Identification and Authenticity Classification of Abnormal Signals Based on High-Frequency Trading Data

Ziheng Zeng
University of Queensland. Nanjing, Jiangsu, China

Abstract: In an environment where high-frequency trading (HFT) has become a dominant force in financial markets, extreme changes in prices and trading volumes often occur: prices, trading volumes, and trading intensities can experience sharp peaks, jumps, or structural changes within a very short period. These abnormal signals may be triggered by real information shocks (such as macroeconomic news or sudden changes in liquidity), or by market microstructure noise or suspicious manipulation. To achieve interpretable anomaly detection and authenticity discrimination in the absence of "true value labels", this study proposes a process of "interpretable features → weakly supervised validation → scoring of uncertain samples". Taking the 1-minute BTCUSDT data provided by Binance (from June 2023 to September 2023) as an example, we construct basic features and trigger combinations on 19,402 abnormal candidate samples obtained through rule triggering, and extract sub-window statistical features (such as log-return, |log-return|, trading volume, and the mean, variance, and ratio features of the NTR in the pre- and post-window) from both sides of the abnormal center time point. Subsequently, we quantify and rank the systematic differences between different pseudo-label groups and select the 8 sub-window features with the highest discrimination. Under the premise of using only likely_true and likely_fake as weak labels, we train an interpretable Logistic Regression as a "scorer" and provide decision points for different business preferences (high recall/high precision) through threshold scanning. Finally, we output continuous risk scores and rankings for uncertain samples, providing high-value candidates for subsequent manual review or more complex models.

Keywords: High-Frequency Trading; Anomaly Detection; Explainable Features; Weakly Supervised Learning; Logistic Regression; Threshold Selection

1. Introduction

In the markets of crypto assets, electronic and quantitative trading, high-frequency data exhibit strong noise and event-driven characteristics: they also present a "spike/jump" pattern, which may correspond to real information shocks, or be by-products of liquidity withdrawal, changes in matching mechanisms, or short-term manipulation/brushing volume behaviors. For research and risk control, the key difficulties lie in:

(1) Abnormalities occur frequently, but the true value labels of "true anomalies/false anomalies" are scarce;

(2) Black-box models alone often fail to explain the sources of anomalies and are difficult to be implemented as usable rules;

(3) The statistical significance of the same anomaly varies under different market conditions, requiring relative measurement and threshold strategies.

The objective of this study is not to pursue the strongest classifier, but to construct an interpretable and gradually upgradable research route. First, it aims to prove which features have systematic differences among different pseudo-label groups and rank the most effective combination of discriminative features. Then, a lightweight and interpretable model is trained under weak supervision to transform these features into usable "discriminative scorers". Finally, continuous scores are provided for a large number of uncertain samples to support manual review or subsequent semi-supervised/PU-learning research.

2. Research Background

In the microstructure and regulatory practices of the international market, one of the most representative cases of "false anomalies" is the manipulation behavior such as spoofing. The US Commodity Exchange Act includes misleading

order cancellations that affect the market in its regulatory scope, providing a legal basis for subsequent academic research and regulatory identification [1]; The CFTC's interpretation of disruptive practices further refined the determination boundaries and enforcement standards for related behaviors [2].

From the perspective of market microstructure research, the interrelationship between high-frequency prices, trading volumes, and transaction counts is an important entry point for identifying abnormal signals. Cartea et al. systematically discussed the relationship between order flow, price shocks, and liquidity in the high-frequency trading environment, and pointed out that observing price fluctuations alone is insufficient to explain abnormal behavior; it is necessary to combine trading intensity and persistence characteristics for a comprehensive judgment [3]. In the context of macro news shocks, Andersen et al.'s research on announcement information and high-frequency price discovery indicates that real information shocks are usually accompanied by increased trading activity and diffusion of volatility [4]; Chaboud et al.'s research on high-frequency trading effects also suggests that different types of shocks exhibit differences in the "price - transaction - persistence" dimension [5]. These conclusions collectively support the design approach of this paper in constructing a feature system using "amplitude, trading volume, transaction count, persistence, and subsequent trend".

In terms of anomaly detection methods, traditional unsupervised methods such as LOF and Isolation Forest can effectively identify statistically significant outliers, but they usually can only answer "whether it is abnormal", and it is difficult to directly answer "whether the anomaly is more like a manipulation signal" [6,7]. Therefore, in the detection layer, this paper retains statistical anomaly identification, and in the discrimination layer, it introduces weak supervision thinking： drawing on the ideas of weak supervision frameworks such as Snorkel, it uses a set of interpretable rules to generate pseudo labels and confidence levels [8]; at the same time, considering the incomplete positive and negative samples and scarce labels, it references the processing methods of PU Learning related research for the treatment of weakly labeled scenarios [9]. This design enables the research to form a "detection - screening events - weak labeling - discrimination" feasible process in the absence of manually labeled data.

