The 1SSHSHFT.*rp1*: Is Simultaneous Stopout Hedging Strategy, a High Frequency Trading, an Edge in Retail Level Application?

Albeos Rembrant[1,3]*, Lanticse Adrian [2,3]

[1]Founder & Chief Executive Officer, [2]Co-Founder

[3]Wildmind Quasars, Mangangalakal ng Kumikinang na Ginto

https://github.com/algorembrant/QAT-QuantitativeAlgorithmicTrading

November 24, 2025

Abstract

This study presents the development and evaluation of an Expert Advisor (EA) for MetaTrader 5 (MT5), implemented in MQL5, that applies a simultaneous stop-out hedging strategy by placing concurrent buy and sell orders around the prevailing XAUUSD price. Parameter testing identified an optimal configuration consisting of a 50-pip hedge distance, a 100-pip stop loss, a 1000-pip take profit, and a trailing stop that advances the stop loss by 40 pips for every 50-pip favorable price movement. This design accounts for an average spread of approximately 20 pips and anticipated slippage of up to 10 pips.

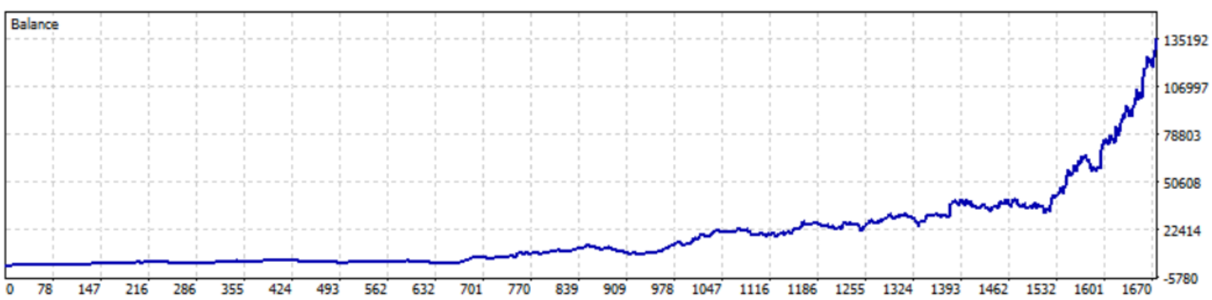


Figure 1: *The Best Demo performance*

The EA was evaluated through multiple demo tests under varying market conditions and showed its strongest performance during periods of high volatility and sustained directional movement. Six demo runs were conducted using an initial capital of USD 1,000. A 12-hour stress test produced a return of 3,067%, a 63.30% win rate, and a maximum drawdown of 23.96%. A subsequent demo run yielded extreme and unrealistic results and was excluded from analysis, highlighting the limitations of demo environments, which do not accurately reflect real-market liquidity, latency, or execution constraints.

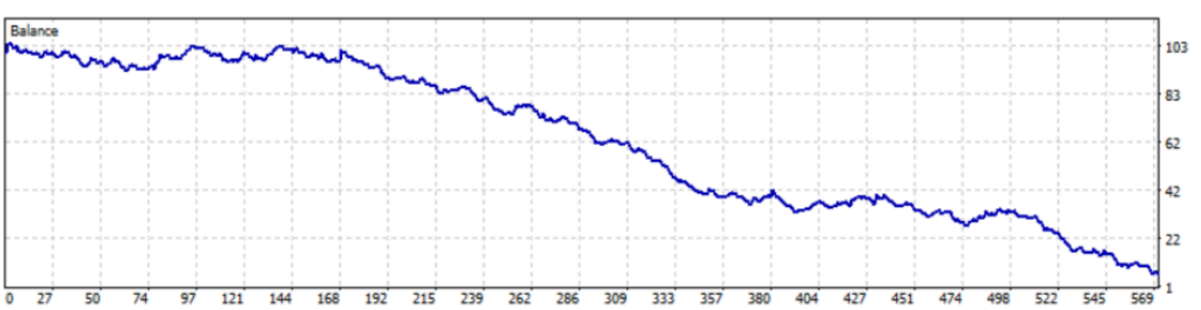


Figure 2: *The Best Live performance*

To assess real-world viability, two live tests were conducted with durations of seven hours and five hours. Although both live tests recorded moderate win rates of 53.91% and 49.62%, the strategy proved unprofitable, with severe drawdowns of 94% and 80.40%. These losses were driven by a deeply negative reward-to-risk profile caused primarily by execution latency, slippage, and spread expansion. In live conditions, these factors distort intended stop-loss and take-profit levels, rendering the strategy structurally disadvantaged. Sustained profitability is therefore unattainable without substantial improvements in execution speed and slippage mitigation.

Introduction

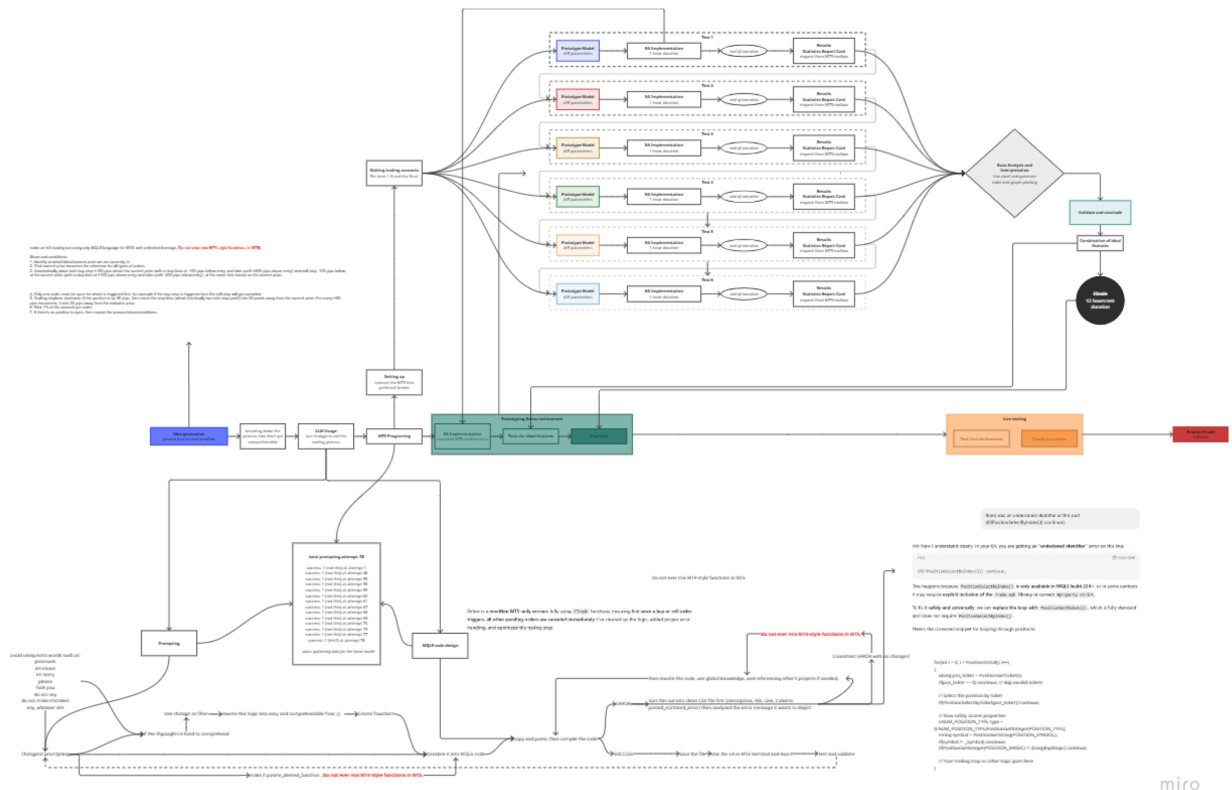
This is the very first paper produced by the company. There is no strict professionalism in making this paper due to a reason that it is only beneficial for the company and not for the outside publication.

Aim of the study

This study aims to develop and evaluate the performance of Simultaneous Stopout Hedging Strategy, a High Frequency Trading (1SSHSHFT.rp1) in both Demo market and Live markets. The results will be compared to see how it differs from each other.

Methodology

Workflow Overview



The Four Phase Methodology

Phase 1: Idea Generation

The methodology begins with the conceptual development of the trading strategy. This phase involves identifying market inefficiencies, defining the core trading logic, and formulating hypotheses regarding potential sources of edge. Strategic assumptions, risk constraints, and target market conditions are established to guide subsequent development.

Phase 2: Prompting and Programming

In this phase, the conceptual strategy is translated into executable logic through structured prompting and systematic programming. The trading rules are implemented in code, resulting in a functional Expert Advisor (EA). This stage focuses on correct order execution logic, parameter definition, and ensuring that the algorithm behaves exactly as intended under predefined conditions.

Phase 3: Demo Environment Testing

The EA is tested extensively in a demo environment to evaluate behavior under simulated market conditions. This phase emphasizes feature extraction, performance analysis, and the identification of optimal parameter combinations. Iterative prototyping is conducted to refine trade management rules, filter ineffective configurations, and stabilize the strategy without exposure to real capital.

Phase 4: Live Environment Testing

The final phase involves deploying the EA in a live trading environment. Fine-tuning and rapid parameter adjustments are performed to address real-world execution issues such as latency, slippage, and spread variability. This stage serves as the definitive validation process, culminating in the finalized EA software trading bot.

