گزارش جامع فنی و استراتژیک
ربات معاملاتی Python + MetaTrader 5

نسخه بررسی‌شده: v0.5.4 | دارایی‌ها: XAUUSD و XAGUSD
و نقشه راه بهبود ML ،گزارش ممیزی معماری، استراتژی، بک‌تست، ریسک

1. خلاصه اجرایی

ارزیابی کلی :ربات از نظر مهندسی نرم‌افزار، کنترل ریسک، ضد نشت داده، بک‌تست و قابلیت توسعه پایه خوبی دارد؛ اما بخش اصلی استراتژی معاملاتی هنوز نسبتاً ساده است .مهم‌ترین فرصت رشد، بازنویسی زیرساخت نیست، بلکه ارتقای Strategy Core، تشخیص ساختار بازار، Liquidity، Feature Engineering و مدل Meta-Labeling است.

بخش	امتیاز	نتیجه
معماری نرم‌افزار	5/5	ماژولار و قابل توسعه
کنترل ریسک	5/5	Position Sizing و Circuit Breaker مناسب
ضد Data Leakage	5/5	طراحی علّی و ورود کندل بعدی
بک‌تست	4/5	event-driven و اجرای Spread/Slippage
Walk-Forward	4/5	وجود Purged/WFO و ابزار تحلیل
MT5 Integration	4/5	و اجرای زنده مناسب Adapter
Regime Detection	4/5	خوب برای نسخه پایه، نیازمند توسعه
Strategy Core	2.5/5	ساده و فیلترهای محدود FVG
AI/ML	2.5/5	Meta model خوب است، Feature ها محدودند
قابلیت توسعه	5/5	ساختار برای نسخه‌های بعدی مناسب است

2. معماری فعلی

ساختار مفهومی فعلی به‌صورت زیر است:

MT5 Data → XAU/XAG → Regime Engine → H4 Bias + M15 → FVG Strategy → Session Gate → Risk Engine → Circuit Breaker → MT5 → Journal → WFO/ML

این معماری از نظر جداسازی مسئولیت‌ها خوب است و بهتر است حفظ شود .توسعه باید عمدتاً در لایه Strategy/Features/Portfolio انجام شود.

2.1. ناسازگاری مهم Config و Live Runner

در تنظیمات پروژه Triple Screen با decision timeframe=M15، trend timeframe=H1 و entry timeframe=M5 تعریف شده است، مستند شده با Config را به‌عنوان مسیر اصلی استفاده می‌کند .بنابراین چیزی که در M15 + H4 bias + FVG اما اجرای فعلی عملاً چیزی که Live Runner انجام می‌دهد یکسان نیست.

باید این قرارداد معماری یکسان شود؛ در غیر این صورت ممکن است مدل ML یا Strategy اولویت اصلاح :بسیار بالا .قبل از بهینه‌سازی دقیقاً یکسان نیست Live و بک‌تست برای سیستمی بهینه شوند که با.

3. نقاط قوت

- طراحی ماژولار و قابل توسعه بدون نیاز به بازنویسی کامل پروژه.
- رعایت علیت در سیگنال و جلوگیری از استفاده از آینده در تصمیم‌گیری.
- ورود در Open روی فرض ورود به‌جای کندل بعدی Close همان کندل.

- مدل‌سازی Spread و Slippage در بک‌تست.
- وجود Event-driven Backtester و ابزارهای Walk-Forward.
- Circuit Breaker چندمرحله‌ای برای کنترل افت سرمایه.
- Journal مناسب برای ساخت دیتاست واقعی معاملات آینده.
- با پروفایل‌های اختصاصی XAGUSD و XAUUSD پشتیبانی جداگانه از.
- بگیرد Buy/Sell تصمیم Raw Candle به‌جای اینکه مدل مستقیماً از Meta-Labeling استفاده از.
- وجود تست‌های regression/causalityدر بررسی پروژه 129 تست پاس شده‌اند؛.