In the construction of model output and interpretable discriminators, this paper adopts Logistic Regression and uses its probability output as a score indicating "more like a real shock". This approach is consistent with Platt's idea of probabilistic output and calibrated scoring, facilitating subsequent threshold scanning, precision-recall trade-off, and uncertain sample ranking [10]. In recent years, there have also been studies attempting to use machine learning to identify market manipulation or abnormal trading patterns, indicating that building discriminators based on interpretable features has practical significance; however, with the setting of only using publicly available high-frequency trading data and explicitly distinguishing "real shocks vs. false anomalies", existing work is still relatively scarce, which constitutes the research space and innovation points of this paper [11].

3. Data and Problem Definition

3.1 Data Source

This project uses the 1-minute granularity BTCUSDT trading data provided by Binance Trading Platform to construct the sample, with the time range from June to September 2023. The K-line fields mainly include open, high, low, close, volume, num_trades, quote_volume, taker_buy_base, taker_buy_quote, etc. The subsequent anomaly detection and feature engineering mainly rely on three dimensions: price changes (yield rate), price range (amplitude), and trading activity (volume/number of transactions).

Through the existing rules triggering and basic feature calculation, we obtain the abnormal candidate sample table cases (a total of 19,402 rows), as shown in Table 1, which includes:

• ts: Abnormal center time;

• trigger_combo: Trigger combination (such as RGV, RG, etc.);

• pseudo_label: Pseudo label (likely_true / likely_fake / uncertain).

Due to the absence of true labels, we have divided the task into two layers:

A) Explainable Difference Verification: In the computable feature space, examine whether there are systematic differences in the features between different pseudo-label groups, and sort

them by discrimination degree;

B) Weak Supervised Discrimination: With only likely_true and likely_fake as weak labels, train a lightweight model to output probability scores, which are used to sort and filter uncertain samples.

Table 1. Statistics of Pseudo-Labeled Sample Quantities

pseudo_label	count
uncertain	12,109
likely_true	5,350
likely_fake	1,943

3.2 External Event Data

To distinguish between the fluctuations caused by macro/regulatory information shocks and potentially endogenous abnormal signals, this study predefines a set of external event windows (UTC time zone) covering regulatory lawsuits, significant judicial rulings' spillover effects, and important international events such as the release of US macro data (CPI) and monetary policy decisions (FOMC) during the period from June to September 2023. After constructing 1-minute OHLCV bars, this paper first performs anomaly detection on returns, volatility, and trading volume based on the robust z-score (rolling median + MAD) of the rolling window, obtaining the set of abnormal bars; subsequently, it matches the timestamp of each bar with the external event windows to classify the anomalies as "external event-driven anomalies" and "non-external event anomalies". The subsequent weakly supervised pseudo-label and discriminator training mainly focuses on non-external event anomalies to reduce the confusion caused by exogenous information shocks in the identification of true and false signals. The event windows are of two granularities: (a) Major news events: using a coarse-grained window of 1-2 days to cover the diffusion of news and multiple reactions; (b) Timely macro announcements (CPI/FOMC): using a tight window of "announcement time ± 2 hours" to cover the main shocks.

The predefined external event windows used to filter exogenous shocks are summarized in Table 2.

Table 2. External Event Windows Used for External-Shock Filtering

#	event	start (UTC)	end (UTC)
1	SEC v Binance (lawsuit)	2023-06-05 00:00:00Z	2023-06-06 23:59:59Z
2	SEC v Coinbase (lawsuit)	2023-06-06 00:00:00Z	2023-06-07 23:59:59Z
3	BlackRock spot BTC ETF file	2023-06-15 00:00:00Z	2023-06-16 23:59:59Z
4	Ripple ruling spillover	2023-07-13 00:00:00Z	2023-07-14 23:59:59Z
5	Aug 17-18 BTC selloff	2023-08-17 00:00:00Z	2023-08-18 23:59:59Z
6	Grayscale court ruling	2023-08-29 00:00:00Z	2023-08-30 23:59:59Z
7	US CPI (Jun) release	2023-07-12 10:30:00Z	2023-07-12 14:30:00Z
8	US CPI (Jul) release	2023-08-10 10:30:00Z	2023-08-10 14:30:00Z
9	US CPI (Aug) release	2023-09-13 10:30:00Z	2023-09-13 14:30:00Z
10	FOMC decision	2023-06-14 16:00:00Z	2023-06-14 20:00:00Z
11	FOMC decision	2023-07-26 16:00:00Z	2023-07-26 20:00:00Z
12	FOMC decision	2023-09-20 16:00:00Z	2023-09-20 20:00:00Z

4. Research Process and Methods

The overall process is as follows: detection → elimination of external events → expansion of feature engineering → weak supervision scoring and classification into three categories → comparative analysis.

