# Backtesting and machine learning applied to a self-developed futures trading strategy

**Anant Sahai, Odysseas Drosis**
**Inspirit Ai, Los Altos California**

---

## Summary

Having traded futures discretionary for about a year as most in this position I noticed that written rules would not reproduce in results. The usual explanation is that a trader's memory is fallible. I wanted to investigate another: the incompleteness of the written rules since the real most important rules are subconscious, applied without being noticed. I have tried to encode my strategy for Nasdaq-100 and S&P 500 futures exactly as I have written it, matched it against ten years of one-minute prices from 7.07 million bars. When traded it lost money: compared to break even probability 33.3%, 31.7% of trades were profitable with expectancy of -0.175R per trade over 205 trades. I have tried to switch parameters and none of the 24 of them resulted in profit. I looked at the only section of specification written without any measurable definition and what it depicted was the market structure or session-structure model: Accumulation-Manipulation-Distribution, that states whether to trade a day or not. Having added it without altering all other inputs resulted in trading 53.0 % of trades profitably with +0.480R of expectancy on 742 trades, positive on all five years of testing. In a separate analysis I have had a look at 300 101 samples through machine learning methods where it resulted in unpredictable short term price direction and predictable movement (expansion of volatility) with 57.6 % of hits against the base probability of 48.2%. The one rule I have never bothered to write down mattered most.

---

## INTRODUCTION

Futures on stock-market indices trade in a continuous auction, where the price only advances if a buyer and seller are willing to exchange. The major E-mini futures (E-mini S&P 500, E-mini Nasdaq-100) run around 23 hours a-day via the Globex trading platform (Chicago Mercantile Exchange), however the bulk of the volume lies in the US cash session (9:30 a.m. and 4 p.m. ET) (1). Because every trade needs a counter-party, a large order cannot be fully completed at one single price. It needs a pool of opposite interest deep enough to absorb it. Larger players need therefore to trade at prices where plenty of other bids/offers are piled up - and it is that mechanical constraint which determines where/how the price moves over a session.

As traders' behaviours are predictable, those concentrations of orders are predictable too. A trader who detects an upward trend and buys a breakout above previous high would also put in a stop-loss just below previous low. Because many people believe in the same logic, it creates thick clusters of sell orders below the visually obvious lows and buy orders above them too. Osler has recorded such occurrences in the currency markets directly; she has noticed that actual stop loss amounts cluster around rounded values and previous extremes and that price moves quicken when the previous clusters are hit too. It comes counter-intuitive. When a prior high is breached, it may not mean just that the price will continue further upwards: this may well be just a way for large sellers to get fills they wish in establishing a short, a way of manipulating market behaviour to their advantage (2). The interpretation of movement by the retail observer is often exactly inverse to the role the same event plays to a big money.

It is this reversal on which trading lines of attack, gathered loosely under the ICT (Inner Circle Trader) banner, are constructed. The approaches are to enter positions against the apparent signal, on the notion that a visible breakout will have been fabricated to furnish a larger participant with the required fills. We will not venture to arbitrate whether such ideas reflect the true inner workings of microstructure; neither will we conclude that they merely provide a story with strong enough adhesive properties such that patterns may be perceived to the user's satisfaction. Both are compatible with a working strategy.

We are interested in solving a simpler, more testable problem. Discretionary traders often claim to perform better than what their written rules could reproduce if those rules were followed strictly. The explanations usually offered for this gap are: recall bias, when winning trades are remembered more readily than losers; selective journaling, not reporting losing days; and outright self-deception. In all cases the error lies in reporting, in that either recall was biased towards positive outcome trades, some unfavourable days were not recorded in the journal, or outright self-deception. The written specification is taken to be complete, the reported performance inflated.

We hypothesized the opposite - that the specification is not complete and the incompleteness is of a non-random, systematic nature. Discretionary traders do keep track of rule-based entities because entry (whether buying, selling, or otherwise) is always a discrete and time-stamped activity fitting naturally within the event-based structure of a trading journal and not all too far off from the more precise specification for their automated brethren. Preconditions, the assessment a trader does of whether a specific day should be attempted at all, even before reaching out for rules of engagement when considering trading at all, are monitored continuously and largely subconsciously - since they do not generate a discrete action, such activities would rarely find space for entry. If some considerable share of the strategy is in fact derived from what has been omitted, then faithful encoding of rules that were actually written should lead to underperformance on the part of the automated trader, which must then be recoverable by detecting what was omitted instead of correcting what was previously put down.

To test this, we coded the discretionary strategy as written word for word exactly, measured performance on 10 years worth of one minute futures data and tried to find the source of disparity. We came to find that the strategy did not make money even before transaction cost. No parameter value that allowed this to be recovered. One pre-condition that had been written accounted for the difference of death and birth. The discretionary trader recorded and knew of this pre-condition in notes and other forms but did not consider it important since they were not

explicitly required to. He added in good faith a second filter that he used to apply - this again was taking winners off the strategy.

---

## RESULTS

### Performance of the strategy as written

First we transcribed and implemented the exact specification of the system. The system instructs the trader to draw the high/low of the first 2 five-minute bars after the open, find a session liquidity point, wait for 1 of several confirmation signals and enter with a stoploss & and a take-profit that is 2x the risked distance away. A section of the specification, the “Accumulation and Distribution” section, provided a chart with no definable points and it was simply ignored, with a note of it being a gap.

The strategy was tested against 2016-2021 at the 2:1 reward-to-risk ratio. The strategy placed 205 trades, of which 31.7 were profitable. The meaning of 31.7 becomes clear once the break even win percentage for 2:1 risk to reward ratio which is 33.3 is revealed (Table 1). At 31.7%, the strategy was not viable.
Dissecting gross from net expectancy is revealing. Trading expectancy is defined in terms of R-units (a single R-unit being equal to the amount risked on that particular trade) which allows the comparison of trades of various sizes. Its average gross expectancy (the expectancy excluding commission and slippage) per trade stood at -0.118R. Add to that the further expectancy of -0.057R incurred as transaction costs and you end up with its disappointing value of -0.175R (Table 1), on average, per trade. It is clear that the problem here stems from flaws within the entry signal per se and not with the efficiency of its trading execution. Commission payments apart, it was already unprofitable. Consequently even lower commissions or improved fills could not have salvaged it. In performance improvement, this eliminates one entire class of option as the problem, as described below, lay not with execution efficiency (lower commissions, better spreads etc) itself.
No discernible pattern appeared in the annual results. The percentage of wins varied randomly between 15.2% (2017) and 43.9% (2018), producing two with positive expectancy out of six (Table 2). A system with a positive indeed an inconsistently applied expectancy would be expected to produce years primarily above the break-even and different greatly, than years scattered above and below no discernable pattern. The returns from this trading system were not only disappointing overall, they were also maddeningly random from year to year.