Dataset

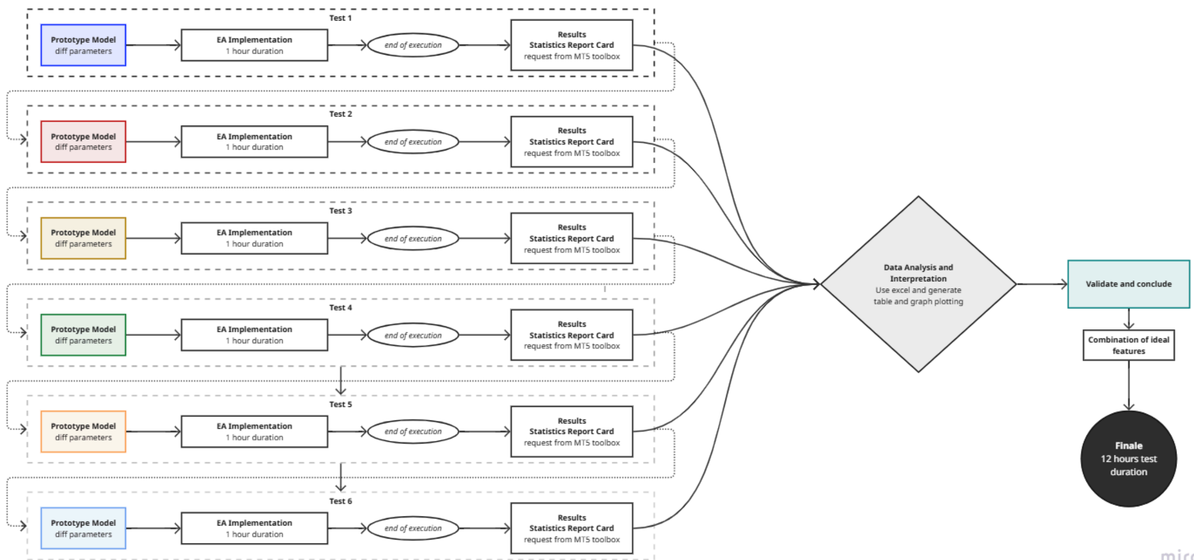
Both live forward testing done in demo and live will be intraday (within a half day 12hrs+).

Trading Logic

If we get a stopout, then hedge by placing a few pips away from the entry, for both sellstop and buystop. The risk per trade is ?%, accounting the full stop loss size, and the take profit is 1R-multiple to infinity.

There is a stoploss trading mechanic attached to the system scalping every profitable movement in the market.

Data gathering framework



Step 0

Creating and programming an EA trading bot using MQL5 script language with hedging mechanics, SL and TP placements, stop loss trailing, and position sizing logic makes sure it risks a fixed amount of percentage yet different lot size per trade, with a millisecond to nanosecond execution depending on the MT5 m/s speed. ***code link***

Step 1

Defining a parameter, set numeric variables for the inputs.

Step 2

Implement and execute the EA Trading bot for 1 hour duration.

Step 3

End of execution.

Step 4

Request statistical report card from MT5 toolbox.

Step 5

Repeat the steps 1 to 4.

Step 6

Generate a table and plot graph using excel spreadsheet for comparison and extraction of ideal features.

Step 7

Execute the final test for 12 hours using the combined ideal features.

Step 8

Final conclusion

Code for Demo Trading

```
//+------------------------------------------------------------------+
//|               #1EATB.mq5                                         |
//|  Auto straddle + hedge on TP/SL, loops indefinitely              |
//+------------------------------------------------------------------+
#property strict
#include <Trade/Trade.mqh>

input double InpRiskPercent = 1.0; // Risk percent per trade
input int    InpPipsEntry = 50;
input int    InpPipsSL    = 100;
input int    InpPipsTP    = 1000;
input int    InpSlippage  = 10;
input ulong  InpMagic     = 7777;
input int    InpTrailTriggerPips = 50;  // Profit pips to start trailing
input int    InpTrailDistancePips = 40; // Trailing distance

CTrade trade;

ulong buyStopTicket  = 0;
ulong sellStopTicket = 0;

//+------------------------------------------------------------------+
double Pip()
{
   double point = SymbolInfoDouble(_Symbol, SYMBOL_POINT);
   int digits = (int)SymbolInfoInteger(_Symbol, SYMBOL_DIGITS);
   return (digits == 3 || digits == 5) ? point * 10.0 : point;
}
//+------------------------------------------------------------------+
double CalculateLotSize(double stopLossPips)
{
   double equity = AccountInfoDouble(ACCOUNT_EQUITY);
   double pipValue = SymbolInfoDouble(_Symbol, SYMBOL_TRADE_TICK_VALUE);
   double lotStep = SymbolInfoDouble(_Symbol, SYMBOL_VOLUME_STEP);
   double minLot = SymbolInfoDouble(_Symbol, SYMBOL_VOLUME_MIN);
   double maxLot = SymbolInfoDouble(_Symbol, SYMBOL_VOLUME_MAX);

   // StopLoss in money
   double slMoney = stopLossPips * pipValue;

   double riskMoney = equity * InpRiskPercent / 100.0;

   double lots = riskMoney / slMoney;

   // Normalize lot to step & min/max
   lots = MathFloor(lots / lotStep) * lotStep;
   if(lots < minLot) lots = minLot;
   if(lots > maxLot) lots = maxLot;

   return(lots);
}
//+------------------------------------------------------------------+
bool PlacePending(ENUM_ORDER_TYPE type, double price, double sl, double tp, ulong &ticket, string comment)
{
   double stopLossPips = (type == ORDER_TYPE_BUY_STOP || type == ORDER_TYPE_BUY_LIMIT) ? (price - sl)/Pip()
                                                                                        : (sl - price)/Pip();
   double lots = CalculateLotSize(stopLossPips);

   MqlTradeRequest req;
   MqlTradeResult  res;
   ZeroMemory(req);
   ZeroMemory(res);

   req.action      = TRADE_ACTION_PENDING;
   req.symbol      = _Symbol;
   req.type        = type;
   req.volume      = lots;
   req.price       = price;
   req.sl          = sl;
   req.tp          = tp;
   req.magic       = InpMagic;
   req.deviation   = InpSlippage;
   req.type_time   = ORDER_TIME_GTC;
   req.type_filling= ORDER_FILLING_FOK;
   req.comment     = comment;

   if(!OrderSend(req, res))
   {
      Print("OrderSend failed: ", res.retcode, " ", res.comment);
      ticket = 0;
      return false;
   }

   ticket = res.order;
   return true;
}
//+------------------------------------------------------------------+
void PlaceStraddle()
{
   double pip   = Pip();
   double ask   = SymbolInfoDouble(_Symbol, SYMBOL_ASK);
   double bid   = SymbolInfoDouble(_Symbol, SYMBOL_BID);
   int digits = (int)SymbolInfoInteger(_Symbol, SYMBOL_DIGITS);

   double buyE  = NormalizeDouble(ask + InpPipsEntry * pip, digits);
   double buySL = NormalizeDouble(buyE - InpPipsSL * pip, digits);
   double buyTP = NormalizeDouble(buyE + InpPipsTP * pip, digits);

   double sellE  = NormalizeDouble(bid - InpPipsEntry * pip, digits);
   double sellSL = NormalizeDouble(sellE + InpPipsSL * pip, digits);
   double sellTP = NormalizeDouble(sellE - InpPipsTP * pip, digits);

   PlacePending(ORDER_TYPE_BUY_STOP, buyE, buySL, buyTP, buyStopTicket, "BUY_STOP_EA");
   PlacePending(ORDER_TYPE_SELL_STOP, sellE, sellSL, sellTP, sellStopTicket, "SELL_STOP_EA");
}
//+------------------------------------------------------------------+
bool PendingOrderExists(ulong ticket)
{
   if(ticket == 0) return false;

   if(OrderSelect(ticket))
      return true;

   return false;
}
//+------------------------------------------------------------------+
bool PositionExists(string type)
{
   for(int i = 0; i < PositionsTotal(); i++)
   {
      ulong ticket = PositionGetTicket(i);
      if(ticket == 0) continue;
      if(!PositionSelectByTicket(ticket)) continue;

      if(PositionGetInteger(POSITION_MAGIC) != (long)InpMagic) continue;
      if(PositionGetString(POSITION_SYMBOL) != _Symbol) continue;

      ENUM_POSITION_TYPE ptype = (ENUM_POSITION_TYPE)PositionGetInteger(POSITION_TYPE);
      if(type == "BUY"  && ptype == POSITION_TYPE_BUY) return true;
      if(type == "SELL" && ptype == POSITION_TYPE_SELL) return true;
   }
   return false;
}
//+------------------------------------------------------------------+
void CancelOrder(ulong &ticket)
{
   if(ticket == 0) return;

   MqlTradeRequest req;
   MqlTradeResult  res;

   ZeroMemory(req);
   ZeroMemory(res);

   req.action = TRADE_ACTION_REMOVE;
   req.order  = ticket;
   req.symbol = _Symbol;

   if(OrderSend(req, res))
   {
      Print("Cancelled pending order: ", ticket);
      ticket = 0;
   }
   else
      Print("Failed to cancel order ", ticket, " ret=", res.retcode);
}
```

```
//+------------------------------------------------------------------+
void CheckAndRebuild()
{
   bool buyPos  = PositionExists("BUY");
   bool sellPos = PositionExists("SELL");

   bool buyActive  = PendingOrderExists(buyStopTicket);
   bool sellActive = PendingOrderExists(sellStopTicket);

   // One triggered → cancel opposite pending
   if(buyPos && sellActive)
      CancelOrder(sellStopTicket);
   if(sellPos && buyActive)
      CancelOrder(buyStopTicket);

   // Loop: if both positions closed → rebuild straddle
   if(!buyPos && !sellPos && !buyActive && !sellActive)
   {
      Print("Both positions closed → placing new straddle");
      PlaceStraddle();
   }

   // Check solo pending orders and cancel them
   if(buyActive && !sellActive)
      CancelOrder(buyStopTicket);
   if(sellActive && !buyActive)
      CancelOrder(sellStopTicket);
}
//+------------------------------------------------------------------+
void TrailStops()
{
   double pip = Pip();

   for(int i=0; i<PositionsTotal(); i++)
   {
      ulong ticket = PositionGetTicket(i);
      if(ticket == 0) continue;
      if(!PositionSelectByTicket(ticket)) continue;

      if(PositionGetInteger(POSITION_MAGIC) != (long)InpMagic) continue;
      if(PositionGetString(POSITION_SYMBOL) != _Symbol) continue;

      double open = PositionGetDouble(POSITION_PRICE_OPEN);
      double current = (PositionGetInteger(POSITION_TYPE) == POSITION_TYPE_BUY) ? SymbolInfoDouble(_Symbol, SYMBOL_BID)
                                                                              : SymbolInfoDouble(_Symbol, SYMBOL_ASK);
      double sl    = PositionGetDouble(POSITION_SL);
      ENUM_POSITION_TYPE type = (ENUM_POSITION_TYPE)PositionGetInteger(POSITION_TYPE);

      double profitPips = (type == POSITION_TYPE_BUY) ? (current - open)/pip
                                                      : (open - current)/pip;

      if(profitPips >= InpTrailTriggerPips)
      {
         double newSL = (type == POSITION_TYPE_BUY) ? current - InpTrailDistancePips*pip
                                                    : current + InpTrailDistancePips*pip;

         // Only move SL forward (do not move backward)
         if((type == POSITION_TYPE_BUY && newSL > sl) || (type == POSITION_TYPE_SELL && newSL < sl))
         {
            trade.PositionModify(ticket, newSL, PositionGetDouble(POSITION_TP));
            Print("Trailing SL modified for ticket ", ticket, " newSL=", newSL);
         }
      }
   }
}
//+------------------------------------------------------------------+
int OnInit()
{
   PlaceStraddle();
   return(INIT_SUCCEEDED);
}
//+------------------------------------------------------------------+
void OnTick()
{
   CheckAndRebuild();
   TrailStops();
}
//+------------------------------------------------------------------+
```

