# CryptoBot Platform

## Feature System Documentation

Version 1.0 | January 2026

# Table of Contents

# 1. System Overview

The CryptoBot Feature System transforms raw OHLCV price data into actionable signals for trading decisions. It provides a comprehensive set of technical, volatility, and regime detection features across multiple timeframes.

## Key Statistics

| Metric | Value |
| --- | --- |
| Total Features | 282 features |
| Per-Timeframe Features | 220 (across 6 TFs) |
| Cross-Timeframe Features | 17 |
| Regime Features | 45 |
| Warmup Period | ~11 days (after window scaling) |
| Compute Time | ~4.6 seconds (typical dataset) |

# 2. Architecture

The feature system uses a registry-based architecture where features are self-describing classes that can be dynamically discovered and computed.

## Component Stack

| Component | File | Purpose |
|---|---|---|
| Base Framework | base.py | Feature class, registry, utility functions |
| Technical | technical.py | MAs, RSI, ATR, returns |
| Volatility | volatility.py | Rolling vol, GARCH, vol regimes |
| Regime | regime.py | BinSeg, MSM, Hybrid regime detection |
| Multi-TF Features | mt_features.py | Per-timeframe feature computation |
| Multi-TF Regime | mt_regime.py | Multi-TF regime detection |
| Feature Engine | engine.py | Main interface for feature computation |
| Feature Analysis | features.py | Correlation, importance analysis |

## Feature Registration

Features are registered using the @register_feature decorator:

```
@register_feature
class MyFeature(Feature):
name = 'my_feature'
lookback = 24
output_type = 'continuous'

def compute(self, df: pd.DataFrame) -> pd.Series:
return df['close'].rolling(24).mean()
```

# 3. Feature Categories

## Naming Convention

Multi-timeframe features follow a consistent naming pattern:

**{category}_{metric}_{timeframe}**

Examples: ret_log_24h, vol_ann_168h, ma_ratio_72h, regime_hybrid_1h

## Category Prefixes

| Prefix | Category | Example Features |
|---|---|---|
| ret | Returns | ret_log, ret_cum, ret_zscore |
| vol | Volatility | vol_std, vol_ann, vol_parkinson, vol_garman |
| rng | Range/Candle | rng_hl, rng_atr, rng_body |
| ma | Moving Averages | ma_6, ma_12, ma_24, ma_ratio, ma_cross |
| mom | Momentum | mom_rsi, mom_roc, mom_stoch |
| vlm | Volume | vlm_ratio, vlm_ma, vlm_zscore |
| regime | Regime Detection | regime_binseg, regime_msm, regime_hybrid |
| x | Cross-Timeframe | x_ma_1h_vs_24h, x_price_position |

# 4. Technical Features

Standard technical analysis indicators adapted for cryptocurrency markets.

## Moving Averages

| Feature | Description | Lookback |
| --- | --- | --- |
| sma_6 | 6-hour SMA (intraday trend) | 6 bars |
| sma_24 | 24-hour SMA (daily trend) | 24 bars |
| sma_72 | 72-hour SMA (3-day trend) | 72 bars |
| sma_168 | 168-hour SMA (weekly trend) | 168 bars |
| ma_score | Count of MAs price is above (0-3) | 72 bars |
| price_vs_sma_* | Price / SMA ratio | Varies |

## Other Technical Indicators

| Feature | Description | Output Type |
| --- | --- | --- |
| atr_14 | 14-period Average True Range | Continuous |
| atr_percent | ATR as percentage of price | Continuous |
| rsi_14 | 14-period Relative Strength Index | Continuous (0-100) |
| returns_1h | 1-hour log returns | Continuous |
| returns_24h | 24-hour log returns | Continuous |
| volume_ratio | Current volume vs 24h average | Continuous |
| hl_range_pct | High-Low range as % of close | Continuous |

# 5. Volatility Features

Volatility is highly predictable (~60% correlation with strong mean reversion) and forms the foundation of the trading strategy. Multiple estimators capture different aspects of volatility.

| Feature | Description | Calculation |
|---|---|---|
| rolling_vol_24h | 24-hour realized vol (annualized) | Std(returns) * sqrt(8760) |
| rolling_vol_168h | Weekly realized vol (annualized) | Std(returns) * sqrt(8760) |
| garch_vol_simple | EWMA volatility (RiskMetrics style) | EWMA(variance) * sqrt(8760) |
| vol_parkinson | Parkinson estimator (high-low) | Based on log(H/L) |
| vol_garman | Garman-Klass estimator (OHLC) | Uses all 4 prices |
| vol_regime | Volatility tercile (0=low, 1=mid, 2=high) | Rolling percentile rank |
| vol_zscore | How extreme is current vol vs history | Z-score of rolling vol |
| vol_of_vol | Volatility of volatility | Std of rolling vol |

## Annualization Factors

Volatility is annualized differently per timeframe:

- 1h: sqrt(8760) = 93.6
- 24h: sqrt(365) = 19.1
- 168h: sqrt(52) = 7.2

# 6. Regime Detection Features

The regime detection system identifies market states (calm vs volatile, trend vs range) using a hybrid approach combining structural (BinSeg) and momentum (MSM) regimes.