### Parameter optimization does not recover performance

Concluding that the edge was absent, we started testing whether instead the poor result arose from wrong parameters being specified. It is plausible that a rule could be stated qualitatively in a written specification, e.g. entry window “9:50-10:30ish” and the wrong numeric value could easily be used when programming this while a strategy that works at one setting fails at another (even when the signal logic itself is held constant). We tested a matrix of 24 configurations including 4 entry window widths, 3 confluence requirements and 3 trade direction filters.

Negative expectancy was shown by all 24 configurations. The best producing run resulted in −0.012R but on only 16 trades, a dataset number too small to recognise it from random occurrences. The best producing runs on an acceptable number of trades still failed to make any progress with −0.022R after 96 trades.

The more interesting aspect than the magnitude of the results was their distribution on the grid. If a parameter truly causes an effect, the search space will be ordered: the performance moves up (or down) consistently as you move along one or more axes, enabling one to locate the optimum. According to results, there was no such ordering: be it the width of the entry window or the trade direction, it did not consistently damage or enhance the performance, meaning it was just a random fluctuation and parameter search should be stopped.
Win rate depends on target distance, not signal quality
Investigate win rate, the number quoted by the practitioner to describe the strategy's performance. Win percentage is intuitive and commonly quoted, but not directly linked to profitability, and we conducted a controlled experiment to show the dependency clearly. Keeping the entry signals absolutely fixed, trading the exact same entries, at the exact same prices, with the same stops, we started playing with just the distance of our profit target. Win rate moved from 21.7% with a 2-to-1 ratio, to 64.9% with a 0.5 ratio (Figure 1, Table 3). All cases continued to lose money.

The effect is arithmetic not empirical. The target is hit more often because the price does not need to move as far before the trade can be closed for profit, however each time it is hit, proportionally less profit is made. As a result, the break-even win rate equal to 1/(1 + reward-to-risk ratio) rises along with the win rate. At the 0.5:1 ratio the strategy had the win rate of 64.9% against break-even of 66.7% - the highest win rate that was tested. However, the loss rate was also highest at -0.132R per trade ( Table 3 ).
Hence, a reported win-rate has no information as to whether the strategy made money without information on the respective target-stop/win-risk. This is the convention followed when describing the performance of future strategies; win rates are coupled with 'expectancy' and 'profit factor', which are free from manipulation by target distance alone.

Wider stop-losses reduce transaction costs but do not restore the edge
It also implies paying a commission for each contract: it costs in total a commission that does not change according to how close/far the stop loss is set from my point of entry. Therefore the component of my capital (my risk) covered by the commission decreases as the length of exit order increases ($5 commission with a risk of only $250 equals a much higher percentage of it than a commission of equal amount paid when I risk ten times as much). The transaction costs amounted to −0.057R and we tested if the widening of the stop could manage to offset the amount thereof.
For the next stage of trading 205 baseline trades were divided into stop-bands for which the gross expectancy and cost drag and therefore net expectancy were calculated (Table 4). The cost drag behaved exactly as expected: it decreased continuously as far as a minimal 0.012R in contrast to the maximum 0.085R in case of the narrowest band. The process seems to be working.The gross expectancy, however, did not behave as we would hoped. The 40-60 tick

band had gross expectancy of +0.293R which appears to support our predictions, but its immediate neighbour 15-25 tick band had an expectancy of −0.756R, based on sample size of 23 and 31 trades respectively (Table 4). Opposite signs on samples of this size in adjacent bands do not imply a relationship between stop width and signal quality, but random noise; selecting the useful band as stoprule means fitting a rule to numbers of observation less than 25, which was not our task then.

**An unwritten precondition accounts for the performance deficit**

Once more I hit a brick wall about parameter changes and came back full stop with the one and only section of the spec, which we did not manage to cope with - this was the Accumulation and Distribution section, in which price clumping up and forming “a ladder for itself” and untapped areas would draw future price etc was the most of the description. This description indicated a visual pattern without disclosing a threshold, a count or anything measurable we might convert into code.

We asked the practitioner directly what the section described. His answer identified a specific and named framework: the Accumulation-Manipulation-Distribution model, in which accumulation is the consolidation of price within the opening range, manipulation is a false breakout from that range which subsequently reverses and distribution is the directional move which follows in the opposite direction. Two relatively short follow-up questions established the last two operational details which we needed; accumulation is measured on the opening range rather than the overnight session and the manipulation price does not have to specifically sweep off a named level.The key thing about this scheme is that it is a precondition, not an additional confirmation signal. It answers “whether it is on at all today.” A day where the market never leaves the opening range is one where accumulation has no need; a session where the market leaves the range and never comes back is trending upward, not manipulating. Neither is an AMD day, and both are therefore filtered out entirely. Furthermore, it defines a direction to trade rather than giving the choice; a sweep of the opening range high dictates that distribution will be downwards, not upwards.

After implementing AMD as a filter, while keeping everything else constant - entry window, date range, instruments, confirmation requirements, risk parameters - both performance and the filter’s impact could be evaluated directly, free of any confounding factors. Win rate improved from 35.9% to 50.0% in 2016 and from 25.5% to 42.9% in 2017, pushing expectancy from negative to positive in both years.