4. نقاط ضعف اصلی

- FVG فعلی تقریباً به یک الگوی سه‌کندلی ساده محدود است و کیفیت Setup را به‌اندازه کافی نمی‌سنجد.
- Liquidity Sweep، MSS/BOS/CHoCH و Displacement به‌عنوان اجزای اصلی Setup در Strategy Core وجود ندارند یا نقش کافی ندارند.
- Multi-Timeframe واقعی H4/H1/M15/M5 در Live هنوز کامل پیاده نشده است.
- Regime Engine عمدتاً به ADX/EMA/ATR و High Volatility متکی است و Chaos را بیش از حد به نزدیک می‌کند.
- Stop Loss فعلی بیشتر mechanical است و می‌تواند با Structure و Volatility سازگارتر شود.
- TP 2.5ثابت R با انعطاف کمی دارد و باید Structure/Expected R مقایسه شود.
- ML و Expected R بیشتر احتمال برد را می‌سنجد و را به‌صورت مستقیم مدل نمی‌کند.
- Featureهای SMC/ICT و Liquidity در Meta Model کم هستند.
- XAU و XAG رفتار یکسان ندارند و باید پارامترهای microstructure و session behavior آن‌ها جداگانه یاد گرفته شود.
- ریسک XAU و XAG باید در سطح Portfolio در بعضی شرایط بالا باشد، چون همبستگی آن‌ها می‌تواند مشترک کنترل شود.
- Thresholdهای Spread ثابت بهتر است به Spread Percentile یا Spread Ratio همان معمول نسبت به وضعیت Session تبدیل شوند.
- Session Gate بهتر است به‌جای یک پنجره زمانی ساده، پروفایل آماری Session داشته باشد.

5. تحلیل Strategy Core

5.1. FVG فعلی

FVG ممکن است کم‌اهمیت جدا نمی‌کند. دو Gap باکیفیت را از یک Setup ساده برای نسخه اولیه مناسب است، اما به‌تنهایی FVG از نظر کد یکسان باشند ولی از نظر ساختار بازار کاملاً متفاوت باشند.

پیشنهاد: FVG فقط آخرین حلقه Setup باشد، نه کل Setup.

Liquidity Sweep → Displacement → MSS/BOS → FVG → Retrace → Entry

5.2. Strategyهای پیشنهادی

Regime	Strategy	منطق
TREND_UP/DOWN	Trend Continuation	HTF bias + Liquidity Sweep + MSS/BOS + Displacement + FVG
RANGE	Mean Reversion	Range High/Low + Liquidity Grab + Rejection + برگشت به Mean
Compression/Breakout	Breakout	Volatility Compression + Liquidity Build-up + Displacement + Retest
CHAOS	No Trade	حذف معاملات در شرایط واقعاً غیرعادی

6. Multi-Timeframe پیشنهادی

ساختار پیشنهادی:

H4 = جهت کلان و مناطق مهم
H1 = و ساختار میانی Bias تاییدکننده

M15 = Setup Detection و Market Structure
M5 = Entry Trigger و بهینه‌سازی قیمت ورود

نکته: اگر M5 فقط برای Trigger و بهینه‌سازی قیمت ورود است، باید در بک‌تست Live دقیقاً همان داده و همان منطق استفاده شود. در غیر این صورت Triple Screen صرفاً روی کاغذ باقی می‌ماند.

7. Market Structure Engine

- تشخیص Swing High / Swing Low با تعریف بدون نگاه به آینده.
- HH / HL / LH / LL.
- BOS برای ادامه ساختار.
- CHoCH برای تغییر احتمالی رژیم.
- MSS برای تأیید تغییر کوتاه‌مدت ساختار.
- اندازه حرکت Displacement نسبت به ATR.
- ثبت فاصله قیمت تا آخرین Structure برای Feature Engineering.

تمام این ویژگی‌ها باید با زمان ایجادشان ثبت شوند تا Data Leakage ایجاد نشود.

8. Liquidity Engine

- Previous Day High / Low.
- Previous Week High / Low در صورت مناسب بودن.
- Asian High / Low.
- Session High / Low.
- Equal High / Equal Low.
- Liquidity Sweep و مقدار نفوذ در Liquidity.
- Distance to Liquidity و زمان سپری‌شده از Sweep.