4.1 Identification of Abnormal Signals

Based on the 1-minute data, three types of basic features are defined: $log\_ret = ln(close\_t) - ln(close\_\{t-1\})$; $range\_pct = (high\_t - low\_t) / close\_t$; $log\_vol = ln(volume\_t)$. For each feature, a rolling mean μ_t and a rolling standard deviation σ_t are calculated using a fixed window length W=60 (REF_WINDOW=60, and the requirement is $min\_periods=W$), and the Z score z_t is obtained: $z\_t = (x\_t - \mu\_t) / \sigma\_t$. When any one of $|z\_ret| \geq 4.0$, $|z\_rng| \geq 4.0$, and $|z\_vol| \geq 4.0$ is satisfied ($Z\_RET = Z\_RNG = Z\_VOL = 4.0$), this minute is marked as an anomaly point ($anomaly\_1m = 1$), otherwise it is 0. Here, minutes with $volume = 0$ are replaced with NaN before taking the logarithm to avoid infinite values caused by $ln(0)$.

To retain the source of anomaly triggering, a trigger combination trigger_combo is further defined: if the yield trigger is recorded as R, the amplitude trigger as G (range), and the volume

trigger as V (volume), then an anomaly point can be represented as a combination of R/G/V (such as RGV, RV, G, etc.). This combination is helpful for subsequent stratified analysis of different types of anomalies.

4.2 Window Expansion Feature Engineering: Extracting Microstructural Features of Non-event Abnormal Points

For the abnormal points that have been filtered out of the influence of external events, a time context expansion window (context window) is constructed centered on the abnormal center time t. In the code of this article, CTX_W is set to 15, meaning a local window of $[t-15min, t+15min]$ is taken, and it is further divided into two "sub-windows": the pre-window $[t-15, t-1]$ and the post-window $[t+1, t+15]$.

In this expanded window/sub-window, the microstructural features are extracted:

(1) Comparison of the volume/transaction count before and after:

$pre\_vol\_mean = mean(volume\_pre)$,
$post\_vol\_mean = mean(volume\_post)$;
$vol\_spike\_ratio = volume\_t / (pre\_vol\_mean + 1e-1$
;
$post\_pre\_vol\_ratio = post\_vol\_mean / pre\_vol\_mea$
(when $pre\_vol\_mean > 0$, otherwise set to NaN);
$pre\_ntr\_mean = mean(num\_trades\_pre)$,
$post\_ntr\_mean = mean(num\_trades\_post)$;
$ntr\_spike\_ratio = num\_trades\_t / (pre\_ntr\_mean +$
.

(2) Volatility structure:

$pre\_lr\_std = std(log\_ret\_pre)$,
$post\_lr\_std = std(log\_ret\_post)$.

(3) Subsequent price behavior:

$fwd\_ret\_5 = ln(close\_\{t+5\} / close\_t)$,
$fwd\_ret\_15 = ln(close\_\{t+15\} / close\_t)$; and define
$continuation\_15 = sign(log\_ret\_t) \cdot fwd\_ret\_15$,
$reversal\_15 = -sign(log\_ret\_t) \cdot fwd\_ret\_15$
($sign \geq 0$ takes +1, otherwise −1).

(4) Persistence: The "medium intensity" duration of fluctuations or trading volumes in the next 0-15 minutes (including t) is counted, persist_rng_ge2 is the number of minutes within the interval $[t, t+15]$ that satisfy $|z\_rng| \geq 2.0$, and persist_vol_ge2 is the number of minutes that satisfy $|z\_vol| \geq 2.0$.

*These features aim to highlight the differences between two types of abnormal mechanisms: (a) Message/real impact type abnormalities usually accompany the synchronous amplification of trading volume and transaction count, and present a certain continuity trend; (b) Suspected manipulation type abnormalities may exhibit signs such as "price or amplitude abnormality but no matching trading volume", "rapid reversal after a short-term pull-up", or "high amplitude persistence but lacking real trading volume support".