Comparing 2017 results under two models is more illustrative of the AMD filter’s effect. From the raw analysis run for a narrow entry window, 2017 was the worst year in terms of win rates (15.2%) and expectations (−0.703R) (Table 2); the filter was able to bring the portfolio to almost break-even. Even in its (baseline) worst stage a winning strategy now seems profitable than the previous scenario of having more losing streaks than profitable ones. A filter that improved only already-favorable periods would suggest it was selecting for conditions that happened to suit the strategy. If AMD filters a year by having more win streaks over losses (2017 under raw strategy had more losses), the real filter must actually work to the advantage of the system by filtering

losing trades and improving expectation, for example, from losses of 0.7R to a profit of almost 0R.

## A believed constraint was excluding valid trades

Having added the AMD filter, a gap of another order of magnitude had existed in the trade rates of the engine & practitioner; the former made 0.85 per week whereas the latter cited three to four. This gap indicated that once again a glitch existed since such differences occurred not due to chance alone.

The difference was traced to one instruction in the written specification. "A genuine breakout tests a price level twice before breaking through it. You enter on the second test." The engine took it literally and required two separate excursions beyond OCLO to make one breakout trade. With AMD models this was impossible. According to the rationale that the manipulation stage IS the signal - that it leaves accumulation, takes out stop losses that sit beyond the range and closes back inside - the first sweep is consistent with the model. Waiting for the second cannot be justified from the point of view of the method. Hence we introduced a single sweep trigger running in parallel to the double test trigger and ran 3 configurations - double test only, single sweep only and both permitted.

Allowing either trigger increased the trade count from 20 to 74 over the second half of 2016 - 3.7x and improved the win rate from 50.0% to 52.7%, expectancy from +0.274R to +0.427R and the profit factor from 1.32 to 1.63 (**Table 6**). Notice this is still accepting double-test signals; we just stopped requiring waiting for one.

Here the usual relationship between frequency and quality - (relaxing the filter on trades and thus allowing less good ones would usually increase frequency at expense of reduced performance) was essentially reversed. Relaxing the filter on trades improved everything and the reason was obvious; the doubled filter requirement did not filter out bad trades - it excluded good ones.

## Full-period performance of the corrected strategy

I compared the results of the corrected setup; AMD filter, both triggers now ON (previously NONE of them was ON), enter period 9:50-12:00, RR=2:1. Year chosen 2017 to 2021 (5 years to have many trades on an annual basis to compare across time but also have fresh data for 2022+).
Number of trades: 742 trades, 393 won, 349 lost (53.0% win rate against 33.3% break-even rate). Expectancy +0.480R per trade. Profit factor 1.714 (**Table 7**). The trade frequency was approximately 2.85 per week which closely matched the requested 3-4 trades per week by the practitioner.
All years had positive expectancy and no sign reversals (**Figure 3**, **Table 8**). Instead of a percentage return we show expectancy and profit factor: the reason for this is that we model position sizing compounding off a growing balance, thus positions later into the back test are

several times larger than those earlier into the back test. The percentage return depends as much upon the position sizing model as the strategy. Expectancy and profit factor on the other hand are independent of account size and thus comparable through the periods.
Win rate declined steadily across the five years, from 55.7% in 2017 to 49.4% in 2021, with expectancy following the same pattern from +0.498R to +0.396R (**Table 8**). At 115 to 173 trades per year, this pattern may reflect sampling variation, but it may also indicate genuine decline in the strategy's effectiveness over time. We address this ambiguity in the Discussion.

## Mechanical trailing stops reduce performance

One practice mentioned by the respondent was not given in the above written specification. Namely, should the price move 70 to 80% to its target place the existing stop-loss on the price of opening or simply take half of its maximum gain when fairly confident? This is to reduce risk; let us test if such actions reduce the risk.
The results of testing four variants: break-even at 70%, break-even at 80%, half-lock at 70% and half-lock at 80% and their comparison with no action are shown in table 9. All 4 variations reduced the trade gains, the best variation is with trailing stop-loss, getting 0.061R against 0.274R for not trailing at all.
This effect is apparent in the number of trades. The trades stopped out remained constant at 10 across all the settings whereas the trades hitting target dropped (from 10 to just 2 on one occasion) (**Table 9**). This means the trailing stop was protecting only winning trades. Losing trades do not track 70% of the way to reach a target - they fall away from an early stage. The trail could thus only affect winning trades - changing some of these to breakeven scraps as they fell away to trail before returning to target.
This result contradicts the rule being a mechanical threshold. When and if to move at all is determined at the discretion of the trader using real time price behaviour to inform his decision, this cannot be factored into the test. This version of the moving average, in effect, may be better than the results derived from the experiment above would indicate.

## Short-term price direction is not predictable from bar structure

The trading results above raised a separate question: how much of the strategy's behavior could be recovered by a model learning directly from price data, without any of the strategy's rules? We addressed this with a supervised learning analysis, using it both as an independent check and as a way to quantify concepts the written specification had left undefined.
Five models are fit to predict the forward return over the next 30 minutes from 17 features describing recent bar structure. The models are trained on 240,081 samples and tested on 60,020 samples chosen chronologically instead of at random (refer to Materials and Methods section). Five of the 17 features were designed in part to define concepts from the specification that were never explicitly defined - as they described how bars overlap and form clumps, reflecting the clumpiness that the accumulation section talks about.
Every one of the five models produced negative
$R^2$

values and thus each performed worse than simply predicting the training-set mean for every observation (Table 10). The directional accuracy (the fraction of the predictions that have the correct sign) ranged from 47.4% to 52.3% and hence was consistent with chance.
All five models produced negative
$R^2$
values, meaning each performed worse than simply predicting the training-set mean for every observation (Table 10). Directional accuracy, the proportion of predictions with the correct sign, ranged from 47.4% to 52.3%, consistent with chance.
I report this outcome then change my approach until I can produce a positive value. Why? A negative does not really matter and is to be expected. Trustworthy short horizon return prediction through nothing but price history forms something of an edge for market participations to exploit. Would it not be surprising if this continued to be possible?

