I built a crypto signals app. Then I measured every prediction it ever made.

May 30, 2026 • by Murat • ~15 min read

TL;DR — THE 90-SECOND VERSION

I spent two years building a crypto signals app. It runs 12 technical indicators on dozens of coins across multiple timeframes, computes a consensus, and pushes alerts when most of the indicators agree. It looks credible. It feels credible. Users assume the math means it works.

Last week, I ran a complete audit on the 3,767 trades my engine has called since I shipped a critical bug fix in May. The numbers were a gut punch:

- **35.5% win rate.** Out of 687 closed trades, 244 won and 443 lost. The math required ~50% to be profitable at the symmetric 1:1 risk-reward I had configured, after Binance's 0.20% round-trip fee.
- **Higher "confidence" scores correlated with WORSE outcomes**, not better. The 75-80% confidence band won only 27% of the time (n=11). The 65-70% band won 40.2%. The 55-60% band won 41.4%. The confidence number my app prominently displayed to users was actively misleading.
- **One of my indicators, Williams %R, was reverse-predictive.** When it disagreed with the engine's overall consensus direction, the signal won 60.5% of the time across 152 trades. When it agreed, only 43%. The "vote" of that indicator should have been flipped.
- **Per $100 deployed on every signal, you'd have lost approximately $1,470 cumulatively** since the fix went live.

I am publishing this because the alternative was shipping another version of the app, taking customers' money for a Pro tier, and continuing to pretend the engine works the way the UI implied. I'm pivoting the product. Here's the methodology, the data, the academic context, and what I'm doing instead.

How the engine worked

For two years, the product was a 12-indicator technical-analysis consensus engine. The architecture would be familiar to anyone who's looked at the inside of a retail signals app:

INDICATOR FAMILY	SPECIFIC IMPLEMENTATION
Momentum	RSI(14), Williams %R(14), Stochastic(14, 3, 3)
Trend	MACD(12, 26, 9), EMA(50)/EMA(200) cross, ADX(14)
Volatility	Bollinger Bands(20, 2), ATR(14)
Cloud	Ichimoku(9, 26, 52)
Volume	OBV, Volume Profile
Mean reversion	CCI(20)

Every hour, the engine pulled OHLCV candles from Binance for each of ~30 tracked coins across the 5m / 15m / 1h / 4h / 1d / 1w / 1M timeframes. It computed each indicator's reading, voted bullish / bearish / neutral per indicator, summed the votes, and produced a directional consensus. If 7 or more out of 12 indicators agreed AND the confidence score was above 55%, the engine published a signal — entry price, target (the Bollinger upper band for bullish, lower for bearish), stop loss (mirrored 1:1 to the target distance for symmetric risk-reward).

72 hours later, an evaluator looked at the next 72 hours of 1h candles and asked: did the target or stop hit first? Win, loss, mixed (neither hit), or unknown (both hit in the same candle).

Every signal got pushed to subscribed users as an Expo notification. The app displayed signals on a Dashboard, the per-coin history on a Track Record screen, and offered a "Returns" calculator that simulated "if you'd traded every signal at $X per signal, you'd have made $Y."

This is the architecture of dozens of crypto signal apps. The math is nominally sound. The data flow is professional. Users assume — because the interface implies — that the system has predictive value.

I assumed the same thing for two years.

What the audit found

In May 2026 I fixed a long-standing R:R asymmetry bug (it had been silently shipping signals with stops tighter than targets, which biased the win-rate down by ~20 points). After the fix, I had a clean dataset. Here is the unvarnished result on 3,767 evaluated trades:

Overall

METRIC	VALUE
Total evaluated trades	3,767
Wins	1,059
Losses	1,524
Mixed (neither target nor stop hit in 72h)	13
Unknown (both target and stop touched same candle)	1,171
Win rate (wins / wins+losses)	**35.5%**
Average PnL per closed trade (before fees)	-0.187%
Average PnL per closed trade after 0.20% fee	-0.39%
Per $100 deployed on every signal: cumulative dollar PnL	-$1,470

The signals did not make money. Across every timeframe, every direction, every coin. The closest any subset came to breakeven was the 4h timeframe — which the codebase's own internal comments described as "the only timeframe historically clearing breakeven" — and which the post-bugfix data showed was actually the **worst** performer, at 25.3% win rate.