## Regime Detection Methods

| Method | Accuracy | Avg Duration | Characteristics |
|---|---|---|---|
| BinSeg (Binary Segmentation) | ~54% | 49 days | Structural shifts, slow |
| MSM (Markov Switching) | ~69% | 4.2 days | Momentum regimes, fast |
| Hybrid (BinSeg + MSM) | ~74% | 3.9 days | Best of both |

## Hybrid Regime States

| State | BinSeg | MSM | Description | Strategy |
|---|---|---|---|---|
| 0 | Calm | Calm | Quiet consolidation | DANGER if bearish |
| 1 | Calm | Volatile | Breakout (BEST returns) | Full position |
| 2 | Volatile | Calm | Recovery phase | Reduced position |
| 3 | Volatile | Volatile | Crisis/momentum | Full position |

# 7. Multi-Timeframe Features

Features are computed independently for each of 6 timeframes, then combined into a unified feature matrix aligned to the 1-hour index.

## Timeframe Hierarchy

| Timeframe | Role | Features Computed |
|---|---|---|
| 1h | Execution timing | All base features |
| 4h | Intraday trends | All base features |
| 12h | Session analysis | All base features |
| 24h | Tactical regime | All base features |
| 72h | Multi-day trends | All base features |
| 168h | Structural regime | All base features |

## Multi-TF Value Proposition

**1. Better Volatility Prediction:** 1h spike + 168h calm = temporary (don't panic) vs 1h spike + 168h rising = structural shift (take action).

**2. Earlier Transition Detection:** Cross-TF disagreement = regime change brewing. If 1h turns bearish while 168h still bullish, watch for confirmation.

**3. Smarter Position Sizing:** Vol direction, not just level. Rising vol from low = reduce size. Falling vol from high = can add.

# 8. Window Scaling

A critical insight: window sizes must scale with timeframe to measure comparable market phenomena. A 168h bar already contains 168 hours of data; computing a 14-bar MA on top creates 2,352 hours of lookback.

## Standard Window Scale (target ~24h lookback)

| Timeframe | Window Size | Hours Covered |
|---|---|---|
| 1h | 24 bars | 24 hours |
| 4h | 6 bars | 24 hours |
| 12h | 2 bars | 24 hours |
| 24h | 1 bar | 24 hours |
| 72h | 1 bar | 72 hours (minimum) |
| 168h | 1 bar | 168 hours (minimum) |

## Extended Window Scale (target ~168h lookback)

| Timeframe | Window Size | Hours Covered |
|---|---|---|
| 1h | 168 bars | 168 hours (1 week) |
| 4h | 42 bars | 168 hours |
| 12h | 14 bars | 168 hours |
| 24h | 7 bars | 168 hours |
| 72h | 3 bars | 216 hours |
| 168h | 2 bars | 336 hours |

## Impact of Window Scaling

Before scaling: 21 days warmup, 104/4,345 complete rows (2.4%)
After scaling: 11 days warmup, 171/4,345 complete rows (3.9%)

A 65% improvement in usable data, plus 14x faster compute time.

# 9. Cross-Timeframe Features

Cross-timeframe features detect alignment or divergence between timeframes, providing early warning of regime transitions.

| Feature | Description |
|---|---|
| x_ma_1h_vs_24h | Trend alignment: +1 (aligned), -1 (diverged), 0 (mixed) |
| x_ma_1h_vs_168h | Short-term vs structural trend alignment |
| x_ma_24h_vs_168h | Daily vs weekly trend alignment |
| x_price_position | Average position of price vs all TF MAs |
| x_price_position_std | Std of price positions (TF agreement) |
| x_vol_1h_vs_168h | Short-term vol vs structural vol ratio |
| x_regime_agree | Number of TFs in same regime state |
| x_regime_transition | Detected regime transition brewing |

## Interpretation

**High x_regime_agree (5-6):** Strong consensus across timeframes. Trend is established.

**Low x_regime_agree (1-2):** Timeframes disagree. Transition likely brewing.

**x_ma divergence:** When short-term crosses against long-term, early reversal signal.

# 10. Feature Engine Usage

## Basic Usage

```
from cryptobot.features import FeatureEngine

engine = FeatureEngine()

# Compute specific features
df = engine.compute(ohlcv_df, ['ma_score', 'rolling_vol_168h'])

# Compute strategy features
df = engine.compute_strategy_features(ohlcv_df)

# Compute feature group
df = engine.compute_group(ohlcv_df, 'trend')
```

## Strategy Features

The default strategy feature set (STRATEGY_FEATURES):

- ma_score, price_vs_sma_6, price_vs_sma_24, price_vs_sma_72
- rolling_vol_168h, garch_vol_simple
- regime_binseg, regime_msm, regime_hybrid_simple, regime_multiplier

## Database Integration

```
# Compute and save to database
result = engine.compute_and_save('XBTUSD', start='2020-01-01')

# Load cached or compute fresh
df = engine.load_or_compute('XBTUSD', force_recompute=False)
```

# 11. Feature Analysis Tools

## Correlation Analysis

```
from cryptobot.features.features import analyze_features

analysis = analyze_features(df, target_col='target')
print_analysis(analysis)
```

## Feature Importance

```
from cryptobot.features.features import calculate_feature_importance

importance = calculate_feature_importance(
df, feature_cols, target_col,
method='all' # 'random_forest', 'mutual_info', 'correlation'
)

# Select top features
selected = select_features(importance, method='top_n', n=15)
```

## Finding Redundant Features

```
from cryptobot.features.features import find_highly_correlated

# Find features with >80% correlation
redundant_pairs = find_highly_correlated(df, feature_cols, threshold=0.80)
```