## Volatility expansion is predictable from the same features

We tested a related idea after concluding there was no direct forward return prediction. Maybe similar features can predict a different property of future price behaviour? Let us treat the task of anticipating future price excursions like a binary classification. Will an excursion beyond a certain threshold relative to ATR be visible during the next thirty minutes of trading (ATR-average true range - a common technical measure of typical bar to bar price move)?
The threshold needed setting for the task to be sensible. The obvious choice when measuring anything with unknown spread, is some multiple of the average true range. If we set this at 1.0 times average true range, our average excursion would generate an average classification rate 86.4% of times labeled as expansion. Thus a model which was simply taught to predict expansion would deliver 86.4%, having learned nothing. Model comparison would therefore deliver little meaning at this threshold. We picked the middle way: set the threshold as the empirical median excursion, which is measured across our whole sample to be 1.62 times average true range. This produced a balanced 50.2%/49.8% split respectively, which is used for all classification work herein.
All five models for this split with the 17 chosen features, outdid the majority class baseline of 48.2% with a gain of 8-9 percentage points, for the neural network this was 57.6%. This is visible in figure 4 and table 11.
The contrast with the result from regression should not be surprising, it is not news but theoretical. Prices are approximately unpredictable but volatility is not: large movements in price cluster together closely in time and so do periods of relative calm. Mandelbrot was the first to illustrate this pattern for commodity prices (3) and Engle formalized it in the autoregressive conditional heteroskedasticity models which depend on recent history for a measure of volatility (4). Our models discovered such clustering, inherent to the data irrespective of any (un)forecastability of returns themselves.
Averaging the probabilities of the five models does not improve the result of the best model (it obtains 57.36% compared to the best single model's result of 57.63%). Ensembling averages out errors that are not shared by the models, the result is the mean average probability, not really the best of any. Not all models have statistically uncorrelated errors and therefore averaging them will not improve the results. This results from 5 models using exactly the same

data and nearly the same techniques to model it. The weights in this average were all about the same, in magnitude from 0.198 to 0.202, since their respective accuracies were about a few tens of percent from one another.

### Model complexity produces measurable overfitting

We decided we would sweep tree maximum depth which took on 8 values for both tree based models measuring the accuracy on training and hold-out testing data for each (Figure 5 and table 13). The maximum depth means how many decisions a tree can make, so in theory how accurately the models were able to divide the data.
With the use of a single decision tree, the pattern characteristic of overfitting began to emerge. The training accuracy rose from 0.587 at the deepest tree (deep=2) to 0.674 (deepest tree, no restriction on depth of the tree), while the test accuracy rose to 0.5642 at depth 8, then decreased to 0.5470. The difference between the training accuracy and the test accuracy increased from 0.034 to 0.127. The gap between training and test accuracy widened from 0.034 to 0.127 (Figure 5), while for testing it was steepest at depth 8 (Table 13).
Because it averaged predictions over 100 independently constructed trees, the random forest could tolerate deeper trees and degraded more gracefully. It reached its optimum test accuracy of 0.5715 at depth 12 and was still producing near optimal results with accuracy at 0.5710 when trees were grown without pruning rules. Table 13 illustrates the variance reduction that allowed the ensemble of trees in the random forest to outperform the stand-alone trees; errors between the stand alone trees do tend to cancel given they possess only partial correlation (5).

### Time of day dominates feature importance

Considering a range of different features in random feature selection helps determine which of the seventeen features most influences the random forest (as illustrated in Figure 6 and Table 14). Importance scores are calculated via impurity during training. The minutes elapsed since the market open contribute 43.6% of total importance to decision makers, which is roughly five times that of the following feature. Most of the rest of the importance is associated with volatility and volume type measures which support the hypothesis of clustering as the operating mechanism.
Given no information about at what times the strategy considered trades viable, it was able to recover the relevance of session timings itself. A check on whether particular time windows do have different behaviour could thus partially be externally provided. This was 43.6% of the relevance.
Among the last important features are the five features built up to capture the accumulation concept of how tight consecutive bars cluster and overlap (see **Table 14**). As the effect of the AMD filter determines the level of success, it might seem weird at first. We will give the explanations of this situation in the Discussion.

---

# DISCUSSION

A discretionary futures scalping strategy was turned into executable code and tested on 10 years of one-minute prices.Executed exactly as specified by the strategy writer, it lost money: 31.7% of trades were profitable against a 33.3% break-even threshold: the expectant was negative even before adding commissions to losses and removing them from profits: **Table 1**. Experiment with 24 permutations of parameter values for the strategy and unable to discover a profitable one. There was no ordered relationship on the parameter-grid and hence future trials of a similar kind would only constitute yet more random probing in the blind. I identified one precondition, not part of the original written strategy and added it to the model (AMD = Accumulation - Manipulation - Distribution Session filter). The strategy win-rate goes to + 53.0 % on 742 trades, the expectation improves to + 0.480 R and the Profit/Loss-factor shoots to 1.714; profitable in each and every of the years tested, including the last two: **Tables 7, 8.** A further correction - removed one constraint that practitioners believed to have implemented, resulted in trade 3, 7 times more frequent and win rate, expectation and Profit/Loss-factor actually improved more! (**Table 6**)

These findings are explained by the possibility that the practitioner's strategy was profitable because it depended on a filter that removed trades even before he started to apply his written down entries rules. A plausible reason for why this would not be written into his trading journal is that discretionary traders write down entry rules because entries are distinct events which have a time stamp and a trading journal is structured around events that occurred. Preconditions, on the other hand, produce no event and are removed all the time, creating no record. However, preconditions could simultaneously hold almost all the profit in a strategy, because a filter that disqualifies up to 90% of potentially good days can have a far greater average impact than any refinements applied to days that stay. This explanation is supported by the fact that the pattern we discovered to be important to the strategy, the AMD pattern, was actually identified by the practitioner before the project was even started. A losing trade was analysed on record with the following statement, "accumulation and distribution pattern was not there," following a remark drawn on a chart of "all of this does not exist without this pattern." No one, let alone the practitioner, had in fact recognised these statements as anything other than comments and notes up until a direct inquiry. We iterate the fact that what has been put forward and defended as explanatory theory here arose studying a single practitioner using a single trading strategy and in order for any claims to be made on the subject of discretion itself, experiments of the kind should be replicated on multiple practitioners using multiple strategies.