By timeframe

TIMEFRAME	WIN RATE	AVG PNL	CLOSED TRADES
5m	42.3%	-0.061%	1,341
15m	43.1%	-0.074%	538
1h	36.6%	-0.444%	317
4h	25.3%	-1.294%	99
1d	0.0%	-4.149%	8

Every timeframe is net-negative. The 4h reading — the one I told myself for two years was the "smart trader's window" — was the worst.

The confidence-score paradox

This is the finding that broke me.

The app prominently displayed a "confidence percentage" next to every signal. Users — including me — naturally assumed higher confidence meant a higher probability of being right. That's how confidence interfaces work.

CONFIDENCE BAND	WIN RATE	SAMPLE SIZE
55-60%	41.4%	1,772
65-70%	40.2%	800
75-80%	27.3%	11 (small sample, but directional)

> ⚠ TRUST BUG, NOT A MATH BUG
>
> Higher confidence ≠ higher accuracy. If anything, the 75-80% band trended worse. The "confidence" number I had been displaying for two years was not a probability. It was a weighted vote count, scaled to look like a probability. Users were reading it as a probability. I was reading it as a probability. The data said it wasn't one.

Per-indicator audit (the Williams %R finding)

I asked a sharper question. For each of the 12 indicators, when it agreed with the overall consensus direction (a "vote") vs. when it disagreed, what was the win rate of the resulting signal?

INDICATOR	AGREE WR	DISAGREE WR	LIFT
Volume Profile	40.7% (n=2,061)	34.6% (n=52)	+6.1
CCI(20)	50.0% (n=14)	46.2% (n=13)	+3.8
OBV	41.2% (n=2,402)	39.5% (n=172)	+1.6
Stochastic(14,3,3)	42.1% (n=57)	44.1% (n=202)	-2.0
MACD(12,26,9)	41.3% (n=2,235)	45.8% (n=24)	-4.6
Williams %R(14)	**43.0% (n=135)**	**60.5% (n=152)**	**-17.6**

Six indicators (RSI, ATR, ADX, Bollinger, Ichimoku, EMA cross) didn't produce directional disagreements at all — they were context indicators that always aligned with the dominant consensus. They added zero directional information.

Three were noise (CCI, OBV, Stochastic) — net-zero or near-zero lift.

One was anti-predictive: **MACD** — slightly worse outcomes when it agreed with consensus than when it disagreed. The most-trusted name in technical analysis, per our data, was reverse-correlated.

And one — **Williams %R** — was strongly reverse-predictive. When it disagreed with the other 11 indicators (152 cases, statistically meaningful), the resulting signal won 60.5% of the time — well above breakeven. When it agreed, only 43%.

> ★ THE WILLIAMS %R FINDING
>
> Williams %R is a momentum oscillator. It's supposed to identify overbought/oversold conditions. According to our 2,583-trade dataset, it does — but in reverse. When Williams %R says "this is the same as everyone else," the trade fails more often than the baseline. When Williams %R says "I disagree," the trade wins more often than the baseline.
> I don't know with full statistical rigor whether this is a true regularity or a sample-window artifact. The honest answer is "it's a finding worth investigating, not a tradeable edge." But it is genuinely interesting research. And it's something I, the engine builder, did not know until I forced myself to look.

The academic context

I am not the first person to discover that retail-applied technical analysis does not consistently produce profits. I am, perhaps, one of the first to publish my own private data confirming it.

The academic literature is unambiguous on this question, and has been for decades. Here is what the rigorous studies say:

Sullivan, Timmermann & White (1999)

Published in the Journal of Finance. Applied White's Reality Check — a bootstrap method that corrects for data-snooping bias — to the 26 technical trading rules from Brock, Lakonishok & LeBaron (1992) that originally found positive results.

They tested across 100 years of Dow Jones Industrial Average data. The conclusion: the rules looked profitable in-sample, but generated **no superior performance out-of-sample**. The Brock/Lakonishok edge was a data-snooping artifact.

