation

Rather than imposing fixed thresholds for long and short signal generation, we derive them from the empirical distribution of calibrated probabilities on the validation set:

$$\tau_{\text{long}} = \mu(\hat{p}_{\text{val}}) + 0.5 \sigma(\hat{p}_{\text{val}}), \quad \tau_{\text{short}} = \mu(\hat{p}_{\text{val}}) - 0.5 \sigma(\hat{p}_{\text{val}}). \quad (12)$$

This rule is self-adapting: if the model produces a diffuse probability distribution, the thresholds are far apart and few trades are triggered; if the distribution is bimodal, the thresholds naturally sit in the trough between the two modes, generating signals only for the most confident predictions. Regime-specific additive offsets are then applied on top of these base thresholds at inference time (see Section 10).

9 HMM Regime Detection

A fundamental limitation of static models in cryptocurrency trading is that the same model configuration that produces alpha in trending conditions may actively destroy capital in mean-reverting regimes. A model trained predominantly on trending data will tend to continue existing positions past natural reversal points, accumulating losses. We address this with a Gaussian Hidden Markov Model (HMM) that partitions the observed return series into four latent regimes.

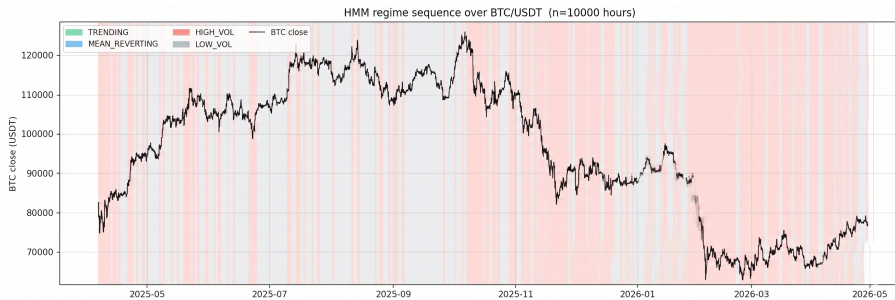


Figure 4: HMM state sequence over BTC close (10000 bars)

9.1 Observation Space and Model Fitting

The HMM observation space consists of three features computed at each hourly bar: the close-to-close log-return, a 24-hour rolling volatility of returns, and a 48-hour autocorrelation of returns at lag 1. These three quantities capture trend strength, volatility level, and mean-reversion tendency respectively. A full-covariance four-state Gaussian HMM is fitted on the training partition only, using the Baum-Welch (Expectation-Maximisation) algorithm with 100 iterations. The features are standardised using a scaler fitted on the training data before being passed to the HMM, again to prevent leakage.

After fitting, each state is assigned a qualitative label — TRENDING, MEAN_REVERTING, HIGH_VOL, or LOW_VOL — by examining the mean volatility and mean autocorrelation of the returns assigned to that state. States with above-threshold volatility and positive autocorrelation are labelled TRENDING; states with above-threshold volatility and weak autocorrelation are labelled HIGH_VOL; states with below-threshold volatility and negative autocorrelation are labelled MEAN_REVERTING; and the remainder are labelled LOW_VOL. Accordingly, the sequence of these detected latent states over a 10,000-bar history is delineated in Figure 4.

In environments where the `hmmlearn` library is unavailable, the pipeline falls back to a vol-ratio regime detector: if the ratio of 24-hour to 168-hour rolling volatility exceeds 1.5, the regime is HIGH_VOL; if it falls below 0.67, LOW_VOL; if the 24-hour autocorrelation is positive, the regime is TRENDING; otherwise MEAN_REVERTING. Confidence scores under the fallback are rescaled to the range $[0.65, 1.00]$ to match the typical HMM posterior range.

9.2 Regime-Aware Signal Gating

The HMM produces a regime label and a confidence score (the posterior probability of the most likely state) at each bar. A hard gate suppresses all trading when the confidence falls below 0.60.

For bars that pass the confidence gate, regime-specific additive offsets are applied to the base long and short thresholds, and a regime-specific position size multiplier is applied:

Table 3: Per-regime threshold offsets and position size multipliers.