9. Setup Scoring

به‌جای Binary Signal، برای هر Setup نمونه مفهومی. یک امتیاز کیفیت بسازید:

HTF Alignment + Liquidity Sweep + MSS/BOS + Displacement + FVG Quality + Session Quality + Volatility Quality + Spread Quality

بررسی کنید کدام Out-of-Sample و WFO ها را ذخیره کنید، سپس با Feature امتیاز نباید صرفاً با دست انتخاب شود. ابتدا Feature ها واقعاً ارزش افزوده دارند.

10. Regime Engine

نسبت به تاریخچه Percentile به‌صورت Volatility پایه خوبی دارد، اما بهتر است ATR Ratio و Slope، EMA، ADX نسخه فعلی با اندازه‌گیری شود/Session/همان نماد.

وضعیت	تعریف پیشنهادی
Low Volatility	ATR percentile < 20%
Normal	20% تا 60%
High	60% تا 85%
Extreme	>85%

High Volatility الزاماً Chaos نیست. Chaos بهتر است ترکیبی از Volatility Shock + Spread Shock + Candle Anomaly + Liquidity Abnormality باشد.

11. Stop Loss

Structure-based SL با Buffer مبتنی بر ATR: ثابت طراحی شده است. پیشنهاد اصلی Buffer فعلی بر اساس چند کندل اخیر و SL نمونه: SL = Swing/Liquidity Level ± k × ATR

مقدار k باید با Walk-Forward تست شود و برای XAU/XAG الزاماً یکسان نباشد.

12. Take Profit

- Fixed RR را به‌عنوان Baseline حفظ کنید.
- 2R، 2.5R و 3R را مقایسه کنید.
- Structure Target مانند Previous High/Low و Liquidity Target را تست کنید.
- Hybrid Target مانند min(2.5R, Liquidity Target) را بررسی کنید.
- به‌جای Win Rate تنها، Expected R و Profit Factor را معیار تصمیم قرار دهید.

13. Break-Even و مدیریت معامله

باعث تبدیل BE برگردد و Entry چندین بار به TP فعال شود. ممکن است قیمت قبل از MFE/MAE نباید بدون تحلیل Break-Even به صفر شود Winner.

ابتدا MFE/MAE معاملات را استخراج کنید و سپس BE Trigger و Trailing را در WFO مقایسه کنید.

14. ML / Meta-Labeling

اصل Meta-Labeling فعلی ارزش حفظ کردن دارد: Strategy Candidate و ML کیفیت آن Candidate توصیه. را تخمین می‌زند نمی‌شود در این مرحله مدل را به Raw Candle → Buy/Sell تبدیل کنید.

14.1. Featureهای پیشنهادی

- H4/H1/M15/M5 trend alignment.
- BOS/CHoCH/MSS.
- Liquidity Sweep.
- Distance to PDH/PDL و Session High/Low.
- FVG size / ATR.
- FVG age و Fill percentage.
- Displacement score.
- Candle body/wick ratios.
- Volume z-score در صورت معتبر بودن Volume دیتای بروکر.
- ATR percentile.
- Spread percentile.
- Session و Session overlap.
- Distance to Daily Open.
- Previous trade state فقط در صورتی که تعریف علّی و بدون Leakage داشته باشد.

14.2. مدل پیشنهادی

Gradient Boosting/GBM و Logistic Regression کافی هستند. Neural Network بزرگ تا زمانی که دیتاست واقعی در مرحله فعلی بزرگ و پایدار ندارید اولویت نیست.

Spread/Slippage پس از هزینه Expected R نیز تخمین زده شود. معیار نهایی می‌تواند P(win)، Expected R بهتر است علاوه بر باشد.

15. معیار تصمیم‌گیری

نقطه سربه‌سر نظری بدون هزینه حدود 28.6 %است. بنابراین سیستم RR=2.5، به‌تنهایی معیار مناسبی نیست. برای Win Rate باید به Profit Factor، Expectancy، Expected R، Max Drawdown، Calmar/Sortino و Stability در OOS توجه کند.

نمونه: P(win)=48% با RR ممکن است مناسب سودده باشد؛ P(win)=60% اگر هم Average Loss بزرگ باشد الزاماً سودده نیست.

16. XAGUSD در برابر XAUUSD

تفکیک پروفایل‌های XAU/XAG اما تصمیم درستی است، Strategy مشترک باشد مشترک نباید به معنای پارامترهای کاملاً مشترک باشد.