Our conclusions are subject to several limitations. The main one is that we have not re-created any trade that has actually been taken by the practitioner and so the encoded rules simply may not be consistent with how the strategy was traded. The practitioner claims a win rate of approximately 65% as per our observation 53.0 and three explanations are indistinguishable: either there are more unencoded preconditions to trade or some real-time judgement not amenable to rules is involved or the recalled figure, instead of tallied from records, is biased upward. Second, the rule to choose between the two instruments according to 'cleaner price action' has been replicated with a proxy counting violations of the opening range, which is inverted: an erratic instrument contained in its range would score clean whereas a smoothly trending instrument that broke out early would score poorly. The measured expectancy of the strategy may be underestimated due to incorrect instrument selection on a significant fraction of signals. Third, our simulation of ambiguous bars in which either the stop or target could have

been hit during a single minute of trading is determined in favor of the former; such choice is necessarily conservative but cannot be verified due to lack of tick-level data which was not feasible. Fourth, annual sample sizes of trades (115-173) are small, causing year to year figures to incorporate wide uncertainties and therefore the result pooled over the entirety of five years are the more reliable estimate.

My next highest priority experiment is direct validation against trades actually made by the practitioner. This could be achieved by running dated sessions through the engine and comparing with the outcome trade by trade. This would, each time resulting in one of three results, all of which would prove to be instructive. If the engine reproduces a trade, the encoded rules are correct for this case and thus the remaining discrepancy must lie somewhere else. If the engine misses a trade that has been taken, such a gate is too restrictive and can be identified. If the engine takes a trade that was declined, extra unconscious filters are in effect and can thus be identified. The AMD pre-condition was discovered by asking what a written phrase meant. Thus my guess is the third case could give way to more (if not much) discoveries. By adhering to a semi-structured interview procedure consisting of questions such as the following, should the engine take trades that would be declined by the trader, a lucky accident can easily be turned into a reproducible technique: Ask the trader what reasons have brought him not to take the declined trade. Using similar procedures upon trading sessions can then answer a further question, i.e., whether such unrecorded preconditions appear across-the-board at discretionary traders or if this represents a singularity.
Present results bring three follow-up, narrower questions. Firstly, the gradual decrease in win rate annually from 55.7% to 49.4% (**Table 8**) could be attributed to variation in sampling or the strategy decay. It is useful to distinguish the true cause by considering the performance in the 2022-2026 period in which no model-tuning has occurred at all. The 2022-2026 is truly an out-sample of any strategy-designing effort, let alone hyper-parameters tuning. Secondly, the rule for instrument selection needs a better definition than the current proxy in terms of instrument. One should get the practitioner to label a sampling sequence of sessions and get help from Machine Learning algorithms like the random forest and SVM to identify the implicit rule that identifies the cleaner. For that to happen, the sample of data would naturally need to increase. Lastly, Machine Learning results have to explain one tension: whilst AMD filtering had decisive effects on the Trading Engine, the features that captured accumulation concept (in other words, the features designed as proxies to capture signals from the accumulation phase) were the variables close to the bottom of features importance (**Table 14**). Here, we suspect the lack of importance was due to construction of poor features reflecting the concept adequately as a twelve-bar window cannot be representative of a phase that lasts for more than just a few candle bars like a session, even with few windows overlapping. A model that uses full sequence or alternatively has features to identify the transition from accumulation to manipulation and again to distribution would be a proper test of how important the concepts are. More generally, if unrecorded conditions weigh heavily on the balance of forces in discretionary strategies in general, than just backtesting implication it would have implications when teachers focus on entry conditions that practitioners often do as an element of the study: this could mean teaching about the least important concepts of a method. Furthermore, when attempts at automating discretionary trading or building AI for discretionary traders, would firstly elicit the condition on

which a trader decides not to enter: a question nobody ever asks because journals prefer to record on their activities and not what they missed.

---

# MATERIALS AND METHODS

## Strategy specification

The strategy proposed in this short analysis is an Intraday Reversal Trading Course for the NQ and ES futures, elaborated by the first author through daily watching and journalling of markets for about an year, before being converted in the written specification followed in this study. The specification in its written form is a procedure outlined in five steps: Check Order Flow of Daily and 15 minute charts; Mark H&H or L&L of 9.30 and 9.35am 5min bars of both futures; Identify session liquidity target; confirm entry with one or several of four continuation signals; Apply risk management using a reward/risk ratio 2/1 to 3/1. Two main sources of ideas have converged in its design: a study of material inspired by the ICT tradition (namely six course-material videos summing to about 18 hours by an educator calling himself TJR; as Tylar [TJRTrades] as found on internet) a structure based on Crude Openings/Open range borrowed from a previous orientation of work towards Zarattini/Aziz (6) and Crabel (7). A section of the written specification, called Accumulation and Distribution describes the concept but does not operationally define it while visually presenting it.

## Data acquisition

I collected historical price data from Databento through the dataset GLBX.MDP3 of schema ohlcv-1m. Date range was 2016-06-21--2026-07-20, both for the NQ and ES parent symbols at the cost of $19.8. I used parent-symbol resolution which will return every instrument under the root symbol. The result was 4,952,381 rows from NQ that had 40 outright contracts and 452,885 calendar spreads. There were 5,612,293 rows from ES that contained 681,335 spreads.

## Continuous contract construction

Since several contract expirations traded simultaneously and printed bars at the same times, a single, continuous series first had to be established before any analysis could take place. Spread instruments that had a hyphen in the symbol were excluded. Trading volume for every outright contract was then summed on each day and the contract with the highest volume became the chosen front month for that day. It was important not to roll back to an earlier expiration since otherwise a low-volume session could quickly revert the series to a contract that had already been taken over. Volume-based rolling was preferred over calendar rolling as it corresponds to how the active contract is determined by a discretionary trader: you follow liquidity instead of a fixed schedule. The procedure resulted in 3,530,028 NQ bars and 3,541,035 ES bars with 40 contract rolls each.

Each bar whose time stamp was 6:00 p.m. or after Eastern time was treated for purposes of assigning it to a trading session as being part of the following calendar day, in conformity with the CME schedule which states trading is from 6:00 p.m. Sunday to 5:00 p.m. Friday. Without

doing this, Monday evening's Asian session would be counted as occurring on Monday, pushing every previous-day high and low that the Strategy uses ahead by one full session.