Demo Trading Results

Parameters-Test1

% Risk per trade	10
Pips Entry hedging	100
Pips SL	100
Pips TP	200
Pips Slippage	10
Pips Spread	20
Pips to start trailing	50
Pips trailing distance	40

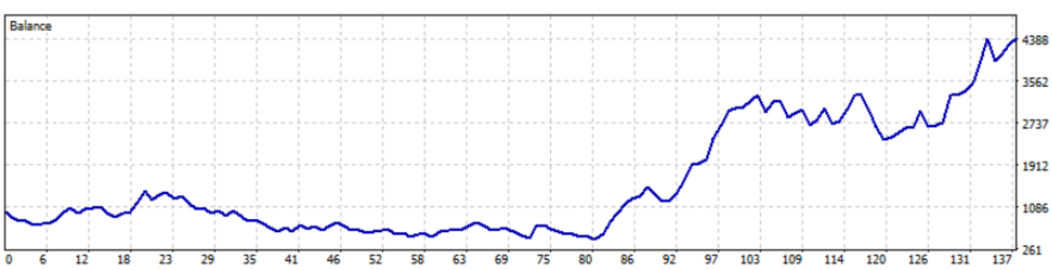
Raw results-Test1

stats link

Trade History Report

Name: **test1**
Account: **270451645 (USD, Exness-MT5Trial17, demo, Hedge)**
Company: **Exness Technologies Ltd**
Date: **2025.11.19 11:30**

Equity: **4 429.13**



Results

Total Net Profit:	3 429.13	Gross Profit:	9 657.09	Gross Loss:	-6 227.96
Profit Factor:	1.55	Expected Payoff:	25.03		
Recovery Factor:	3.60	Sharpe Ratio:	0.15		
Balance Drawdown:					
Balance Drawdown Absolute:	540.44	Balance Drawdown Maximal:	953.13 (67.47%)	Balance Drawdown Relative:	67.47% (953.13)
Total Trades:	137	Short Trades (won %):	74 (71.62%)	Long Trades (won %):	63 (58.73%)
		Profit Trades (% of total):	90 (65.69%)	Loss Trades (% of total):	47 (34.31%)
		Largest profit trade:	548.00	Largest loss trade:	-440.00
		Average profit trade:	107.30	Average loss trade:	-132.51
		Maximum consecutive wins ($):	13 (2 098.94)	Maximum consecutive losses ($):	3 (-894.00)
		Maximal consecutive profit (count):	2 098.94 (13)	Maximal consecutive loss (count):	-894.00 (3)
		Average consecutive wins:	3	Average consecutive losses:	1

Conclusion-Test1

The results of T1 suggest that the 67.47% max drawdown outcome occurred because the entry distances between the hedging orders were too far apart. Therefore, the distance was reduced to 50 pips from the current price for both stop orders. With this adjustment, the subsequent maximal drawdowns in the following tests were indeed reduced.

Parameters-Test2

% Risk per trade	10
Pips Entry hedging	50
Pips SL	100
Pips TP	200
Pips Slippage	10
Pips Spread	20
Pips to start trailing	50
Pips trailing distance	40

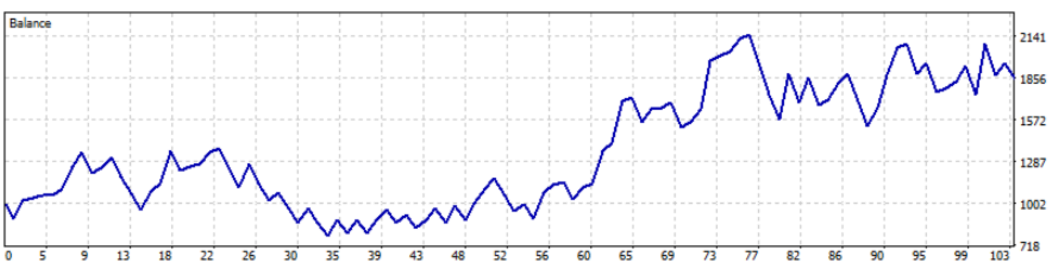
Raw results-Test2

[*stas link*]

Trade History Report

Name: **Test2**
Account: **206675473 (USD, Exness-MT5Trial7, demo, Hedge)**
Company: **Exness Technologies Ltd**
Date: **2025.11.19 14:21**

Equity: **1 849.47**



Results

Total Net Profit:	849.47	Gross Profit:	6 051.53	Gross Loss:	-5 202.06
Profit Factor:	1.16	Expected Payoff:	8.25		
Recovery Factor:	1.36	Sharpe Ratio:	0.11		
Balance Drawdown:					
Balance Drawdown Absolute:	213.70	Balance Drawdown Maximal:	623.89 (28.95%)	Balance Drawdown Relative:	42.88% (590.26)
Total Trades:	103	Short Trades (won %):	60 (63.33%)	Long Trades (won %):	43 (62.79%)
		Profit Trades (% of total):	65 (63.11%)	Loss Trades (% of total):	38 (36.89%)
		Largest profit trade:	348.00	Largest loss trade:	-215.00
		Average profit trade:	93.10	Average loss trade:	-136.90
		Maximum consecutive wins ($):	7 (634.06)	Maximum consecutive losses ($):	3 (-583.00)
		Maximal consecutive profit (count):	693.81 (6)	Maximal consecutive loss (count):	-583.00 (3)
		Average consecutive wins:	2	Average consecutive losses:	1

Conclusion-Test2

The results of T2 suggest that lowering the TP distance could lead to consecutive wins that compound into a higher ending equity balance. However, the improvement turned out to be not as significant as expected. Nonetheless, the drawdown became very low due to the adjusted parameter.

Parameters-Test3

% Risk per trade	10
Pips Entry hedging	50
Pips SL	100
Pips TP	100
Pips Slippage	10
Pips Spread	20
Pips to start trailing	50
Pips trailing distance	40

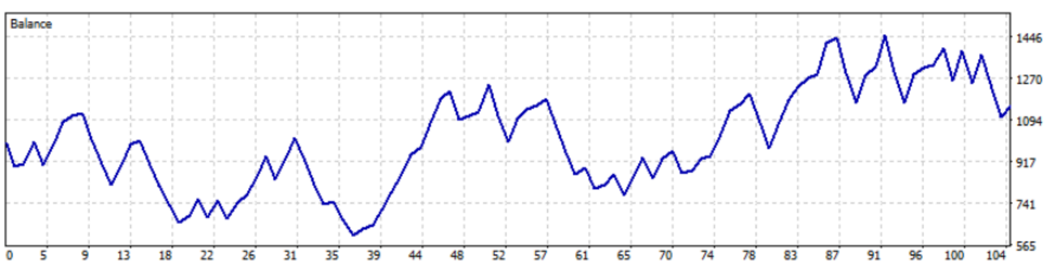
Raw results-Test3

[*stats link*]

Trade History Report

Name: **Test3**
Account: **274899951 (USD, Exness-MT5Trial6, demo, Hedge)**
Company: **Exness Technologies Ltd**
Date: **2025.11.20 15:52**

Equity: **1 159.71**



Results

Total Net Profit:	159.71	Gross Profit:	4 049.00	Gross Loss:	-3 889.29
Profit Factor:	1.04	Expected Payoff:	1.54		
Recovery Factor:	0.31	Sharpe Ratio:	0.06		
Balance Drawdown:					
Balance Drawdown Absolute:	392.71	Balance Drawdown Maximal:	517.09 (45.99%)	Balance Drawdown Relative:	45.99% (517.09)
Total Trades:	104	Short Trades (won %):	54 (66.67%)	Long Trades (won %):	50 (62.00%)
		Profit Trades (% of total):	67 (64.42%)	Loss Trades (% of total):	37 (35.58%)
		Largest profit trade:	132.00	Largest loss trade:	-155.87
		Average profit trade:	60.43	Average loss trade:	-105.12
		Maximum consecutive wins ($):	10 (610.33)	Maximum consecutive losses ($):	4 (-344.00)
		Maximal consecutive profit (count):	610.33 (10)	Maximal consecutive loss (count):	-344.00 (4)
		Average consecutive wins:	3	Average consecutive losses:	2

Conclusion-Test3

The results of T3 suggest that increasing the TP distance can result in a higher profit factor by relying on the SL trailing mechanics.