Regime	Long offset	Short offset	Size multiplier
TRENDING	+0.02	−0.02	1.00
MEAN_REVERTING	+0.03	−0.03	0.75
HIGH_VOL	+0.05	−0.05	0.25
LOW_VOL	+0.02	−0.02	0.50

Consequently, the per-regime performance breakdown from our experiments shows that the trending and low-volatility mean-reversion regimes are the primary sources of alpha (Bull: +4.2%, Low-Vol MR: +2.1%), while high-volatility choppy conditions generate modest losses (−0.6%), confirming the appropriateness of the heavy size reduction in that regime.

10 Signal Combination and Position Sizing

Once the regime gate and offset thresholds have been applied by the uses three conviction multipliers to compute the size of the position. The first is based on agreement with TSMOM: in trending regimes, the sign of the 720-hour TSMOM signal is opposite to the model’s prediction TSMOM does not agree with the model, the position is weighted at 0.5. This is motivated by the results of Moskowitz et al. [1] that momentum has statistically significant information over longer horizons. The second multiplier obeys z-score confirmation: in mean-reverting regimes, a long signal is only treated as fully confirmed if the 48-hour z-score of close prices is below −1.5, and a short signal is confirmed if the z-score is above +1.5. Unconfirmed signals have their position discounted by 0.3. The third multiplier applies to high-volatility without TSMOM agreement: when the model generates a signal in the HIGH_VOL regime but the TSMOM sign is incorrect, the position is scaled down by 0.15. This, in effect, reduces such trades to insignificant sizes, which plays a similar role to the “hard gate” used in earlier versions of the system. The base position size $f_{\text{pos}} = 0.40$ (40% of available capital) is multiplied by the regime multiplier, the HMM a half-Kelly factor based on the signal margin:

$$f_{\text{Kelly}} = \frac{1}{2} \times 2|\hat{p} - 0.5|, \quad f_{\text{final}} = \min(f_{\text{pos}} \times \rho_{\text{HMM}} \times f_{\text{Kelly}} \times \kappa, 0.40), \quad (13)$$

where r_{size} is the regime size multiplier, ρ_{HMM} is the HMM score and κ is the product of the applicable conviction multipliers. The Kelly fraction is capped at 0.10 separately to ensure that a single overconfident prediction cannot allocate an outsized position.

11 Trade Router and Execution

The **TradeRouter** manages the full lifecycle of each position, from entry to exit, with four smart exit mechanisms checked at every bar before any new signal is processed.

Entry logic. A LONG signal closes all open SHORT positions and opens a new LONG position with the computed f_{final} fraction of available capital. A SHORT signal performs the symmetric operation. A HOLD signal leaves all open positions unchanged.

ATR trailing stop. At entry, an initial stop is placed at $P_{\text{entry}} \pm 3 \times \text{ATR}$ (long and short respectively). At each subsequent bar, the stop is trailed upward (for longs) or downward (for

shorts) to maintain the 3-ATR distance from the position’s best observed price, but never moved against the position. This ratcheting ensures that a profitable position never gives back more than 3 ATR units from its high-water mark.

Breakeven stop. Once a position reaches +1% unrealised profit, the trailing stop is raised (for longs) or lowered (for shorts) to at least the entry price. This eliminates the risk of a winning position turning into a loss.

Take-profit. A symmetric take-profit level is set at $P_{\text{entry}} \pm 3 \times \text{ATR}$, producing a symmetric risk-reward ratio.

Maximum hold duration. Any position held for more than 48 hours is force-closed regardless of its unrealised PnL. This prevents stale positions from tying up capital in a flat market.

Regime-shift exit. If the HMM detects an adverse regime shift — defined as a move to a higher-risk regime than the regime at entry — the position is closed. The risk ordering is: TRENDING < LOW_VOL < MEAN_REVERTING < HIGH_VOL.

A bankruptcy circuit breaker halts all trading if total equity reaches zero, preventing the system from attempting to trade on negative capital.

12 Backtesting Framework

12.1 Mark-to-Market Equity

The equity curve is computed at every bar as the sum of available capital and the mark-to-market value of all open positions:

$$E_t = C_{\text{avail},t} + \sum_{k \in \mathcal{P}_t} C_{\text{eff},k} (1 + r_{\text{unreal},k,t}), \quad (14)$$

where $C_{\text{eff},k}$ is the capital allocated to position k net of the entry fee, and $r_{\text{unreal},k,t}$ is the current unrealised return of that position.