## Data quality verification

The two generated bars series were checked against no-nos such as high smaller or equal to low bars (not encountered), open/close value outside of bar´s own high/low range (not encountered again), duplicated timestamps (not encountered) or flat bar where high equals low (~1.84% is fine since there does not happen a significant action during the night). Asian, London and New York session hours presence is explicitly checked. This is necessary because strategy is dependent on over-night levels and a file inadvertently truncated so as to represent only regular trading hours would offer results believable at first instance, yet utterly senseless.

## Rule implementation

The Engine consists of seven Python modules totaling 7,653 lines and verified by 89 unit tests, of which some examples are in Appendix A.

The team adopted a two-bar definition of swing points, where a swing high is an up bar immediately followed by a down bar priced at the higher of the two; a swing low is its mirror image. Since one use of swing points is as a measure by which to enter, the software automatically chose the more conservative of the high and low of a swing point than the lowest low/highest high. Doji bars where the open equals the close are skipped when pairing. A break of the market structure required a bar to close beyond the most recent confirmed swing than just its wick exceeding the level (it does this because one of the uses of these levels is for collecting liquidity and a wick is the actual example of this that we wanted to collect for ourselves using the stop-hunt) and was therefore used as the high/low that would break the swing of the bar.

The range was produced from the high and low wicks of the 9.30 and 9.35 a.m. five minute bars on each instrument. A touch was recorded when a wick met a range but the bar closed back inside it, a breach when the bar closed beyond it. Episodes of breached bars next to each other were grouped and their greatest excursion from the wick boundary stored.

A phase structure which represented the accumulation, manipulation and distribution (AMD) cycle was assigned by the AMD classifier to each session. The first of the three phases of accumulation was the range in which the price was opening. The manipulation phase was defined as the first occasion at which price closed outside this specified range and it lasted till price closed back within the opening range. The phase during which the price trended in the opposing direction was termed 3rd as distribution. Some sessions in which the price failed to leave the range or left it and did not return to the ranges, were of class non-AMD and hence ignored for trading. Then electing the price direction, it was defined relative to the side being swept.

Levels of liquidity were the Asian (6 p.m. to 3 a.m.) and London (3 a.m. to 8:30 a.m.) session and the close of the previous day. A level is tapped when it is traded through after first being formed. Wicks counted now, we are trying to figure out where there are resting orders than how the structure is broken. We excluded the New York extremes as they are still forming now and that would be information unavailable at time of decision.

Confirmation signals were a combination of three patterns. A fair value gap, a 3-bar pattern in which the wicks of the 1st and 3rd bars do not overlap so leaving behind a region of price through which price traded absent of opposing interest - tracked and recorded in the states formed, tapped and inverted; where inversion is required price had closed across to the other side. A SMT divergence, pairing swing points between the two instruments within maximum one bar tolerance and identifying leading behaviour. Break of structure on minute time-frame was the third used. The appearance of either one would qualify.
A stop-loss was placed beyond the second most recent one-minute swing in the direction of the trade - the second was used as the most recent swing is usually the one level price has bounced off and is too close to suffer regular fluctuation. Targets were specified as the above multiple of stop-loss and limited to the nearest untapped level (e.g. nearest swing to the stop-loss with no wick intrusion (the definition of unused for those familiar)). When traded using this strategy, the number of contracts was selected such that a loss would reduce equity by the same amount each time.

## Prevention of look-ahead bias

The simulation used the following restrictions to ensure it never reacted using knowledge not available at the time of the decision: A swing point is defined by two bars and it cannot be known until the second bar has been fully established. This confirmation timestamp where available in the engine which disallowed the break-of-structure detection firing before this point on the swing being broken; Secondly range of unusable prices was implemented relative to the close of 9:40am swing so that the price bars at the boundaries of the swing had no affect; Thirdly the liquidity-taps and the gap-states was added with a cutoff time - which allowed the simulation to query the state of the market as of the given bar, without reference to subsequently formed bars - each of which has a unit test that will fail if the guarantees are broken. (show more - 2).

## Trade simulation

Stops were filled at stop price plus one tick. Target fills occurred at target price (no slippage allowed as limit orders are at target or better). If a stop and target coincided within a signal bar, the outcome was a stop out for simulation. The reason why two orders filled inside the same bar is unknown without tick level data, and resolving it favorably has been noted as a way of increasing your result during backtesting. Commission was $2.50 / contract/side.
We mention one implementation error here for reproducibility. The profit target was originally based on the close, while entry was 1 tick worse resulting in a mean reward to risk ratio of 1.69 to 1. Correcting this *lowered* reported win-rate from 38.3 to 21.7 percent, which was artificially inflated by profits more achievable than those which meet specification. All result reporting in this manuscript uses the correct method.

## Machine learning dataset construction

There were too few (~few hundred trades), to train five model classes with hyper-parameter search at the trade level and not over-fit noise instead of signal, hence I used bar (candle) level data ~300,101 samples gathered from 2598 sessions(days).

Finally, seventeen features were calculated on five minute bars within a preceding twelve-bar (one hour) window. These bars captured information that could infer: - Five about compression: trailing ratio of mean range of bar relative to total window span; ratio of body span relative to range of bar; trailing amount of overlap of the next bar with the current bar; trailing standard deviation of closing prices; how much bar ranges changed. - Five related to direction and smoothness: trailing net move relative to total movement; trailing net move relative to 5-minutes average true range; trailing ratio of wick of bar relative to range of bar; trailing proportion of bars moving upwards; trailing bar to bar autocorrelation of returns. - Three about volatility and momentum, - Two about volume traits. - Two describe current state: amount of elapsed minutes since market started; span between close market and mid-point of first ranges of bars.
Forward return relative to average true range for the next six bars (1 bar duration = 5 minutes and each day has 78 5-min bars) and a binary signal on whether maximum price excursion of the next six bars exceeds some multiple of average true range comprise two target variables, where the normalization ensures that the 2016 and 2026 price excursion levels or direction are comparable, unlike raw point movements which are not directly comparable across different price levels. Features used information until bar size t, whereas targets use information from bar t+1 to t+6; targets and features do not share common parts in data; last 6 bars are neglected from original data since their targets will span till next training session start.