4.3 Construction of Sub-window Features and Verification of Differences

To enhance the explanatory power, we constructed symmetrical time sub-windows around each central time point of the anomaly (e.g., ±30 minutes). Within these sub-windows, we calculated statistics for both the "pre-anomaly window" and the "post-anomaly window", and constructed ratio/peak features to capture whether the anomaly manifested as "structural changes before and after the event". The central point t was not included in the statistics to avoid directly leaking the extreme values of the triggering point into the comparison features. The representative sub-window features used in this project include:

• The mean and standard deviation of log-return (pre_lr_mean, pre_lr_std, post_lr_mean, post_lr_std);
• The mean of |log-return| (pre_abs_lr_mean, post_abs_lr_mean);
• The mean/median of trading volume (pre_vol_mean, pre_vol_median, post_vol_mean, post_vol_median);
• The ratios and peak ratios before and after (such as vol_spike_ratio_post_pre, abs_lr_mean_ratio_post_pre, ntr_spike_ratio_post_pre).

4.4 Generate Weak Tags

Since the true labels (manipulated/non-manipulated) are difficult to obtain directly, this paper adopts the weak supervision idea and assigns scores to abnormal samples using heuristic rules without introducing manual annotations. The rules are divided into two groups: fake_score (more like fake anomalies/manipulations) and true_score (more like real shocks/message-driven).

(1) fake_score rules (+1 for each satisfied rule)

$vol\_spike\_ratio < 2$ and $|z\_rng| \geq 3$: Amplitude abnormal but trading volume does not increase synchronously;

$ntr\_spike\_ratio < 2$ and $|z\_ret| \geq 3$: Yield rate abnormal but transaction count does not increase

synchronously; *persist_rng_ge*2 ≥ 5 and *reversal*_15 = 1: High amplitude and continuous, with a reversal within 15 minutes.

(2) true_score rules (+1 for each satisfied rule)

vol_spike_ratio ≥ 3 and *ntr_spike_ratio* ≥ 3 and |*z_ret*| ≥ 3 : Fluctuation accompanied by simultaneous increase in trading volume and transaction count;

post_pre_vol_ratio ≥ 1.2 and *continuation*_15 = 1 : Subsequent transactions continue to increase and price continues;

*persist_vol_ge*2 ≥ 5 and |*z_ret*| ≥ 4 : High fluctuation and active trading have persistence.

(3) False labels and confidence

If *true_score* ≥ 2 and *fake_score* = 0 , mark as likely_true;

If *fake_score* ≥ 2 and *true_score* = 0 , mark as likely_fake;

The rest are marked as uncertain.

At the same time, the weak label confidence is defined:

pseudo_conf = *max*(*fake_score*, *true_score*) / 3 .

It is used for subsequent threshold scanning/sample weighting (if using a model) or for sorting reference for uncertain samples.

4.5 Feature Ranking

This study further conducted "feature difference verification" on the abnormal samples to identify the most discriminative explanatory features and form the input for the subsequent discriminator. Based on the existing weakly labeled pseudo_label, the samples included 12,109 uncertain ones, 5,350 likely_true ones, and 1,943 likely_fake ones. On the sub-window samples, we used likely_true as the positive class reference and regarded the remaining samples (including likely_fake and uncertain ones) as the control group. We calculated Cohen's d for all sub-window features.

$$d = (\mu 1 - \mu 2) / sp \quad (1)$$

The calculation results were sorted by |d| from high to low, and the missing rate (na_rate) was recorded to exclude variables with severe or unstable missingness. The results showed that the "leading/lagging peaks/proportion" features related to trading volume and trading intensity were the most discriminative, such as vol_spike_ratio_post_pre, ntr_sprike_ratio_post_pre, etc.To visualize the relative discrimination strength of the candidate features, Figure 1 presents the overall ranking of effect sizes based on |Cohen's d|.