Parameters-Test4

% Risk per trade	10
Pips Entry hedging	50
Pips SL	100
Pips TP	1000
Pips Slippage	10
Pips Spread	20
Pips to start trailing	50
Pips trailing distance	40

Raw results-Test4

stats link

Trade History Report

Name: **Test4**
Account: **270454941 (USD, Exness-MT5Trial17, demo, Hedge)**
Company: **Exness Technologies Ltd**
Date: **2025.11.20 19:45**

Equity: **106 174.56**

Results

Total Net Profit:	105 174.56	Gross Profit:	345 596.00	Gross Loss:	-240 421.44
Profit Factor:	1.44	Expected Payoff:	409.24		
Recovery Factor:	2.50	Sharpe Ratio:	0.23		
Balance Drawdown:					
Balance Drawdown Absolute:	210.03	Balance Drawdown Maximal:	42 072.25 (52.48%)	Balance Drawdown Relative:	52.48% (42 072.25)
Total Trades:	257	Short Trades (won %):	131 (64.89%)	Long Trades (won %):	126 (73.02%)
		Profit Trades (% of total):	177 (68.87%)	Loss Trades (% of total):	80 (31.13%)
		Largest profit trade:	13 018.28	Largest loss trade:	-8 989.00
		Average profit trade:	1 952.52	Average loss trade:	-3 005.27
		Maximum consecutive wins ($):	12 (5 602.49)	Maximum consecutive losses ($):	4 (-19 963.00)
		Maximal consecutive profit (count):	33 409.63 (6)	Maximal consecutive loss (count):	-19 963.00 (4)
		Average consecutive wins:	3	Average consecutive losses:	1

Conclusion-Test4

The results of T4 suggest that the EA trading bot performs best during highly volatile sessions, particularly during the opening of London hours. Do not be misled by the result showing 46,259.25 in end capital with 169 trades within a 1-hour duration. That outcome is not reliable because demo forward testing has no access to actual liquidity resting at every price level. In reality, it is almost impossible to get filled repeatedly using more than 20+ lots in rapid succession.

Parameters-Test5

the same as test4

% Risk per trade	10
Pips Entry hedging	50
Pips SL	100
Pips TP	1000
Pips Slippage	10
Pips Spread	20
Pips to start trailing	50
Pips trailing distance	40

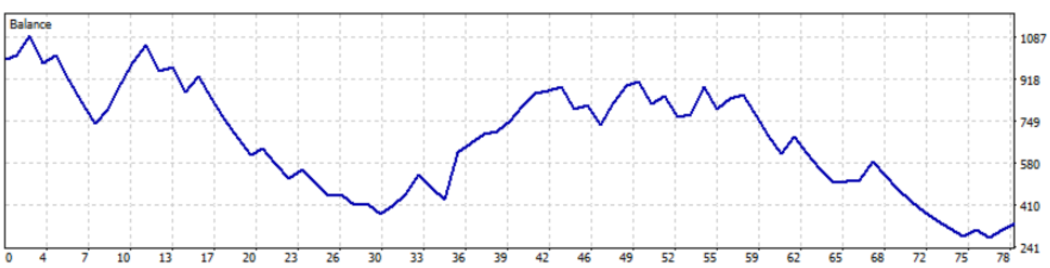
Raw results-Test5

stats link

Trade History Report

Name:	**Test5**
Account:	**270456114 (USD, Exness-MT5Trial17, demo, Hedge)**
Company:	**Exness Technologies Ltd**
Date:	**2025.11.20 19:44**

Equity: **340.69**



Results

Total Net Profit:	**-659.31**	Gross Profit:	**1 852.56**	Gross Loss:	**-2 511.87**
Profit Factor:	**0.74**	Expected Payoff:	**-8.45**		
Recovery Factor:	**-0.81**	Sharpe Ratio:	**-0.09**		
Balance Drawdown:					
Balance Drawdown Absolute:	**718.04**	Balance Drawdown Maximal:	**813.94 (74.27%)**	Balance Drawdown Relative:	**74.27% (813.94)**
Total Trades:	**78**	Short Trades (won %):	**44 (52.27%)**	Long Trades (won %):	**34 (52.94%)**
		Profit Trades (% of total):	**41 (52.56%)**	Loss Trades (% of total):	**37 (47.44%)**
		Largest profit trade:	**190.36**	Largest loss trade:	**-109.00**
		Average profit trade:	**45.18**	Average loss trade:	**-67.89**
		Maximum consecutive wins ($):	**9 (453.34)**	Maximum consecutive losses ($):	**7 (-303.00)**
		Maximal consecutive profit (count):	**453.34 (9)**	Maximal consecutive loss (count):	**-320.98 (4)**
		Average consecutive wins:	**2**	Average consecutive losses:	**2**

Conclusion-Test5

The results of T5 use the same parameters as T4, and they suggest that the EA trading bot loses money and produces a lower trade count during low-volatility periods at the end of the London session before the New York session opens.

Parameters-Test6

% Risk per trade	5
Pips Entry hedging	50
Pips SL	100
Pips TP	1000
Pips Slippage	10
Pips Spread	20
Pips to start trailing	50
Pips trailing distance	40

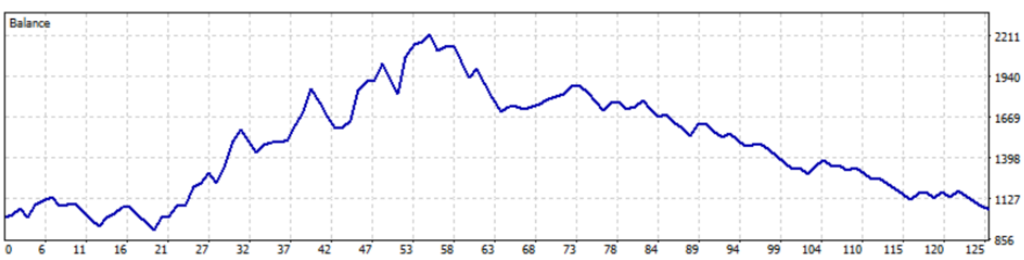
Raw results-Test6

[***stats link***]

Trade History Report

Name: Test6
Account: 272840495 (USD, Exness-MT5Trial14, demo, Hedge)
Company: Exness Technologies Ltd
Date: 2025.11.21 06:28

Equity: 1 051.67



Results

Total Net Profit:	51.67	Gross Profit:	3 144.18	Gross Loss:	-3 092.51
Profit Factor:	1.02	Expected Payoff:	0.41		
Recovery Factor:	0.04	Sharpe Ratio:	0.03		
Balance Drawdown:					
Balance Drawdown Absolute:	78.97	Balance Drawdown Maximal:	1 172.64 (52.72%)	Balance Drawdown Relative:	52.72% (1 172.64)
Total Trades:	125	Short Trades (won %):	67 (55.22%)	Long Trades (won %):	58 (56.90%)
		Profit Trades (% of total):	70 (56.00%)	Loss Trades (% of total):	55 (44.00%)
		Largest profit trade:	242.43	Largest loss trade:	-111.00
		Average profit trade:	44.92	Average loss trade:	-56.23
		Maximum consecutive wins ($):	7 (423.97)	Maximum consecutive losses ($):	4 (-170.00)
		Maximal consecutive profit (count):	433.42 (6)	Maximal consecutive loss (count):	-282.00 (3)
		Average consecutive wins:	3	Average consecutive losses:	2

Conclusion-Test6

The results of T6 suggest that launching the bot during the HMR zone is not ideal. A lower bet size combined with the highest volatility conditions tends to lead only to breakeven.

Demo Discussion

Overview

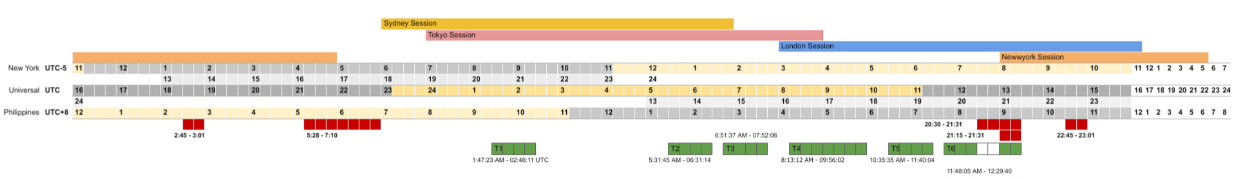
[*stats link*]

Parameters	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
risk per trade %	*10*	*10*	*10*	*10*	*10*	*5*
pips entry hedging	*100*	*50*	*50*	*50*	*50*	*50*
pips sl	*100*	*100*	*100*	*100*	*100*	*100*
pips tp	*200*	*200*	*100*	*1000*	*1000*	*1000*
pips slippage	*10*	*10*	*10*	*10*	*10*	*10*
pips spread	*20*	*20*	*20*	*20*	*20*	*20*
pips to start trailing	*50*	*50*	*50*	*50*	*50*	*50*
pips trailing distance	*40*	*40*	*40*	*40*	*40*	*40*

Results Overview						
start capital	*1000*	*1000*	*1000*	*1000*	*1000*	*1000*
end capital	*4 429.13*	*1 849.47*	*1 159.71*	*106 174.56*	*340.69*	*1 051.67*
trades	*137*	*103*	*104*	*257*	*78*	*125*
start time	*9:47:23 AM*	*1:31:45 PM*	*2:51:37 PM*	*4:13:12 PM*	*6:35:35 PM*	*7:48:05 - 8:29:40*
end time	*10:46:11 AM*	*2:31:14 PM*	*15:52:06*	*17:56:02*	*19:40:04*	*9:05:24 - 9:24:16*
duration	*0:58:48*	*0:59:29*	*1:00:29*	*1:42:50*	*1:04:29*	*1:00:27*
profit factor	*1.55*	*1.16*	*1.04*	*1.44*	*0.74*	*1.02*
recovery factor	*3.6*	*1.36*	*0.31*	*2.5*	*-0.81*	*0.04*
expected payoff	*25.03*	*8.25*	*1.54*	*409.24*	*-8.45*	*0.41*
sharpe ratio	*0.15*	*0.11*	*0.06*	*0.23*	*-0.09*	*0.03*
max drawdown %	*67.47*	*28.95*	*45.99*	*52.48*	*74.27*	*52.72*
winrate %	*65.69*	*63.11*	*64.42*	*68.87*	*52.56*	*56.00*
	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**