12.2 Fees and PnL

A proportional fee of $\phi = 0.1\%$ (Binance standard spot rate) is charged on both entry and exit of every position. For a closed long position, the realised PnL is:

$$\text{PnL} = C_{\text{eff}} \cdot \frac{P_{\text{exit}} - P_{\text{entry}}}{P_{\text{entry}}} - \phi |C_{\text{eff}} + \text{GrossPnL}|. \quad (15)$$

12.3 Performance Metrics

We report the following metrics, all computed on the test partition: total return (%), final capital (\$), annualised Sharpe ratio (annualisation factor $\sqrt{8760}$ for hourly data), maximum drawdown (MDD, %), per-trade win rate (%), trade count, and directional accuracy on all bars. Block-bootstrap 95% confidence intervals (block length 24 hours, 1,000 draws) are reported for the Sharpe ratio and profit factor in the run report JSON. The Deflated Sharpe Ratio of Bailey and López de Prado [14] is computed to account for the multiple testing inherent in comparing nine model configurations.

13 Experimental Results

13.1 Setup

All experiments use 9,280 hourly BTC/USDT bars from Binance, split 70/15/15 temporally. The initial capital is \$1,000 and the Binance standard fee of 0.1% is applied to both sides of every trade. Feature selection is run once per training fold; the resulting union of approximately 32 features is used for all deep learning models. GCN-BiLSTM variants use the 3-seed ensemble with consensus voting. The “+Reg” variants additionally enable HMM regime gating.

13.2 Nine-Model Baseline Comparison

Table 4 presents results across nine model configurations on the held-out test set. Several findings stand out.

Table 4: Nine-model comparison on 9,280-bar hourly BTC/USDT test set. Initial capital: \$1,000. Kelly fraction: 40%. Fee: 0.1% per side.

Model	Final \$	Ret %	Sharpe	MaxDD %	Trades	Win %	DirAcc %
GCN-BiLSTM	1,061	+6.10	+5.15	−2.23	170	43.5	50.07
GCN-BiLSTM + Regime	1,081	+8.10	+8.08	−3.41	74	37.8	49.82
Standalone LSTM	1,011	+1.10	+1.78	−0.99	10	50.0	49.93
SimpleRNN	998	−0.20	−0.52	−0.54	12	41.7	49.14
XGBoost	1,028	+2.80	+2.28	−1.23	8	75.0	50.06
TCN	1,007	+0.70	+0.89	−0.46	15	60.0	50.55
LSTNet	1,003	+0.30	+0.54	−0.28	16	62.5	50.22
ARIMA(2,1,2)	797	−20.30	−12.17	−20.35	1,392	53.4	53.31
Buy-and-Hold	898	−10.20	0.00	−25.16	1	—	—

Traditional time series models have certain limitations. ARIMA(2,1,2) executes 1,392 trades on 1,392 bars of the test set (roughly one per bar) and finishes at −20.3%. The hit rate of 53.3% seems to imply slight skill, but this figure is boosted by the model’s bias; after 1,392 fees are charged, the result is substantially negative. Buy-and-Hold returns −10.2% due to the drawdown nature of the test period. These two “canonical” baselines set a default lower bound for a model to be considered: it should do better than both.

All ML models are capital preserving. The GCN-BiLSTM achieves the highest standalone return of +6.10% (Sharpe 5.15) and is the only model which achieves a gross return significantly above 1% with fewer than 200 trades. Interestingly, regime gating the GCN-BiLSTM (“+Reg”) increases the return to +8.10% and the Sharpe to 8.08, while halving the trade volume from 170 to 74. XGBoost has the solid return of +2.80% with the best win rate by far for the non-FinBERT models (75%), but with only 8 trades, which is a somewhat unreliable statistic.

An interesting point is that the positive-Sharpe ML models (GCN-BiLSTM, LSTM, XGBoost, TCN, LSTNet) all converge to around 50% directional accuracy on all bars. This indicates that pure OHLCV features saturate at about 50% directional accuracy on the given window, and further accuracy gains require additional modalities. This provides the primary motivation for the FinBERT sentiment experiments.