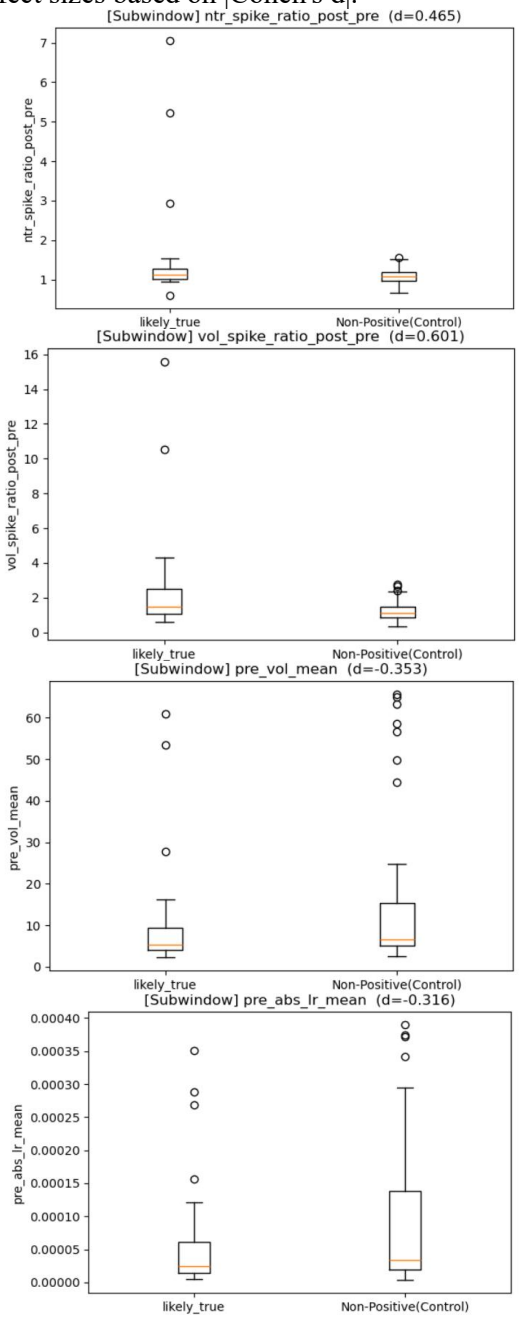


Figure 1. Ranking of the Effect Sizes of Sub-window Characteristics (|Cohen's d|)

The specific values of the highest-ranked sub-window features are listed in Table 3.

Table 3. Top Sub-window Features Ranked by Effect Size

feature	cohens_d	na_rate	abs_d	direction
vol_spike_ratio_post_pre	0.601093	0.0	0.601093	likely_true higher
ntr_spike_ratio_post_pre	0.464725	0.0	0.464725	likely_true higher
pre_vol_mean	-0.353445	0.0	0.353445	likely_true lower
pre_abs_lr_mean	-0.316145	0.0	0.316145	likely_true lower

abs_lr_mean_ratio_post_pre	0.287410	0.0	0.287410	likely_true higher
pre_ret_mean	-0.281054	0.0	0.281054	likely_true lower
pre_lr_std	-0.250311	0.0	0.250311	likely_true lower
pre_vol_median	-0.245918	0.0	0.245918	likely_true lower
pre_lr_mean	0.244678	0.0	0.244678	likely_true higher
post_vol_mean	0.150523	0.0	0.150523	likely_true higher
post_ret_mean	0.098899	0.0	0.098899	likely_true higher

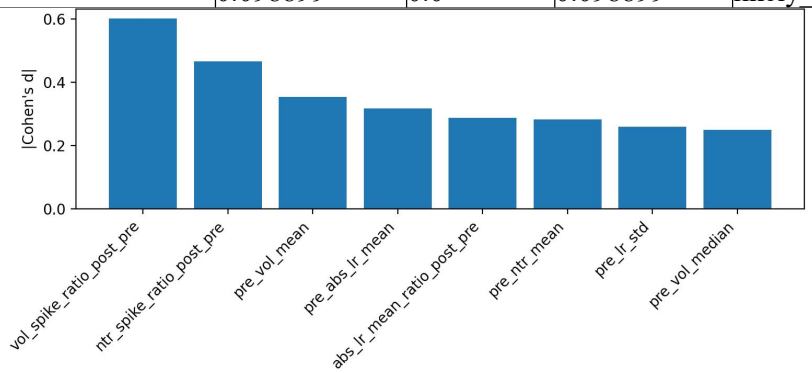


Figure 2. Top Subwindow Features by Effect Size

Figures 1–2 and Table 3 jointly show the discrimination strength of the candidate sub-window features. Based on these results, the study retains the variables with stronger effect sizes and lower missing rates for subsequent model construction.

This operation aims to select a group of candidate features that have the greatest discriminatory potential and are easily interpretable, providing an input variable set for the subsequent construction of a unified scorer.

4.6 Weak Supervision/Score Verification: Transform "Explanatory Features" into Usable Discriminators

During the model training phase, only the two relatively confident classes of samples in the weak labels are used, namely *pseudo_label* $\in$ {*likely_true*, *likely_fake*}. In the actual data, there are 7,293 samples (59 columns) entering the training process, among which *likely_true* = 5,350 and *likely_fake* = 1,943; and likely_true is mapped to the binary label $y = 1$, while likely_fake is mapped to $y = 0$. y=0. y=0. Subsequently, stratified random partitioning (*stratify=y*, *random_state*=42) is performed on the training set to obtain the training set *Train* = 5,105 and the test set *Test* = 2,188, which are used to evaluate the generalization performance of the discriminator.