Timestamp

[*Image link*]



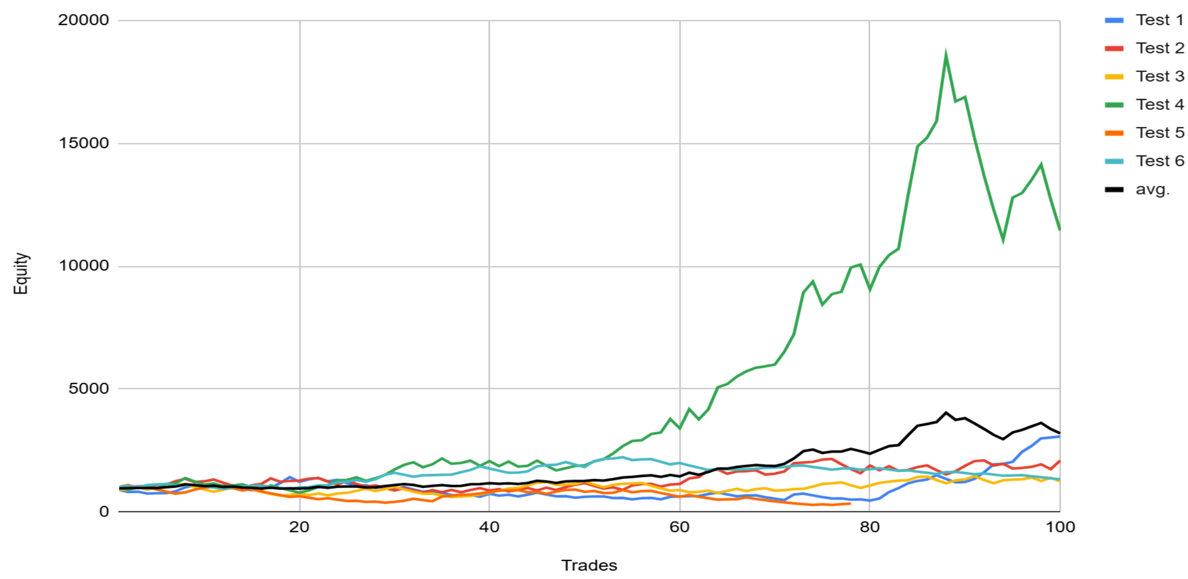
Seasonal sessions cause variations in daylight, which leads many countries to adjust their clocks (DST), resulting in changes in time offsets such as UTC+ or UTC− [*Info link*]. For example, New York was once a UTC-4 but right now is at -5 offset. The red boxes represent HMR zone periods, and the green boxes represent the time where the test was implemented.

Summary

The EA trading bot performs best when using a 50-pip entry hedging distance, a 100-pip stop loss, and a 1000-pip take profit, with the stop loss trailing by 40 pips for every 50-pip movement in favor of the trade. It is most effective when launched during high-volatility periods such as session openings, performs poorly during low-volatility conditions at session endings, and delivers only moderate results during extreme volatility found in HMR zones. When these parameters are combined with a lower bet-size risk percentage, the EA trading bot remains profitable.

Demo Prototype Conclusion

The first 100 trades within 1hour



Comparing the equity curves across these tests is not entirely reliable because each test was launched at a different market session, even though all had a duration of one hour. The first test was launched near the Tokyo market open. The second was launched at the Sydney market close. The third began at the Tokyo market close. The fourth was launched at the London open. The fifth took place during the middle of the London session. The sixth was launched during the High Margin Requirement (HMR) zone while transitioning into the New York session.

These varied launch times expose the EA to different candle behaviors and volatility profiles. Because each session behaves differently, the outcome naturally varies. During the first 100 trades within that hour, the EA only reflects how profitable it can be under those specific, and essentially random, market conditions. This is why the average equity curve is a more meaningful indicator than any single-session result.

Market volatility is unevenly distributed across the trading day. Opening sessions (Tokyo, London, and New York opens) typically show increased volatility due to liquidity influx and institutional order placements. Mid-session periods often exhibit more stable or reduced volatility, while closing sessions can either taper off or spike depending on daily position adjustments. Meanwhile, transition zones like the HMR period can create sudden bursts of volatility as markets overlap or as key economic catalysts approach.

Summary

This uneven distribution of volatility across opening, mid-session, closing, and HMR conditions explains why the outcomes differ so significantly. The EA's performance is therefore heavily influenced by the timing of the test rather than purely by its logic or strategy.

Parameters-Final Test

(12h, 7am to 7pm UTC+8)

% Risk per trade	3
Pips Entry hedging	50
Pips SL	100
Pips TP	1000
Pips Slippage	10
Pips Spread	20
Pips to start trailing	50
Pips trailing distance	40

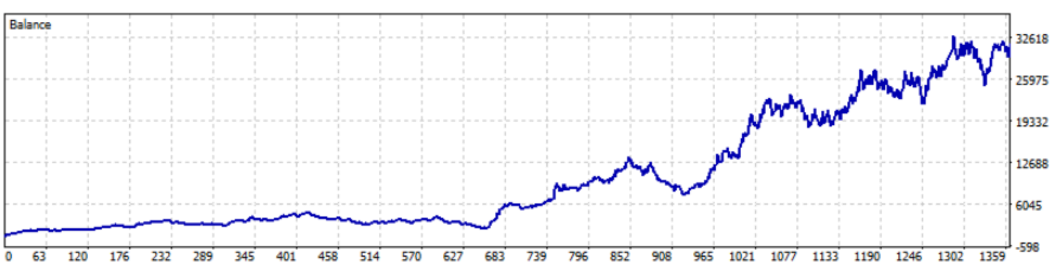
Raw results-Final Test

stats link

Trade History Report

Name: **runallday**
Account: **270461496 (USD, Exness-MT5Trial17, demo, Hedge)**
Company: **Exness Technologies Ltd**
Date: **2025.11.21 19:02**

Equity: **31 859.91**



Results

Total Net Profit:	30 672.76	Gross Profit:	189 845.89	Gross Loss:	-159 173.13
Profit Factor:	1.19	Expected Payoff:	22.60		
Recovery Factor:	3.88	Sharpe Ratio:	0.10		
Balance Drawdown:					
Balance Drawdown Absolute:	0.00	Balance Drawdown Maximal:	7 895.24 (23.96%)	Balance Drawdown Relative:	52.99% (2 568.07)
Total Trades:	1357	Short Trades (won %):	681 (63.58%)	Long Trades (won %):	676 (63.02%)
		Profit Trades (% of total):	859 (63.30%)	Loss Trades (% of total):	498 (36.70%)
		Largest profit trade:	1 995.38	Largest loss trade:	-1 247.24
		Average profit trade:	221.01	Average loss trade:	-319.62
		Maximum consecutive wins ($):	13 (5 522.62)	Maximum consecutive losses ($):	6 (-4 494.10)
		Maximal consecutive profit (count):	5 522.62 (13)	Maximal consecutive loss (count):	-4 494.10 (6)
		Average consecutive wins:	3	Average consecutive losses:	2

Demo Final Conclusion

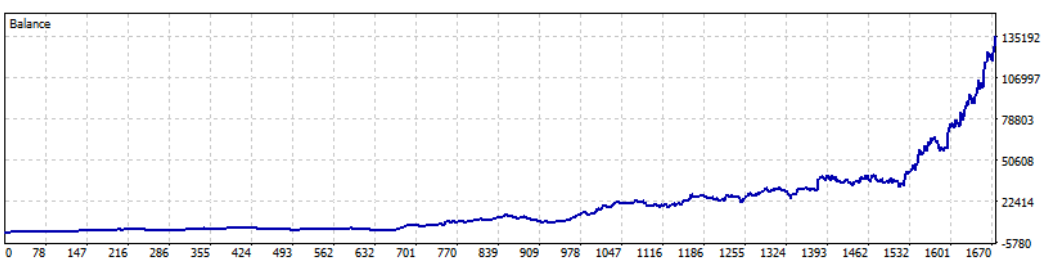
Across the 12-hour window, the EA demonstrated strong profitability but high risk exposure while using 3% risk per trade. The high win rate combined with rapid execution produced significant returns, but the large drawdowns and low Sharpe ratio confirm that the strategy is heavily dependent on volatility conditions and benefits artificially from demo-level liquidity, although the maximum lot size did not exceed 10 lots ($1000 margin).

In a live environment, where large orders cannot be filled instantly, the bot's performance would likely decline. Still, under ideal execution conditions, the EA shows clear profitability and robustness during tests under volatile market hours.

Extra

7am - 9pm (14 hrs)

Balance:	132 510.99	Free Margin:	100 823.39
Credit Facility:	0.00	Margin:	32 295.78
Floating P/L:	608.18	Margin Level:	412.19%
Equity:	133 119.17		



Results

Total Net Profit:	131 510.99	Gross Profit:	453 361.11	Gross Loss:	-321 850.12
Profit Factor:	1.41	Expected Payoff:	78.84		
Recovery Factor:	13.20	Sharpe Ratio:	0.11		
Balance Drawdown:					
Balance Drawdown Absolute:	0.00	Balance Drawdown Maximal:	9 962.69 (14.84%)	Balance Drawdown Relative:	52.99% (2 568.07)
Total Trades:	1668	Short Trades (won %):	834 (64.39%)	Long Trades (won %):	834 (63.55%)
		Profit Trades (% of total):	1067 (63.97%)	Loss Trades (% of total):	601 (36.03%)
		Largest profit trade:	8 787.38	Largest loss trade:	-4 098.00
		Average profit trade:	424.89	Average loss trade:	-535.52
		Maximum consecutive wins ($):	13 (5 522.62)	Maximum consecutive losses ($):	6 (-4 494.10)
		Maximal consecutive profit (count):	24 300.22 (10)	Maximal consecutive loss (count):	-5 800.24 (3)
		Average consecutive wins:	3	Average consecutive losses:	2

Fact Checking

Until it is proven, some questions arise.