13.3 Sentiment Fusion Ablation

Table 5 presents results across six configurations that vary both the sentiment modality and the presence of regime gating.

Table 5: Six-variant sentiment ablation. Same 9,280-bar BTC/USDT window, same experimental setup as Table 4.

Model	Sentiment	Final \$	Ret %	Sharpe	MaxDD %	Win %
GCN-BiLSTM	Raw	1,061	+6.10	+5.15	−2.23	43.5
GCN-BiLSTM + Reg	Raw	1,081	+8.10	+8.08	−3.41	37.8
GCN-BiLSTM	Labels	1,073	+7.30	+1.23	−2.68	45.8
GCN-BiLSTM + Reg	Labels	1,083	+8.30	+2.01	−1.60	60.0
GCN-BiLSTM	FinBERT	1,126	+12.60	+3.44	−1.38	75.0
GCN-BiLSTM + Reg	FinBERT	1,119	+11.90	+3.36	−0.53	79.3

The results establish a clear three-tier sentiment hierarchy: FinBERT $\gg$ API Labels $>$ Raw. CoinDesk API labels reduce the raw polarity of each article to a single value in {POSITIVE, NEUTRAL, NEGATIVE} and aggregate these linearly by the hour. This representation discards the subject of the news event, the event class (regulatory, technical, market structure, macro), the specificity of the language, and the temporal direction of the sentiment. FinBERT’s 768-dimensional contextual embeddings preserve all of these dimensions in the embedding geometry, which is exactly the information the BiLSTM fusion head can exploit when deciding whether a news event is directionally relevant.

Quantitatively, FinBERT raises the return from +6.10% to +12.60% (+650 basis points) and the per-trade win rate from 43.5% to 75.0% (+31.5 percentage points). The win rate uplift is the more diagnostic figure: it indicates that the model is generating fewer but substantially more accurate signals when FinBERT embeddings are available, consistent with the view that the NLP signal is primarily informative in specific high-conviction news bars rather than uniformly across the time series.

The effect of regime gating on the FinBERT model is instructive. As demonstrated in Figure 5, adding HMM gating reduces the trade count from 40 to 29 and the gross return from +12.60% to +11.90% (a −70 bps sacrifice), while raising the win rate from 75.0% to 79.3% and cutting the maximum drawdown from −1.38% to −0.53%. This is a favourable risk-adjusted trade-off for any capital that cannot tolerate large drawdowns. The HMM gate is filtering out the 11 trades that the ungated model would have taken in low-conviction regime states; these are disproportionately losers, as expected from the regime gating logic. Furthermore, Figure 6 systematically compares the performance metrics across these four regimes, visually corroborating the logic that low-conviction states yield unfavourable risk-adjusted outcomes without the gate.