- ATR و volatility profile جداگانه.
- Spread profile جداگانه.
- Session behavior جداگانه.
- SL/TP buffer جداگانه.
- حداکثر lot جداگانه و محدودیت نقدشوندگی.
- Thresholdهای Setup با WFO جداگانه.

جمع XAG بیشتری برای WFO و OOS در نظر گرفت تا زمانی که Experimental را XAG و Core را می‌توان XAU، در وضعیت فعلی شود.

17. Portfolio Risk برای دو دارایی

XAU و XAG را از نظر ریسک کاملاً مستقل فرض نکنید. در بعضی شرایط همبستگی آن‌ها زیاد می‌شود و دو معامله هم‌جهت مشترک بسازند Exposure می‌توانند یک ریسک معامله دوم، correlation تعریف کنید؛ مثلاً مجموع ریسک باز محدود باشد و در صورت افزایش Portfolio Risk Cap: پیشنهاد کاهش یابد.

نمونه مفهومی: Correlation بالا → Risk Multiplier پایین‌تر برای معامله جدید.

18. Spread و Microstructure

Spread ثابت و Threshold بهتر است به وضعیت عادی همان نماد Session نرمال شود.

Spread Ratio = Current Spread / Median Session Spread

نمونه مفهومی: 0.8 خوب، 1.0 نرمال، 1.5 گران، 2.0 رد. این اعداد فقط نقطه شروع تست هستند و نباید به‌عنوان مقدار نهایی فرض شوند.

19. Session Engine

به‌جای یک پنجره زمانی ثابت، Sessionها را جدا کنید:

- London Open
- London
- NY Open
- London/NY Overlap
- NY
- Late NY

هایی Session را جداگانه اندازه‌گیری کنید. سپس فقط Volatility و Spread، Average R، Profit Factor، Win Rate، Session برای هر که در OOS پایدار هستند وزن بالاتری بگیرند.

20. Journal و دیتاست آینده

Journal فعلی یکی از مهم‌ترین دارایی‌های پروژه است. آن را به دیتاست آموزشی تبدیل کنید.

- symbol, direction, entry, SL, TP, result, R
- regime, session, HTF bias
- BOS/CHoCH/MSS
- liquidity sweep
- FVG size/age/fill
- ATR percentile
- spread percentile
- MFE/MAE
- distance to liquidity
- displacement score
- model probability
- expected R
- reason for rejection/acceptance

پس از چند صد تا چند هزار معامله با کیفیت و برچسب‌گذاری ثابت، این دیتاست ارزش بسیار بیشتری برای ML خواهد داشت.

21. بک‌تست و Validation

- Event-driven Backtester فعلی را حفظ کنید.
- Spread و Slippage را حفظ کنید.
- Entry on next candle را حفظ کنید.
- Causality tests را حفظ کنید.
- Purged Walk-Forward را حفظ کنید.
- هر تغییر Strategy باید در In-Sample، Validation و Out-of-Sample بررسی شود.
- Optimize پارامترهای زیاد را همزمان نکنید.
- به دنبال Stability Across Windows باشید، نه بهترین یک بازه.

Strategy تست پاس‌شده نشانه خوبی برای سلامت نرم‌افزار است، اما به‌هیچ‌وجه اثبات سوددهی آینده نیست. 129

22. چیزهایی که فعلاً نباید تغییر کند

- Architecture ماژولار
- MT5 Adapter
- Event-driven Backtester
- Walk-Forward/Purged WFO
- Circuit Breaker
- Journal
- Anti-Leakage Design
- Spread/Slippage Modeling

این اجزا زیرساخت هستند و نباید با هر تغییر Strategy دوباره بازنویسی شوند.