## Model training and validation

For continuous target, which is ridge regression; logistic regression for binary target. K-nearest neighbours of k = 50 and distance weighting. Decision tree, random forest of 100 trees and multiplayer perceptron with 64x32 hidden layers and early stopper. Five classes were accessed with scikit-learn (8). Models that are scale sensitive, which are linear models, k-nearest neighbours and neural networks were all implemented in a pipeline so standardising is only fitted using a trainingw set. If split is done after standardising, information of the test set could be incorporated into the training.
The data set had to be split chronologically since subsequent samples within the data set were close to being duplicates resulting from the overlapping feature windows and using a random selection would therefore leak information relating to the training samples and subsequent evaluation of models, to the test set. Split 80/20 chronologically led to training samples ranging from early 2016 to mid 2024 (240081 samples) and test set samples ranging from mid 2024 to mid 2026 (60020 samples). The training / test split was implemented so that the models being trained could identify and learn features up until 2024, but then could be tested against recent data, but not data available to them at the time.
Results of all runs were tested against a baseline where the training-set mean (regression) or the majority class (classification) was predicted. With financial data a model can output plausible results even when it has not learned anything and comparison against baseline is what differentiates subtle and valid signals apart from noise.

## Statistical treatment

Expectancy is written as R units meaning values can become comparable across differences in position size and accounts. Net Profit / Gross loss equals the profit factor. The breakeven win

rate is (1/(1+Reward/Risk)). If the number of trades falls into the region of less than 30 or so trades the results are tagged as provisional and no conclusions are reached from them.

# REFERENCES


1. CME Group. *E-mini S&P 500 futures contract specifications*. Chicago: CME Group, 2023.
2. Osler, C. L. "Currency orders and exchange rate dynamics: An explanation for the predictive success of technical analysis." *The Journal of Finance* 58, no. 5 (2003): 1791–1820.
3. Mandelbrot, B. "The variation of certain speculative prices." *The Journal of Business* 36, no. 4 (1963): 394–419.
4. Engle, R. F. "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation." *Econometrica* 50, no. 4 (1982): 987–1007.
5. Breiman, L. "Bagging predictors." *Machine Learning* 24, no. 2 (1996): 123–140.
6. Zarattini, C., and A. Aziz. "Beat the market: An effective intraday momentum strategy for S&P 500 ETF (SPY)." *SSRN Electronic Journal* (2023). SSRN 4416622.
7. Crabel, T. *Day trading with short term price patterns and opening range breakout*. Greenville, SC: Traders Press, 1990.
8. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research* 12 (2011): 2825–2830.


---

# FIGURES AND TABLES

**Figure 1. Win rate rises as profit targets move closer, while profitability falls.** Line plot showing observed win rate (solid line) and the win rate required to break even (dashed line) across three reward-to-risk ratios: 0.5:1, 1:1, and 2:1. Entry signals were identical across all three conditions; only the distance to the profit target was varied. The observed win rate falls as the ratio increases, but the break-even requirement falls faster, so the gap between them is smallest at the widest target. Data in Table 3.

**Figure 2. The AMD filter improves win rate in both years tested.** Grouped bar chart with two clusters (2016 and 2017), each containing two bars representing the filter disabled and enabled. The y-axis shows win rate; a dashed horizontal line marks the 33.3% break-even threshold for a 2:1 reward-to-risk ratio. All parameters other than the filter were held constant between conditions. Data in Table 5.

**Figure 3. Expectancy remains positive in every year tested.** Bar chart with year on the x-axis (2017 through 2021) and expectancy in R units on the y-axis, with a horizontal reference

line at zero. Trade count for each year is annotated above the corresponding bar. Data in Table 8.

**Figure 4. All five models exceed the baseline at classifying volatility expansion.** Horizontal bar chart with model class on the y-axis and test-set accuracy on the x-axis. A vertical dashed line marks the 48.2% majority-class baseline. Evaluated on 60,020 held-out samples. Data in Table 11.

**Figure 5. Training and test accuracy diverge as tree depth increases.** Line plot with maximum tree depth on the x-axis (2, 3, 5, 8, 12, 16, 20, and unconstrained) and accuracy on the y-axis. Four series show training and test accuracy for the decision tree and for the random forest. The decision tree's test accuracy peaks at depth 8 and declines thereafter while its training accuracy continues to rise; the random forest's test accuracy remains near peak at all depths. Data in Table 13.

**Figure 6. Time of day dominates feature importance.** Horizontal bar chart showing the ten highest-importance features from the random forest classifier, sorted in descending order. Importance values are impurity-based scores produced during model training. Data in Table 14.

---

**Table 1. Baseline performance of the strategy as written, 2016–2021.**

| Metric | Value |
|---|---|
| Trades | 205 |
| Win rate | 31.7% |
| Break-even win rate at 2:1 | 33.3% |
| Gross expectancy | −0.118R |
| Cost drag | 0.057R |
| Net expectancy | −0.175R |
| Profit factor | 0.75 |

**Table 2. Annual baseline performance.**

| Year | Trades | Win rate | Net expectancy |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 2016 | 13 | 38.5% | −0.029R |
| 2017 | 33 | 15.2% | −0.703R |
| 2018 | 41 | 43.9% | +0.214R |
| 2019 | 49 | 30.6% | −0.227R |
| 2020 | 38 | 26.3% | −0.309R |
| 2021 | 31 | 38.7% | +0.059R |

**Table 3. Win rate and profitability across reward-to-risk ratios.** Entry signals were identical in all three conditions; only target distance varied. Data from 2016–2017.

| Reward:risk | Win rate | Break-even required | Net expectancy | Profit factor |
|---|---|---|---|---|
| 0.5 : 1 | 64.9% | 66.7% | −0.132R | 0.63 |
| 1.0 : 1 | 54.8% | 50.0% | −0.027R | 0.88 |
| 2.0 : 1 | 21.7% | 33.3% | −0.513R | 0.44 |

**Table 4. Performance by stop-loss width.** Baseline trades, 2016–2021.

| Stop width (ticks) | Trades | Win rate | Gross expectancy | Cost drag | Net expectancy |
|---|---|---|---|---|---|
| 0–15 | 107 | 38.3% | +0.040R | 0.085R | −0.045R |
| 15–25 | 31 | 9.7% | −0.756R | 0.039R | −0.795R |
| 25–40 | 25 | 24.0% | −0.304R | 0.027R | −0.331R |
| 40–60 | 23 | 43.5% | +0.293R | 0.021R | +0.272R |
| 60+ | 19 | 26.3% | −0.220R | 0.012R | −0.232R |