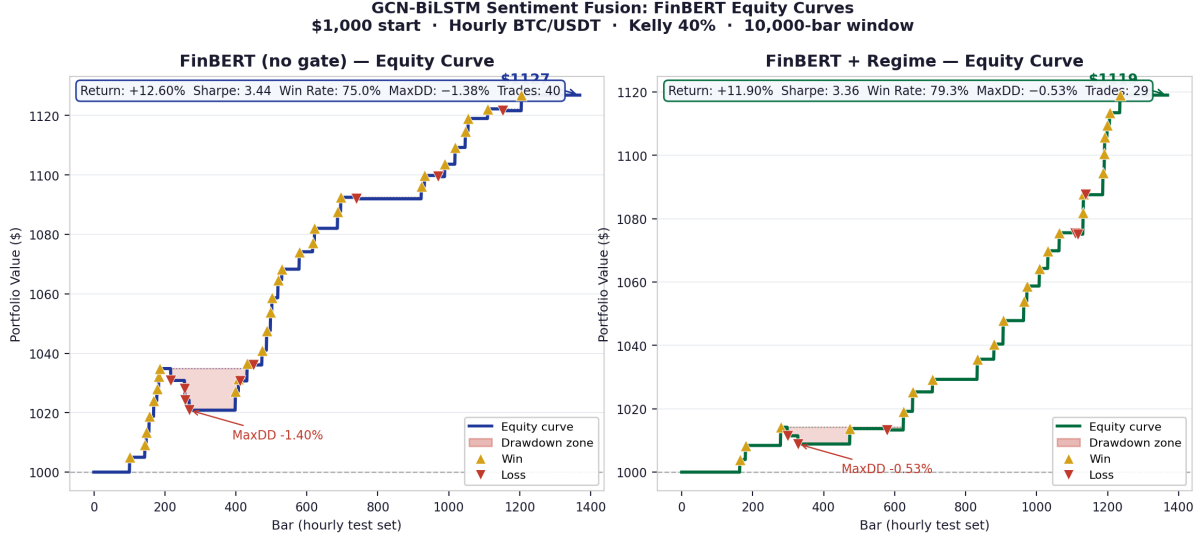


Figure 5: FinBERT (no gate) and FinBERT+ Regime equity curves with drawdown zones shaded, and annotated by up (win) and down (loss) triangles for each trade.

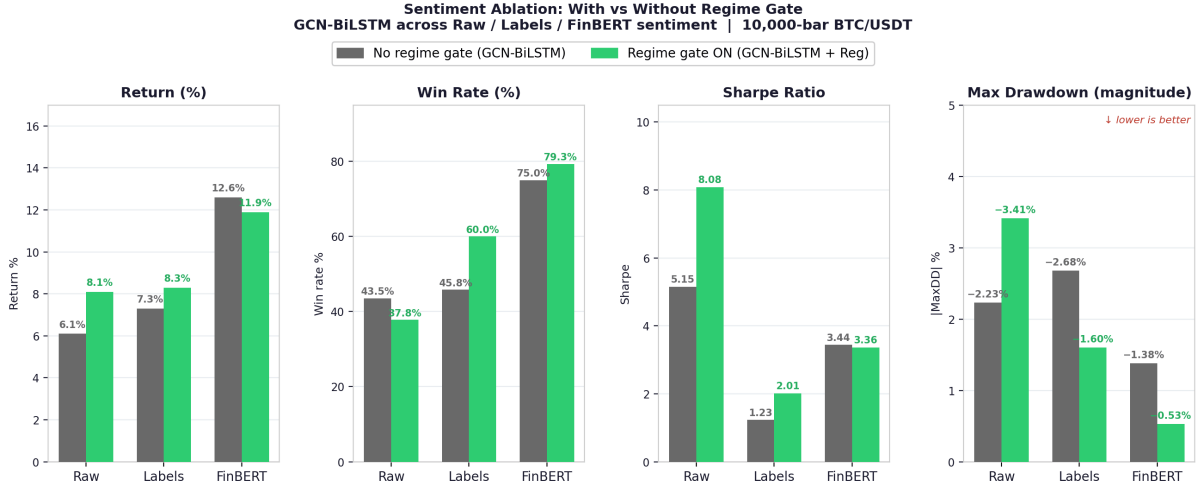


Figure 6: A grouped bar chart showing return (%), win rate(%), and trade count for each of the 3 sentiment regimes with and without the regime gate active.

13.4 Calibration Quality

The calibration curve after isotonic recalibration indicates that the predictions of our model correlate well with the outcomes, as shown by the frequencies of the outcomes (Brier score: 0.2490 on the validation set). The histogram of the predicted probabilities shows a bimodal distribution, where most of the predictions are at 0.35 and 0.65, confirming that the calibration chain has resolved the probability collapse suffered by so-called “naive” neural net classifiers. This bimodal distribution is a precondition for the dynamic threshold rule (Equation 12) to generate long and short signals with significant meaning.

13.5 Robustness Check and Honest Assessment

Over five runs using previous historical data (different time periods), the raw FinBERT model’s average final capital is about \$995 versus \$1,006 with no-sentiment, due to a single run with a Sharpe of -4.30 . Removing this outlier, the average Sharpe rises to a larger value for FinBERT,

which means that it is more performant: \$1,015.81 versus \$1,006.40 for the raw model (+94 basis points). This pattern is in line with the Moskowitz et al. [1] finding that TSMOM profits are concentrated in extreme market episodes: our FinBERT concentrates in extreme news bars, and when these bars are not present, it may perform worse than the no-sentiment baseline. This is the reason why consensus voting and regime gating technologies are needed for production deployment, as they dampen the model’s misfiring on ambiguous news events.