The discriminator model selects an interpretable and simple Logistic Regression, and adopts a preprocessing - modeling integrated pipeline: missing values are imputed using the median (median imputation), features are standardized (standardization), and category weight balance is enabled in Logistic Regression (*class_weight*="*balanced*") to alleviate the imbalance in the class ratio of likely_true and likely_fake (*solver*="*liblinear*", *max_iter*=3000). The classification outcomes at the default threshold are first illustrated through the confusion-matrix heatmap in Figure 3.

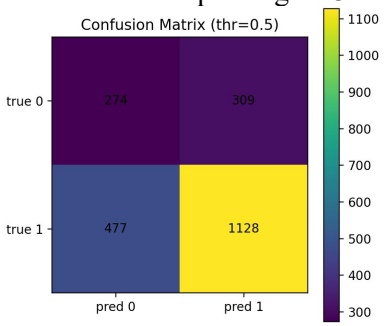


Figure 3. Heatmap of Confusion Matrix

On the test set, the model achieved an ROC-AUC of 0.6267 and a PR-AUC (Average Precision) of 0.8201. When the default decision threshold thr = 0.5 was used(see Table 4), the confusion matrix was:

Table 4. Confusion Matrix at thr = 0.5

Actual / Predicted	pred 0	pred 1
true 0	274	309
true 1	477	1128

Report:

	precision	recall	f1-score	support
0	0.3648	0.4700	0.4108	583
1	0.7850	0.7028	0.7416	1605
accuracy			0.6408	2188
macro avg	0.5749	0.5864	0.5762	2188
weighted avg	0.6730	0.6408	0.6535	2188

Figure 4. Outcome of Accuracy and Recall Rate

$precision(1) = TP / (TP + FP) = 1128 / (1128 + 309) = 0.$

$recall(1) = TP / (TP + FN) = 1128 / (1128 + 477) = 0.7028$

$F1(1) = 2PR / (P + R) = 0.7416$

$accuracy = (TP + TN) / total = (1128 + 274) / 2188 = 0.6408$

Furthermore, in order to obtain a more applicable running point, this study conducted a grid scan of the threshold (ranging from 0.05 to 0.95 in equal intervals), calculated the precision/recall/F1/accuracy for each threshold, and selected the threshold with the best F1 value as the reference running point, as shown in Figure 4.

The first rows of the threshold-scanning results are reported in Table 5.

Table 5. Threshold Sweep Head

thr	precision	recall	f1	accuracy
0.05	0.733425	0.999377	0.845992	0.733090
0.10	0.733425	0.999377	0.845992	0.733090
0.15	0.733181	0.998131	0.845383	0.732176
0.20	0.733509	0.997508	0.845078	0.731718
0.25	0.732691	0.995639	0.844163	0.730347

Based on the actual output, the optimal F1 threshold is thr = 0.05. At this point, the precision is 0.7334, the recall is 0.9994, the F1 score is 0.8460, and the accuracy is 0.7331.

Meanwhile, this study has plotted and reported the Precision–Recall curve and the ROC curve to comprehensively present the trade-off relationships under different thresholds.The overall trade-off between precision and recall is further illustrated in Figure 5, which reports both the Precision–Recall curve and the ROC curve.

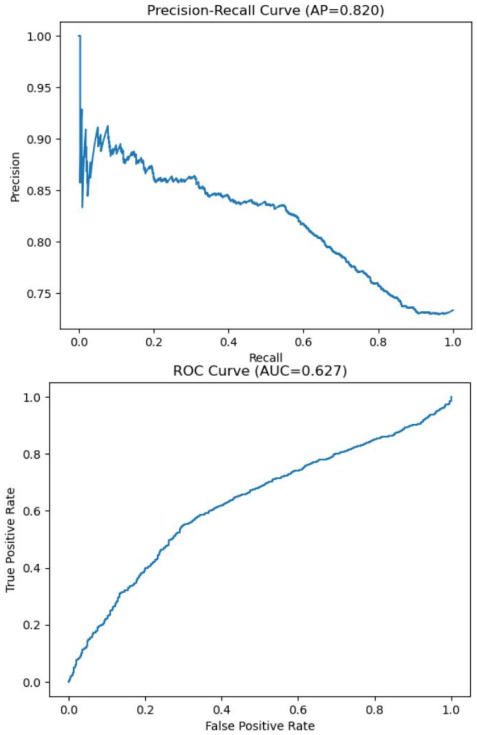


Figure 5. PR and ROC Curve

5. Experimental Results

The feature summary table used in the experiment is full_features_with_subwindow.csv, which contains 19,402 samples and 59 fields. The sample distribution of the pseudo labels pseudo_label is:

uncertain = 12,109 ; *likely_true* = 5,350 ; *likely_fake* = 1,943.