**Table 5. Effect of the AMD filter with all other parameters held constant.**

| Year | Configuration | Trades | Win rate | Net expectancy |
|---|---|---|---|---|
| 2016 | AMD disabled | 39 | 35.9% | −0.120R |

| | | | | |
|---|---|---|---|---|
| 2016 | AMD enabled | 20 | 50.0% | +0.274R |
| 2017 | AMD disabled | 98 | 25.5% | −0.422R |
| 2017 | AMD enabled | 35 | 42.9% | +0.072R |

**Table 6. Effect of permitting single-sweep entries alongside double-test entries.** Second half of 2016.

| Configuration | Trades | Win rate | Expectancy | Profit factor | Trades/week |
|---|---|---|---|---|---|
| Double test only | 20 | 50.0% | +0.274R | 1.32 | 0.77 |
| Both triggers | 74 | 52.7% | +0.427R | 1.63 | 2.93 |

**Table 7. Full-period performance of the corrected strategy, 2017–2021.**

| Metric | Value |
|---|---|
| Trades | 742 |
| Winning trades | 393 |
| Losing trades | 349 |
| Win rate | 53.0% |
| Break-even win rate at 2:1 | 33.3% |
| Expectancy | +0.480R |
| Profit factor | 1.714 |
| Trades per week | 2.85 |

**Table 8. Annual performance of the corrected strategy.**

| Year | Trades | Win rate | Expectancy |
|---|---|---|---|
| 2017 | 115 | 55.7% | +0.498R |
| 2018 | 153 | 55.6% | +0.551R |

| | | | |
|---|---|---|---|
| 2019 | 141 | 53.2% | +0.486R |
| 2020 | 173 | 52.0% | +0.479R |
| 2021 | 160 | 49.4% | +0.396R |

**Table 9. Effect of mechanical trailing stops.** 2016, 20 setups.

| Variant | Win rate | Expectancy | Scratches | Targets reached | Stop-outs |
|---|---|---|---|---|---|
| No trail | 50.0% | +0.274R | 0 | 10 | 10 |
| Break-even at 70% | 35.0% | −0.043R | 3 | 7 | 10 |
| Break-even at 80% | 40.0% | +0.061R | 2 | 8 | 10 |
| Lock 0.5R at 70% | 50.0% | −0.183R | 8 | 2 | 10 |
| Lock 0.5R at 80% | 50.0% | +0.048R | 4 | 6 | 10 |

**Table 10. Forward return prediction.** Evaluated on 60,020 held-out samples.

| Model | $R^2$ | Directional accuracy |
|---|---|---|
| Baseline (predict mean) | −0.0007 | 47.4% |
| Linear | −0.0009 | 49.1% |
| K-nearest neighbors | −0.0338 | 49.2% |
| Decision tree | −0.0283 | 52.3% |
| Random forest | −0.0091 | 50.2% |
| Neural network | −0.1647 | 48.7% |

**Table 11. Volatility expansion classification.** Evaluated on 60,020 held-out samples.

| Model | Accuracy | F1 score |
|---|---|---|

| | | |
|---|---|---|
| Baseline (majority class) | 48.2% | 0.651 |
| Linear | 57.3% | 0.569 |
| K-nearest neighbors | 56.3% | 0.547 |
| Decision tree | 56.4% | 0.563 |
| Random forest | 57.2% | 0.568 |
| Neural network | 57.6% | 0.557 |

**Table 12. Ensemble versus individual model performance.**

| Method | Accuracy |
|---|---|
| Simple average of five models | 57.36% |
| Performance-weighted average | 57.37% |
| Best individual model | 57.63% |

**Table 13. Maximum tree depth sweep.**

| Model | Max depth | Training accuracy | Test accuracy | Gap |
|---|---|---|---|---|
| Decision tree | 2 | 0.5867 | 0.5526 | 0.034 |
| Decision tree | 5 | 0.5951 | 0.5623 | 0.033 |
| Decision tree | 8 | 0.6039 | 0.5642 | 0.040 |
| Decision tree | 12 | 0.6218 | 0.5577 | 0.064 |
| Decision tree | 20 | 0.6599 | 0.5496 | 0.110 |
| Decision tree | Unconstrained | 0.6735 | 0.5470 | 0.127 |
| Random forest | 2 | 0.5908 | 0.5657 | 0.025 |
| Random forest | 8 | 0.6150 | 0.5682 | 0.047 |

| | | | | |
|---|---|---|---|---|
| Random forest | 12 | 0.6513 | 0.5715 | 0.080 |
| Random forest | 20 | 0.6869 | 0.5709 | 0.116 |
| Random forest | Unconstrained | 0.6915 | 0.5710 | 0.121 |

**Table 14. Feature importance, random forest classifier.**

| Rank | Feature | Importance |
|---|---|---|
| 1 | Minutes since market open | 0.436 |
| 2 | Volume trend | 0.090 |
| 3 | Volume ratio | 0.085 |
| 4 | Average true range change | 0.081 |
| 5 | Net move (ATR-normalized) | 0.049 |
| 6 | True range (normalized) | 0.048 |
| 7 | Distance from opening-range midpoint | 0.047 |
| 8 | Average true range (normalized) | 0.035 |
| 9 | Bar range variability | 0.032 |
| 10 | Wick-to-range ratio | 0.016 |

---

## APPENDIX A — CODE AVAILABILITY AND TEST COVERAGE

All analysis code is available at https://github.com/anantsahai09-lab/trading-decision-support . The repository contains the rules engine, the machine-learning pipeline, a command-line interface, a web application, and the complete test suite.

Test coverage. Eighty-nine unit tests across four modules, with emphasis on boundary conditions and prevention of look-ahead bias: swing detection (17 tests), opening range (19), session liquidity (22), and confirmation signals (31).

Codebase size. Engine modules 4,180 lines; machine-learning pipeline 720; application 1,240; tests 1,513; total 7,653.