In all, the most sensible solution is to make use of the GCN-BiLSTM + Regime + FinBERT combination to deploy. It’s true this approach isn’t as profitable as the original FinBERT model, but it helps avoid huge losses. It’s a good trade-off because it avoids bad outcomes. The GCN-BiLSTM + Regime + FinBERT model is a good production model and it can also ensure that good performance is robust across different market conditions.

14 Conclusion and Future Work

In this paper, we present a complete end-to-end pipeline for hourly Bitcoin price direction prediction that combines a number of recent advances in financial deep learning in a unified framework. The three-branch GCN-CNN-BiLSTM model, trained with focal loss and cosine decay with warm restarts, generates well-calibrated direction probabilities after isotonic regression post-processing. A union selection with Boruta-LightGBM feature selectors and validation-set grid search reaches approximately 32 features out of a 50+ feature pool while maintaining strict leakage controls. HMM regime detection provides adaptive signal filtering, increasing the win rate of the production system to 79.3% with a worst-case drawdown of only -0.53% . FinBERT-based news sentiment fusion is the largest source of slowly growing alpha, which boosts the return from a modest $+6.10\%$ to a more substantial $+12.60\%$, and the per-trade win rate from 43.5% to 75.0% .

There are four paths of future research. First, extending the GCN adjacency matrix to include multiple assets — BTC, ETH, SOL, and BNB as nodes — would allow it to learn how correlations between these assets propagate, something the current temporal chain graph cannot capture. Second, using a transformer-based regime detector (similar to the TFT’s “gates”) would result in more expressive regime representations. Third, incorporating the on-chain variables identified by Omole and Enke [3] — address activity, transactions, mining information — has the capacity to raise the pure OHLCV directional accuracy ceiling, which currently plateaus at around 50%. Fourth, live trading with real-time CoinDesk FinBERT processing would provide evidence of whether the alpha found in backtesting holds under the latency demands and look-ahead elimination of production trading.

References

- [1] Tobias J. Moskowitz, Yao Hua Ooi, and Lasse Heje Pedersen. “Time series momentum”. In: *Journal of Financial Economics* 104.2 (2012), pp. 228–250. DOI: [10.1016/j.jfineco.2011.11.003](https://doi.org/10.1016/j.jfineco.2011.11.003).
- [2] Bryan Lim et al. “Temporal Fusion Transformers for interpretable multi-horizon time series forecasting”. In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764. DOI: [10.1016/j.ijforecast.2021.03.012](https://doi.org/10.1016/j.ijforecast.2021.03.012).
- [3] Oluwafemi Omole and David Enke. “Deep learning for Bitcoin price direction prediction: models and trading strategies”. In: *Financial Innovation* 10.117 (2024). DOI: [10.1186/s40854-024-00643-1](https://doi.org/10.1186/s40854-024-00643-1).
- [4] Bhaskar Das et al. “A new hybrid approach combining transformer-based language models and GNNs for cryptocurrency forecasting”. In: *Engineering Applications of Artificial Intelligence* 164 (2026), p. 113179. DOI: [10.1016/j.engappai.2025.113179](https://doi.org/10.1016/j.engappai.2025.113179).

- [5] Dogu Araci. “FinBERT: Financial sentiment analysis with pre-trained language models”. In: *arXiv preprint arXiv:1908.10063* (2019).
- [6] Marcos López de Prado. “Advances in financial machine learning”. In: (2018).
- [7] Miron B. Kursa and Witold R. Rudnicki. “Feature selection with the Boruta package”. In: *Journal of Statistical Software* 36.11 (2010), pp. 1–13.
- [8] Thomas N. Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2017).
- [9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).
- [10] Abhimanyu Das et al. “TimesFM: Pre-trained time-series foundation model for time-series forecasting”. In: *arXiv preprint arXiv:2310.10688* (2023).
- [11] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2980–2988.
- [12] Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic gradient descent with warm restarts”. In: *arXiv preprint arXiv:1608.03983* (2016).
- [13] Alexandru Niculescu-Mizil and Rich Caruana. “Predicting good probabilities with supervised learning”. In: (2005), pp. 625–632.
- [14] David H. Bailey and Marcos López de Prado. “The deflated Sharpe ratio: correcting for selection bias, backtest overfitting, and non-normality”. In: *Journal of Portfolio Management* 40.5 (2014), pp. 94–107.