23. نقشه راه اجرایی پیشنهادی

مرحله	کار	اولویت	هدف
P0	یکسان‌سازی Config و Live Runner	بحرانی	جلوگیری از اختلاف Backtest/Live
P1	واقعی Multi-Timeframe H4/H1/M15/M5	خیلی بالا	دقیق‌تر Entry و Bias

P2	Market Structure Engine	خیلی بالا	BOS/CHoCH/MSS
P3	Liquidity Engine	خیلی بالا	PDH/PDL/Session/Equal High-Low/Sweep
P4	SMC Setup Engine	خیلی بالا	Sweep + Displacement + MSS + FVG
P5	سه Strategy بر اساس Regime	بالا	Trend/Range/Breakout
P6	Setup Scoring	بالا	تفکیک Setup های ضعیف و قوی
P7	ML Feature Expansion	بالا	قوی‌تر Meta-Labeling
P8	Expected R Model	بالا	تصمیم بر مبنای ارزش اقتصادی
P9	Portfolio Risk	بالا	کنترل همزمان XAU/XAG
P10	Dynamic SL/TP	متوسط/بالا	سازگاری با Structure/Volatility
P11	Session/Spread Profiles	متوسط	بهبود کیفیت Execution
P12	Live Paper/Demo Validation	بحرانی	بررسی اختلاف بازار واقعی و بک‌تست

24. ترتیب توسعه پیشنهادی کد

1. اول قرارداد Data و Timeframeها را ثابت کنید.
2. MarketStructureEngine را اضافه کنید.
3. LiquidityEngine را اضافه کنید.
4. FVG را به یک Component از Setup تبدیل کنید.
5. Trend/Range/Breakout Strategyها را جدا کنید.
6. Regime Engine را به Router تبدیل کنید.
7. FeatureExtractor واحد بسازید تا همان Feature در Backtest و Live تولید شود.
8. Meta Model را با Featureهای جدید آموزش دهید.
9. Expected R model را اضافه کنید.
10. PortfolioRiskEngine را قبل از PositionSizer قرار دهید.
11. SL/TP dynamic را با WFO آزمایش کنید.
12. تمام نتایج را در Journal استاندارد ذخیره کنید.
13. بعد از OOS پایدار، Demo Forward Test انجام دهید.
14. تنها پس از Forward Test موفق، ریسک Live را به‌تدریج افزایش دهید.

25. معماری هدف

Market Data → Multi-Timeframe Engine → Regime Engine → Strategy Router → Market Structure + Liquidity → Setup Scoring → Meta Model (Pwin + Expected R) → Portfolio Risk → Position Sizing → Circuit Breaker → MT5 → Journal → WFO/Postmortem → Learning

در این معماری AI نقش فیلتر و رتبه‌بندی Setup را دارد، نه پیشگویی مستقیم قیمت. این رویکرد هم قابل توضیح‌تر است و هم برای جلوگیری از Overfitting منطقی‌تر.

26. چک‌لیست قبل از Live

- Backtest و Live دقیقاً از یک Strategy Code استفاده کنند.
- Feature generation در Backtest و Live یکسان باشد.
- هیچ Feature از آینده استفاده نکند.
- Spread/Slippage در بک‌تست واقع‌گرایانه باشد.
- OOS مستقل وجود داشته باشد.
- حداقل چند پنجره WFO نتایج نسبتاً پایدار بدهند.
- XAU و XAG جداگانه ارزیابی شوند.

- Portfolio Risk Cap فعال باشد.
- Circuit Breaker فعال باشد.
- Kill Switch دستی و خودکار تست شده باشد.
- Journal کامل باشد.
- Demo Forward Test قبل از Live انجام شود.

27. نتیجه نهایی

این پروژه ارزش توسعه دارد و لازم نیست از صفر نوشته شود. Strategy Core نه، کیفیت و عمق مهم‌ترین مشکل فعلی. بهترین مسیر این است که Python، MT5 یا معماری کلی FVG ساده را به یک Setup چندمرحله‌ای بر پایه Liquidity + Market Structure + Displacement + FVG سپس، ML تبدیل کنید و Meta-Labeler را به‌عنوان روی همین Setup‌های ساختاریافته قرار دهید.

مبتنی بر Liquidity Sweep + MSS/BOS + Displacement + FVG، 3) ارتقای ML از صرفاً P(win) به P(win) + Expected R + Position Sizing.

واقعی، و اثبات Edge هدف این نسخه نباید «بیشتر معامله کردن» باشد. هدف باید «حذف معاملات کم‌کیفیت، حفظ معاملات با پایداری Edge در Out-of-Sample و Forward Test» باشد.

پایان گزارش