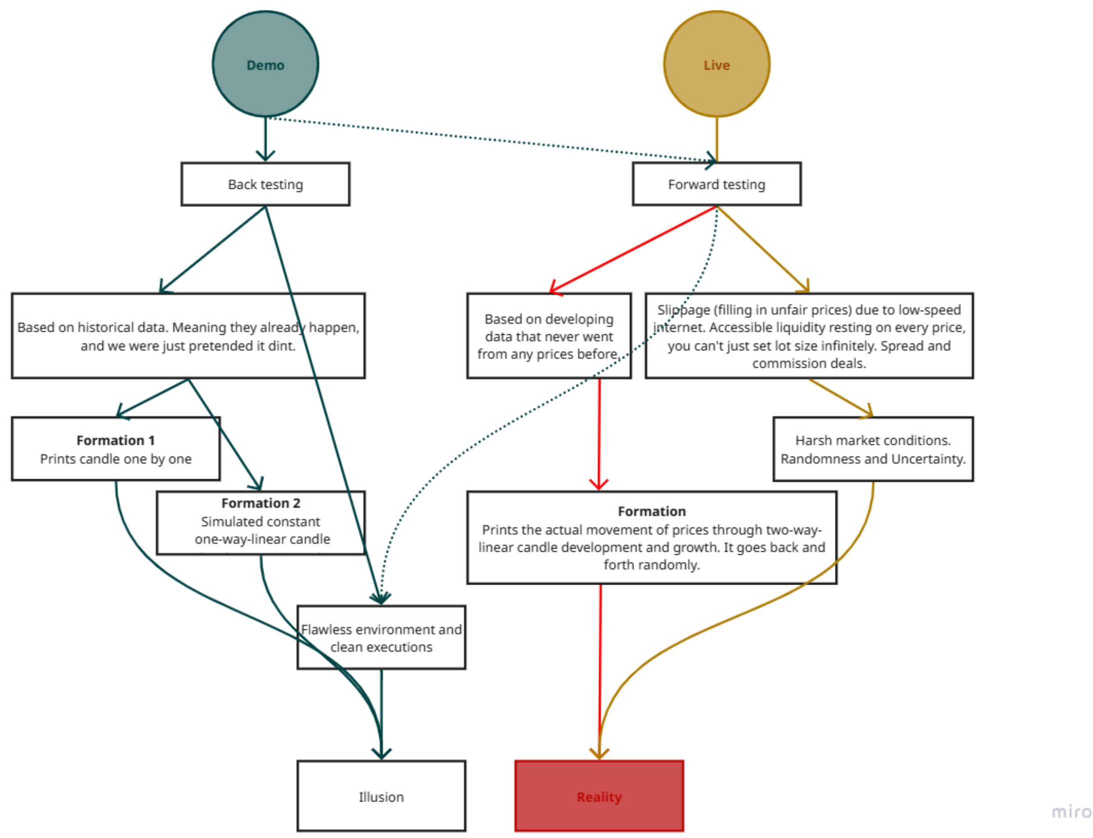
1. Is EA trading bot profitable?
2. Exness offers a cent account, does position sizing in dollar accounts similar to cent account?
3. Can we withdraw the profits from using an EA trading bot?
4. Is it possible to implement EA trading bot on a Live account?
5. Is using EA trading bot illegal? *(could be stated by the law)*

"Yes, you can use an EA trading bot on a live account with Exness. Expert Advisors (EAs) are supported on MT4 and MT5 desktop trading terminals. Ensure your account has sufficient funds, as EAs may be disabled without them. Additionally, you can withdraw profits made using an EA trading bot, as the withdrawal rules are the same as for regular trading accounts. Make sure to use the same payment method for withdrawals as you used for deposits.

Regarding the lot size for a Cent account, it differs from a Standard account. In a Cent account, 1 cent lot equals 0.01 standard lot. This makes Cent accounts ideal for beginners or those testing strategies with lower risk. For example, a balance of 5 USD in a Cent account will appear as 500 USC (500 cents).

Using EA trading bots is not illegal. Exness allows the use of third-party Expert Advisors (EAs) or robots for automated trading. However, different conditions and features may apply depending on your account type, platform, financial product, client jurisdiction, and other factors. If you have any specific concerns or need further clarification, feel free to ask!" ***exness***

Difference between Demo and Live Environment



Live Testing

The mechanics of a live market can be partially simulated in a demo environment by adjusting certain parameters, but several critical variables cannot be accurately replicated. For example, you can set a fixed slippage value of 10 pips during testing, but in real market conditions, slippage is random and can even exceed 100 pips. This exposes you to losses far greater than what you intended to risk.

Access to true resting liquidity at different price levels is another factor that cannot be recreated in a demo setting. Because of this, demo results often appear cleaner, more favorable, and unrealistic, making them unreliable for drawing solid conclusions.

For these reasons, live forward testing remains the most accurate method for evaluating a strategy. It exposes the harsh realities of real market mechanics—execution delays, fluctuating spreads, variable slippage, and liquidity limitations.

In demo testing, order-filling behavior tends to be overly idealized. In contrast, live trading requires dealing with real-world frictions such as slippage caused by spreads, latency, or slow internet connections. Execution speed becomes critical, especially in algorithmic trading, where you are effectively competing with institutional algorithms. Even slight delays can result in missed fills or unfavorable prices.

Having a fast and stable internet connection provides a significant competitive advantage. It ensures your orders reach the market faster, increasing the likelihood of being filled at your intended price. In high-frequency or algorithmic trading, every millisecond matters. A faster connection reduces latency, minimizes slippage, and allows you to react more efficiently to liquidity and market changes. In a market dominated by speed, a superior internet connection can directly translate into better execution and potentially higher profitability.