In this paper, only likely_true and likely_fake are used as trainable samples to reduce the interference of pseudo label noise on the model learning, resulting in a total of 7,293 trainable samples. Under the binary classification setting, let likely_true be the positive class (y = 1) and likely_fake be the negative class (y = 0), and the corresponding class counts are: y = 1 (5,350), y = 0 (1,943). Based on the verification of sub-window feature differences (effect size ranking), the following sorting table is obtained:

The actual features used in this training are as follows:

vol_spike_ratio_post_pre, ntr_spike_ratio_post_pre, pre_abs_lr_mean, abs_lr_mean_ratio_post_pre, pre_vol_median, pre_vol_mean_x, pre_ntr_mean_x, pre_lr_std_x。

In terms of model selection, to balance interpretability and simplicity of implementation, this paper adopts Logistic Regression as the discriminator, fills in the missing values with the median, standardizes the features (using StandardScaler), and uses *class_weight*="*balanced*" to alleviate the imbalance of the classes of likely_true and likely_fake. The size of the training set is 5,105 and the size of the test set is 2,188. The following indicators were obtained on the test set: The aggregate evaluation metrics of the classifier are reported in Table 6.

Table 6. Summary of Model Performance

Metric	Value
ROC-AUC	0.627
PR-AUC (AP)	0.820
Test set size	2,188

Among them, *PR−AUC* pays more attention to the precision-recall trade-off in the case of class imbalance; *ROC−AUC* reflects the overall ranking ability of the model (0.5 represents the random level). An AUC value of 0.627 indicates that the model has a certain degree of discrimination ability, but there is still room for improvement.

5.1 Confusion Matrix and Threshold

Influence

For a more concrete interpretation of the predictions, the confusion matrix at the default threshold is reported in Table 4. Taking thr = 0.5 as an example (rows = true[0,1], cols = pred[0,1]).

Among the corresponding classification results, the precision of the positive class (likely_true) on the test set is 0.7850, the recall is 0.7028, and the F1 score is 0.7416 (with a support of 1605); the precision of the negative class (likely_fake) on the test set is 0.3648, the recall is 0.4700, and the F1 score is 0.410 (with a support of 583); the overall accuracy is 0.6408. This result indicates that in the context where there is noise in the weakly supervised pseudo-labels, the model can still learn certain distinguishing signals from the microscopic structural features of the sub-windows and provide relatively stable recognition capabilities for "anomalies that are more likely to be likely_true".

The first rows of the threshold-scanning results are reported in Table 5.

Table 5. Threshold Sweep Head

thr	precision	recall	f1	accuracy
0.05	0.733425	0.999377	0.845992	0.733090
0.10	0.733425	0.999377	0.845992	0.733090
0.15	0.733181	0.998131	0.845383	0.732176
0.20	0.733509	0.997508	0.845078	0.731718
0.25	0.732691	0.995639	0.844163	0.730347

The threshold scanning results show that when the threshold is low (for example, $thr = 0.05$), the model's recall rate is close to 1 ($recall \approx 0.9994$), and the corresponding F1 reaches the highest value of this scan ($F1 \approx 0.8460$, $precision \approx 0.7334$, $accuracy \approx 0.7331$). This indicates that the selection of the threshold significantly affects the "screening strategy": a lower threshold is more inclined towards high recall (suitable for minimizing the omission of candidate events), while a higher threshold is more inclined towards high precision (suitable for reducing false alarms). Therefore, threshold scanning is used in this paper to transition from "probability output" to "available decision rules" and supports the setting of thresholds under different application goals.

In the weak supervision framework, uncertain samples do not participate in training but are used as a screening pool for final scoring and ranking. Specifically, this paper uses the trained Logistic Regression to calculate the probability score *p_likely_true* for all uncertain samples (i.e., $P(y=1 \mid x)P(y=1|x)P(y=1 \mid x)$), and outputs it to the file uncertain_scored.csv, which is used to select the candidate set more likely to belong to "true shocks/message-driven anomalies" in descending order of scores in the subsequent steps.

5.2 Uncertainty in Sample Scoring and Output

After the model training is completed, the score (the probability of belonging to 1) is calculated for the 12,109 samples with pseudo_label=uncertain, and the file "uncertain_scored.csv" is exported. This file can be used for:

• Sorting by score and selecting the Top-K for manual review;

• Setting two thresholds ($high\ threshold = high\ confidence\ true$, $low\ threshold = high\ confidence\ fake$), to form high-quality pseudo labels, which can be used for subsequent stronger models (such as XGBoost/PU-learning) or calibration (Platt scaling/isotonic).