Code for Live Trading

Tweaked parameters and Added features

```
//+------------------------------------------------------------------+
//|              #1EATB.mq5                                          |
//|   Auto straddle + hedge on TP/SL, loops indefinitely             |
//|   Pure MT5 functions, no MT4-style functions                     |
//|   Trailing stop added & cancel solo pending orders               |
//|   Position sizing: risk 1% of account per order                  |
//+------------------------------------------------------------------+
#property strict
#include <Trade/Trade.mqh>

input double InpRiskPercent = 1.0; // Risk percent per trade
input int    InpPipsEntry = 50;
input int    InpPipsSL    = 100;
input int    InpPipsTP    = 1000;
input int    InpSlippage  = 10;
input ulong  InpMagic     = 7777;
input int    InpTrailTriggerPips = 50;  // Profit pips to start trailing
input int    InpTrailDistancePips = 40; // Trailing distance

CTrade trade;

ulong buyStopTicket  = 0;
ulong sellStopTicket = 0;

// === ADDED: track which position tickets we've already adjusted (to run only once) ===
ulong adjustedTickets[]; // dynamic array of position tickets already handled
// ==========================================================================================

//+------------------------------------------------------------------+
double Pip()
{
   double point = SymbolInfoDouble(_Symbol, SYMBOL_POINT);
   int digits = (int)SymbolInfoInteger(_Symbol, SYMBOL_DIGITS);
   return (digits == 3 || digits == 5) ? point * 10.0 : point;
}
//+------------------------------------------------------------------+
double CalculateLotSize(double stopLossPips)
{
   double equity = AccountInfoDouble(ACCOUNT_EQUITY);
   double pipValue = SymbolInfoDouble(_Symbol, SYMBOL_TRADE_TICK_VALUE);
   double lotStep = SymbolInfoDouble(_Symbol, SYMBOL_VOLUME_STEP);
   double minLot = SymbolInfoDouble(_Symbol, SYMBOL_VOLUME_MIN);
   double maxLot = SymbolInfoDouble(_Symbol, SYMBOL_VOLUME_MAX);

   // StopLoss in money
   double slMoney = stopLossPips * pipValue;

   double riskMoney = equity * InpRiskPercent / 100.0;

   double lots = riskMoney / slMoney;

   // Normalize lot to step & min/max
   lots = MathFloor(lots / lotStep) * lotStep;
   if(lots < minLot) lots = minLot;
   if(lots > maxLot) lots = maxLot;

   return(lots);
}
//+------------------------------------------------------------------+
bool PlacePending(ENUM_ORDER_TYPE type, double price, double sl, double tp, ulong &ticket, string comment)
{
   double stopLossPips = (type == ORDER_TYPE_BUY_STOP || type == ORDER_TYPE_BUY_LIMIT) ? (price - sl)/Pip()
                                                                                         : (sl - price)/Pip();
   double lots = CalculateLotSize(stopLossPips);

   MqlTradeRequest req;
   MqlTradeResult  res;
   ZeroMemory(req);
   ZeroMemory(res);

   req.action      = TRADE_ACTION_PENDING;
   req.symbol      = _Symbol;
   req.type        = type;
   req.volume      = lots;
   req.price       = price;
   req.sl          = sl;
   req.tp          = tp;
   req.magic       = InpMagic;
   req.deviation   = InpSlippage;
   req.type_time   = ORDER_TIME_GTC;
   req.type_filling= ORDER_FILLING_FOK;
   req.comment     = comment;

   if(!OrderSend(req, res))
   {
      Print("OrderSend failed: ", res.retcode, " ", res.comment);
      ticket = 0;
      return false;
   }

   ticket = res.order;
   return true;
}
//+------------------------------------------------------------------+
void PlaceStraddle()
{
   double pip  = Pip();
   double ask  = SymbolInfoDouble(_Symbol, SYMBOL_ASK);
   double bid  = SymbolInfoDouble(_Symbol, SYMBOL_BID);
   int digits  = (int)SymbolInfoInteger(_Symbol, SYMBOL_DIGITS);

    double buyE  = NormalizeDouble(ask + InpPipsEntry * pip, digits);
    double buySL = NormalizeDouble(buyE - InpPipsSL * pip, digits);
    double buyTP = NormalizeDouble(buyE + InpPipsTP * pip, digits);

    double sellE  = NormalizeDouble(bid - InpPipsEntry * pip, digits);
    double sellSL = NormalizeDouble(sellE + InpPipsSL * pip, digits);
    double sellTP = NormalizeDouble(sellE - InpPipsTP * pip, digits);

    PlacePending(ORDER_TYPE_BUY_STOP, buyE, buySL, buyTP, buyStopTicket, "BUY_STOP_EA");
    PlacePending(ORDER_TYPE_SELL_STOP, sellE, sellSL, sellTP, sellStopTicket, "SELL_STOP_EA");
}
//+------------------------------------------------------------------+
bool PendingOrderExists(ulong ticket)
{
   if(ticket == 0) return false;

   if(OrderSelect(ticket))
      return true;

   return false;
}
//+------------------------------------------------------------------+
bool PositionExists(string type)
{
   for(int i = 0; i < PositionsTotal(); i++)
   {
      ulong ticket = PositionGetTicket(i);
      if(ticket == 0) continue;
      if(!PositionSelectByTicket(ticket)) continue;

      if(PositionGetInteger(POSITION_MAGIC) != (long)InpMagic) continue;
      if(PositionGetString(POSITION_SYMBOL) != _Symbol) continue;

      ENUM_POSITION_TYPE ptype = (ENUM_POSITION_TYPE)PositionGetInteger(POSITION_TYPE);
      if(type == "BUY"  && ptype == POSITION_TYPE_BUY) return true;
      if(type == "SELL" && ptype == POSITION_TYPE_SELL) return true;
   }
   return false;
}
//+------------------------------------------------------------------+
void CancelOrder(ulong &ticket)
{
   if(ticket == 0) return;
```

```
    MqlTradeRequest req;
    MqlTradeResult  res;

    ZeroMemory(req);
    ZeroMemory(res);

    req.action = TRADE_ACTION_REMOVE;
    req.order  = ticket;
    req.symbol = _Symbol;

    if(OrderSend(req, res))
    {
       Print("Cancelled pending order: ", ticket);
       ticket = 0;
    }
    else
                Print("Failed to cancel order ", ticket, " ret=",
res.retcode);
}
//+------------------------------------------------------------------+
void CheckAndRebuild()
{
    bool buyPos  = PositionExists("BUY");
    bool sellPos = PositionExists("SELL");

    bool buyActive  = PendingOrderExists(buyStopTicket);
    bool sellActive = PendingOrderExists(sellStopTicket);

    // One triggered → cancel opposite pending
    if(buyPos && sellActive)
       CancelOrder(sellStopTicket);
    if(sellPos && buyActive)
       CancelOrder(buyStopTicket);

    // Loop: if both positions closed → rebuild straddle
    if(!buyPos && !sellPos && !buyActive && !sellActive)
    {
       Print("Both positions closed → placing new straddle");
       PlaceStraddle();
    }

    // Check solo pending orders and cancel them
    if(buyActive && !sellActive)
       CancelOrder(buyStopTicket);
    if(sellActive && !buyActive)
       CancelOrder(sellStopTicket);
}
//+------------------------------------------------------------------+
void TrailStops()
{
    double pip = Pip();

    for(int i=0; i<PositionsTotal(); i++)
    {
       ulong ticket = PositionGetTicket(i);
       if(ticket == 0) continue;
       if(!PositionSelectByTicket(ticket)) continue;

                     if(PositionGetInteger(POSITION_MAGIC)   !=
(long)InpMagic) continue;
       if(PositionGetString(POSITION_SYMBOL) != _Symbol)
continue;

                                     double     open                    =
PositionGetDouble(POSITION_PRICE_OPEN);
       double current = (PositionGetInteger(POSITION_TYPE)
==  POSITION_TYPE_BUY)  ?  SymbolInfoDouble(_Symbol,
SYMBOL_BID)
                                                                                    :
SymbolInfoDouble(_Symbol, SYMBOL_ASK);
       double sl     = PositionGetDouble(POSITION_SL);
                         ENUM_POSITION_TYPE     type     =
(ENUM_POSITION_TYPE)PositionGetInteger(POSITION_T
YPE);

       double profitPips = (type == POSITION_TYPE_BUY)  ?
(current - open)/pip
                                           : (open - current)/pip;

       if(profitPips >= InpTrailTriggerPips)
       {
          double newSL = (type == POSITION_TYPE_BUY)  ?
current - InpTrailDistancePips*pip
                                                               : current +
InpTrailDistancePips*pip;

          // Only move SL forward (do not move backward)
          if((type == POSITION_TYPE_BUY && newSL > sl) ||
(type == POSITION_TYPE_SELL && newSL < sl))
          {
                           trade.PositionModify(ticket,  newSL,
PositionGetDouble(POSITION_TP));
                Print("Trailing SL modified for ticket ", ticket, "
newSL=", newSL);
          }
       }
    }
}
//+------------------------------------------------------------------+
// === ADDED FUNCTIONS: helpers to mark/check adjusted
tickets ===
bool IsTicketAdjusted(ulong ticket)
{
    if(ticket == 0) return true; // treat zero as 'adjusted' to avoid
processing
    for(int i = 0; i < ArraySize(adjustedTickets); i++)
       if(adjustedTickets[i] == ticket) return true;
    return false;
}

void MarkTicketAdjusted(ulong ticket)
{
    if(ticket == 0) return;
    if(IsTicketAdjusted(ticket)) return;
    int newSize = ArraySize(adjustedTickets) + 1;
    ArrayResize(adjustedTickets, newSize);
    adjustedTickets[newSize-1] = ticket;
}
//
====================================================
===================

// === ADDED FUNCTION: adjust SL once after order filled
===
// For each newly-detected position (matching EA magic &
symbol), if we haven't already
// handled it, check its current SL pips. If current SL pips !=
InpPipsSL and current != 100,
// then move SL to be exactly InpPipsSL from open. Execute
only once per position.
void AdjustPositionSLOnce()
{
    double pip = Pip();
            int    digits    =    (int)SymbolInfoInteger(_Symbol,
SYMBOL_DIGITS);

    for(int i=0; i<PositionsTotal(); i++)
    {
       ulong ticket = PositionGetTicket(i);
       if(ticket == 0) continue;

       // If already handled, skip
       if(IsTicketAdjusted(ticket)) continue;

                     if(!PositionSelectByTicket(ticket))    {
MarkTicketAdjusted(ticket); continue; } // defensive

          // Only consider positions opened by this EA on this
symbol
              if(PositionGetInteger(POSITION_MAGIC)   !=
(long)InpMagic) { MarkTicketAdjusted(ticket); continue; }
       if(PositionGetString(POSITION_SYMBOL) != _Symbol)
{ MarkTicketAdjusted(ticket); continue; }

                        ENUM_POSITION_TYPE     ptype     =
(ENUM_POSITION_TYPE)PositionGetInteger(POSITION_T
YPE);
                                        double      open      =
PositionGetDouble(POSITION_PRICE_OPEN);
       double sl   = PositionGetDouble(POSITION_SL);
       double tp   = PositionGetDouble(POSITION_TP);

       double currentSLPips = 0.0;
       bool hasSL = (sl > 0.0);

       if(hasSL)
       {
          if(ptype == POSITION_TYPE_BUY)
             currentSLPips = (open - sl) / pip;  // positive if SL is
below open
          else
             currentSLPips = (sl - open) / pip;  // positive if SL is
above open

          // Round to integer pips (to compare with input which is
integer)
          currentSLPips = MathAbs(currentSLPips);
          currentSLPips = MathRound(currentSLPips);
       }
       else
       {
          // No SL currently set → treat as different (0)
          currentSLPips = 0.0;
       }
```

```
            // If current SL pips == 100 → do NOT execute the
function on this pos (but mark handled)
        if(MathAbs(currentSLPips - 100.0) < 0.5)
        {
            Print("Position ", ticket, " has SL ~100 pips; skipping
adjustment as requested.");
           MarkTicketAdjusted(ticket);
           continue;
        }

          // If current SL pips equals input → nothing to do (but
mark handled)
        if(MathAbs(currentSLPips - (double)InpPipsSL) < 0.5)
        {
           // already at desired SL pip distance
           MarkTicketAdjusted(ticket);
           continue;
        }

        // Otherwise, set SL to be InpPipsSL away from open
        double newSL = 0.0;
        if(ptype == POSITION_TYPE_BUY)
              newSL = NormalizeDouble(open - InpPipsSL * pip,
digits);
        else
              newSL = NormalizeDouble(open + InpPipsSL * pip,
digits);

        // Attempt modify
        bool mod = trade.PositionModify(ticket, newSL, tp);
        if(mod)
           Print("Adjusted SL once for ticket ", ticket, " from ",
sl, " to ", newSL, " (target pips=", InpPipsSL, ")");
        else
           Print("Failed to adjust SL for ticket ", ticket, " ret=",
GetLastError());

        // Mark handled in any case (execute only once)
        MarkTicketAdjusted(ticket);
   }
}
// === END ADDED FUNCTION ===

//+------------------------------------------------------------------+
int OnInit()
{
   PlaceStraddle();
   return(INIT_SUCCEEDED);
}
//+------------------------------------------------------------------+
void OnTick()
{
    // First: check for newly filled positions and adjust SL (only
once per filled order)
   AdjustPositionSLOnce();

   CheckAndRebuild();
   TrailStops();
}
//+------------------------------------------------------------------+
```

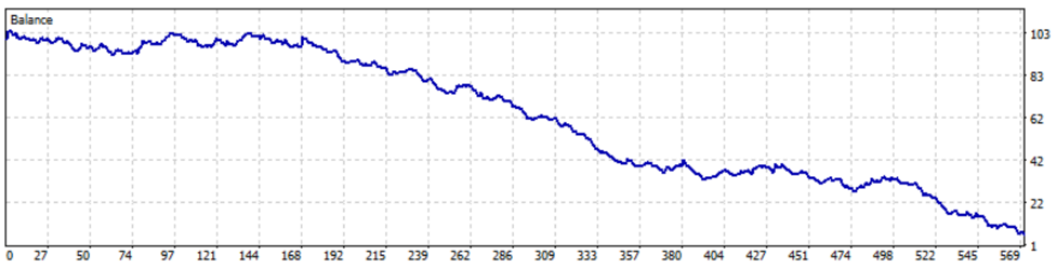
Live Trading Results

Raw results-LiveTest1

Trade History Report

Name:	**LiveTest1**
Account:	**257199749 (USC, Exness-MT5Real36, real, Hedge)**
Company:	**Exness Technologies Ltd**
Date:	**2025.11.24 14:11**

Equity: **6.10**



Results

Total Net Profit:	**-93.90**	Gross Profit:	**197.50**	Gross Loss:	**-291.40**
Profit Factor:	**0.68**	Expected Payoff:	**-0.16**		
Recovery Factor:	**-0.96**	Sharpe Ratio:	**-0.14**		
Balance Drawdown:					
Balance Drawdown Absolute:	**93.90**	Balance Drawdown Maximal:	**98.10 (94.15%)**	Balance Drawdown Relative:	**94.15% (98.10)**
Total Trades:	**588**	Short Trades (won %):	**292 (55.48%)**	Long Trades (won %):	**296 (52.36%)**
		Profit Trades (% of total):	**317 (53.91%)**	Loss Trades (% of total):	**271 (46.09%)**
		Largest profit trade:	**4.50**	Largest loss trade:	**-2.20**
		Average profit trade:	**0.62**	Average loss trade:	**-1.08**
		Maximum consecutive wins ($):	**9 (4.20)**	Maximum consecutive losses ($):	**8 (-8.10)**
		Maximal consecutive profit (count):	**6.80 (8)**	Maximal consecutive loss (count):	**-8.10 (8)**
		Average consecutive wins:	**2**	Average consecutive losses:	**2**

After 7 hours 04 minutes 12 seconds, with a trade of 588—this LiveTest1 result shows a system with a decent win-rate of 53.91% but a heavily negative reward-to-risk ratio. Losses are significantly larger than profits, leading to a 94% drawdown and almost total equity loss.

Issue

1. I didn’t expect that the spread on the cent live account would be 30 pips. The demo test was done on a dollar account (since a demo cent account is not available on Exness), which had a 20 pip spread.
2. Upon the test running, constant different combinations of parameters have been applied and slight tweaking i could think of but none of that gave significant changes on the equity drawdown.

Proven on live

1. The position-sizing logic works correctly. 0.01 lot in the demo equaled $1, while 0.01 lot in the live cent account equaled 1 cent.
2. The threshold logic also works: 1% of 100 cents is 1 cent, which corresponds to 0.01 lot. The EA continued executing 0.01-lot trades even when the account balance dropped below 100 cents, confirming consistent behavior.

Tweaked parameter

1. **Issue:** Severe slippage resulted in losing more than intended.

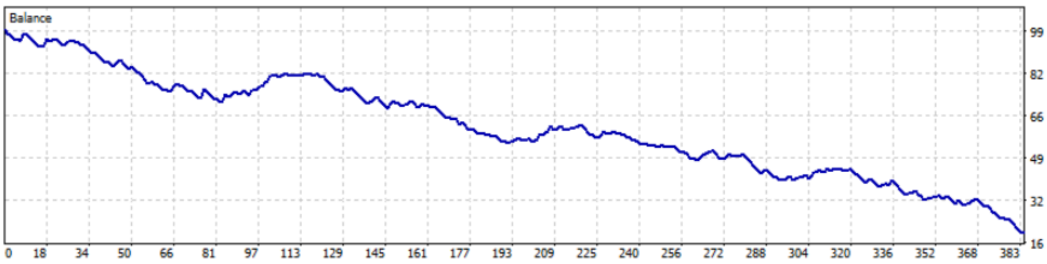
   Add a feature. If the executed SL (in pips) does not match the EA’s initial input, automatically adjust the SL price to match the intended pip value. If the SL is already at 100 pips, do not execute the adjustment. This ensures the EA will not exceed the intended maximum SL of 100 pips, preventing excessive losses caused by slippage.

LiveTest2

Trade History Report

Name: **LiveTest2**
Account: **263204408 (USC, Exness-MT5Real37, real, Hedge)**
Company: **Exness Technologies Ltd**
Date: **2025.11.24 19:42**

Equity: **20.10**



Results

Total Net Profit:	**-79.80**	Gross Profit:	**113.70**	Gross Loss:	**-193.50**
Profit Factor:	**0.59**	Expected Payoff:	**-0.20**		
Recovery Factor:	**-0.99**	Sharpe Ratio:	**-0.22**		
Balance Drawdown:					
Balance Drawdown Absolute:	**80.40**	Balance Drawdown Maximal:	**80.40 (80.40%)**	Balance Drawdown Relative:	**80.40% (80.40)**
Total Trades:	**391**	Short Trades (won %):	**182 (50.00%)**	Long Trades (won %):	**209 (49.28%)**
		Profit Trades (% of total):	**194 (49.62%)**	Loss Trades (% of total):	**197 (50.38%)**
		Largest profit trade:	**2.90**	Largest loss trade:	**-1.70**
		Average profit trade:	**0.59**	Average loss trade:	**-0.98**
		Maximum consecutive wins ($):	**11 (8.00)**	Maximum consecutive losses ($):	**7 (-7.80)**
		Maximal consecutive profit (count):	**8.00 (11)**	Maximal consecutive loss (count):	**-7.80 (7)**
		Average consecutive wins:	**2**	Average consecutive losses:	**2**

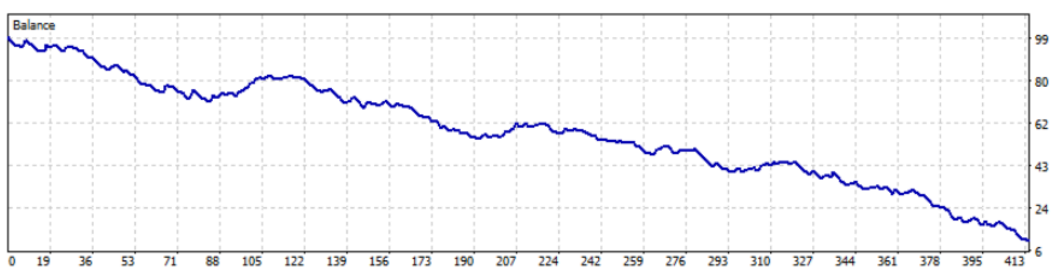
After 5 hours, 21 minutes, 43 seconds, with a trades of 391 —this LiveTest2 result is no different from the previous test, and shows a system with a decent win-rate of 49.62% but a heavily negative reward-to-risk ratio. Losses are significantly larger than profits, leading to a 80.40% drawdown and almost total equity loss.

Live Trading Final Conclusion

Both LiveTest1 and LiveTest2 clearly demonstrate that despite having acceptable win-rates (53.91% and 49.62%), the strategy remains deeply unprofitable due to a heavily negative reward-to-risk profile—large losses consistently outweigh small profits, leading to extreme drawdowns of 94% and 80.40%. The root cause behind this imbalance is not the strategy logic itself but the latency and slippage inherent in live execution. Orders are filled far worse than expected, distorting the intended stop-loss and take-profit distances, and ultimately destroying the reward-to-risk ratio. In a live environment where execution delay and widening spreads are unavoidable, the strategy becomes structurally disadvantaged, making profitability unattainable until execution speed and slippage control are addressed.

Extra (5 hours, 57 minutes, 41 seconds)

Equity: 10.60



Results

Total Net Profit:	-89.40	Gross Profit:	121.90	Gross Loss:	-211.30
Profit Factor:	0.58	Expected Payoff:	-0.21		
Recovery Factor:	-1.00	Sharpe Ratio:	-0.22		
Balance Drawdown:					
Balance Drawdown Absolute:	89.70	Balance Drawdown Maximal:	89.70 (89.70%)	Balance Drawdown Relative:	89.70% (89.70)
Total Trades:	421	Short Trades (won %):	196 (50.51%)	Long Trades (won %):	225 (48.89%)
		Profit Trades (% of total):	209 (49.64%)	Loss Trades (% of total):	212 (50.36%)
		Largest profit trade:	2.90	Largest loss trade:	-2.00
		Average profit trade:	0.58	Average loss trade:	-1.00
		Maximum consecutive wins ($):	11 (8.00)	Maximum consecutive losses ($):	7 (-7.80)
		Maximal consecutive profit (count):	8.00 (11)	Maximal consecutive loss (count):	-7.80 (7)
		Average consecutive wins:	2	Average consecutive losses:	2

Conclusion

The comparative evaluation of demo and live trading results reveals a fundamental divergence between theoretical strategy performance and real-world execution feasibility. In the demo environment, the Expert Advisor (EA) demonstrated strong apparent profitability within a 12-hour testing window while operating at a 3% risk per trade. High win rates, rapid order execution, and favorable liquidity assumptions enabled significant returns, despite elevated drawdowns and a low Sharpe ratio. These characteristics indicate that the strategy is highly dependent on volatility conditions and benefits materially from the unrealistically efficient execution inherent to demo environments.

In contrast, live trading results expose the structural weaknesses of the strategy. Although win rates in LiveTest1 and LiveTest2 remained moderate (53.91% and 49.62%), the strategy proved consistently unprofitable. Losses systematically outweighed gains, resulting in extreme drawdowns of 94% and 80.40%. This deterioration is not attributable to flaws in the strategy's conceptual logic, but rather to execution-related factors intrinsic to live markets. Latency, slippage, and spread expansion materially distorted the intended stop-loss and take-profit structure, collapsing the reward-to-risk profile that supported demo-level profitability.

Collectively, these findings demonstrate that the strategy's apparent edge is execution-dependent rather than structurally robust. While the EA can exhibit profitability under idealized execution conditions, it becomes fundamentally disadvantaged in live environments where execution delays and liquidity constraints are unavoidable. As a result, sustained profitability is unattainable without substantial improvements in execution speed, slippage control, and liquidity-aware trade management.