6. Discussion and Limitations

(1) False label noise: likely_true/likely_fake is derived from the combination of rules and external signals, and there is inherent mislabeling. The upper limit of AUC will be limited by the noise.

(2) Feature absence and sample coverage: Some anomalies may lack sub-window features or have missing values. Further improvement of alignment strategies and missing value handling (such as robust interpolation/group standardization) is needed.

(3) Model capability: The advantage of Logistic Regression is its interpretability and robustness, but it is limited in expressing non-linear boundaries. Later, linear SVM, tree models, or GBDT with monotonic constraints can be introduced while retaining interpretability.

(4) Evaluation criteria: Currently, the evaluation uses false labels as "weak true values". To form more rigorous conclusions, it is necessary to introduce manual annotation, small sample event auditing, or cross-time period generalization testing.

7. Conclusion

In the absence of true labels, this study proposed and implemented a research process for explaining feature verification. On the 1-minute

data of BTCUSDT, this study constructed sub-window statistics features around the anomaly center, quantified and ranked the systematic differences between groups using Cohen's d, and obtained a high-discrimination feature set centered on the changes in trading volume and trading intensity before and after; a lightweight and explainable Logistic Regression scorer was trained to achieve an *ROC−AUC* of 0.627 and a *PR−AUC* of 0.820 on the test set, and an operable precision-recall trade-off was given through threshold scanning. Then, continuous risk scores and sorting results were output for a large number of uncertain samples, providing a basis for subsequent manual review and semi-supervised upgrade.

References

[1] Cornell Law School, Legal Information Institute. 7 U.S.C. §6c(a)(5)(C) (Commodity Exchange Act anti-spoofing provision).

[2] U.S. Commodity Futures Trading Commission (CFTC). (2013). Interpretive Guidance and Policy Statement Regarding Disruptive Practices (including spoofing).

[3] Cartea, Á., Jaimungal, S., & Penalva, J. (2015). Algorithmic and High-Frequency Trading. Cambridge University Press.

[4] Andersen, T. G., Bollerslev, T., Diebold, F. X., & Vega, C. (2003). Micro effects of macro announcements: real-time price discovery in foreign exchange. American Economic Review.

[5] Chaboud, A. P., et al. (2004). The High-Frequency Effects of Macroeconomic Announcements on Prices and Trading Activity in the Foreign Exchange Market. Finance and Economics Discussion Series, Board of Governors of the Federal Reserve System.

[6] Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. SIGMOD.

[7] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. ICDM.

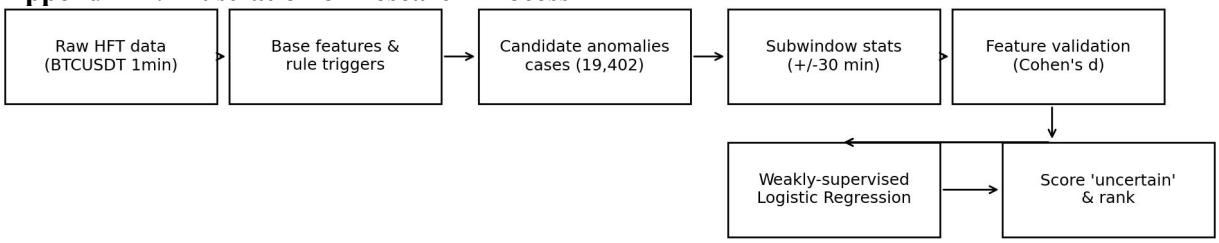
[8] Ratner, A., et al. (2020). Snorkel: Rapid training data creation with weak supervision. VLDB Journal.

[9] Bekker, J., & Davis, J. (2018). Learning from Positive and Unlabeled Data: A Survey. arXiv:1811.04820.

[10] Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.

[11] Zulkifley, M. A., Munir, N. N. A., Sukor, N. S. A., & Ameerbakhsh, O. A. (2023). Survey on Stock Market Manipulation Detectors Using Artificial Intelligence. Computers, Materials & Continua. (Tech Science Press)

Appendix A: Illustration of Research Process



The core output files of this project are as follows:

• full_features_with_subwindow.csv: Main sample table (with sub-window features merged)

• feature_weights.csv: Feature weights for Logistic Regression (for interpretability)

• model_metrics.txt: Model evaluation metrics and threshold scanning results

• uncertain_scored.csv: Scoring and ranking results for uncertain samples