Bajgrowicz & Scaillet (2012)

Published in the Journal of Financial Economics. Revisited 115 years of DJIA data using the False Discovery Rate methodology. Their conclusion:

> "Even in-sample, the performance is completely offset by the introduction of low transaction costs. Persistence tests show that, even with the more powerful FDR technique, an investor would never have been able to select ex ante the future best-performing rules."

In plain English: even when a TA rule looks great in retrospect, you couldn't have picked it in advance. And once you account for transaction costs, even the in-sample winners weren't winners.

Wei (2024) — the crypto verdict

The most recent rigorous test. Wei tested **7,846 technical trading rules + 5 log moving average-based ratios** across 12 cryptocurrencies including Bitcoin. The result: **only 2 rules** survived both in-sample and out-of-sample profitability tests — the short-term log moving average ratio AND the Hashrate Index. Out of nearly 8,000 strategies, 2 had real edges.

In other words: the vast majority of TA rules don't work on crypto either, and even the small number that survive rigorous testing barely beat buy-and-hold.

Retail trader outcomes

The reality at the user level is brutal.

- **Brazilian Securities Commission study (2019)** following every retail day trader in Brazil over 300+ trading days: **97% lost money.** Only ~1% earned more than the Brazilian minimum wage.
- **Taiwanese long-running study (1992-2006)**: **only 1% of day traders** were consistently profitable over multiple years.

These are the academically established baselines. My 35.5% win rate is not a bug. It is the academically expected outcome for any 12-indicator ensemble TA system applied to crypto by a retail user.

Why I'm pivoting (and what I'm building now)

I have three choices when I look at this data honestly:

Choice 1: Hide the data, keep selling the signals. This is what most signal services do. It's also what crypto pump-and-dump groups, Telegram "VIP signal" sellers, and most "AI-powered crypto trading" apps do. They never publish their measured accuracy. They post selective screenshots of winners. They never let users see the full track record. This is morally unacceptable to me. It is also commercially fragile — eventually a user computes the real accuracy and the trust collapses.

Choice 2: Fix the engine, then sell signals. This is the academically honest path, but it's a multi-year research project with a low probability of success. Renaissance Technologies has the resources to find a TA-adjacent edge. I, a solo founder, do not. The Wei 2024 result (2 surviving rules out of 7,846) tells me the search space is enormous and the survival rate is brutal.

Choice 3: Change what the product is. Stop selling predictions. Start selling context. Show users what TA says AND what the crowd thinks AND when those two stories disagree. Don't claim to predict prices. Claim to surface the gap, because the gap is informative even when neither side is right.

I'm choosing 3.

The new product

Starting with v1.1.0, Cryptochartics is positioning around a single tagline:

> ★ THE PIVOT, IN ONE SENTENCE
>
> **See when crypto crowd sentiment and technical indicators disagree.**

Concretely:

- The **Pulse tab** is the new headline. For every tracked coin, we show crowd sentiment (aggregated from CoinGecko community engagement + Wikipedia interest spikes) alongside the 12-indicator TA consensus. When they disagree — when the engine says BULLISH but the crowd is 28% bullish — we flag it as a divergence.
- The **"confidence %"** is gone from signal cards. Replaced with plain language: "8 of 12 indicators bullish." No probability claim. The number is what the data is. Nothing more.
- The **"Returns" screen** is renamed **"Engine Accuracy."** Same calculations, honest framing. The hero number is no longer "you'd have made $X." It's the win rate, with the breakeven line drawn, with the academic context inline.
- **Push notifications** lead with divergence events ("⚠ Divergence · BTC 4h — engine bullish, crowd 28% bullish"), not generic consensus pings.

The engine doesn't go away. It becomes one input among several, with its measured accuracy on display.

What divergence is actually useful for

I'm not claiming divergence is a predictive trading signal. I'm explicitly not claiming that. Here's what I think it actually is:

- **Context for the trade decision the user is already going to make.** If you're already going to buy ETH, knowing that the crowd is 70% bullish (lots of company) vs. 30% bullish (you're betting against everyone) is information worth having.
- **An attention filter.** Most coins on most days have aligned sentiment + TA — nothing to look at. The interesting moments are the ones where they pull apart. Divergence is a "look here" gesture, not a "do this" command.
- **Educational.** Watching divergences resolve over weeks teaches you something about market psychology that neither pure TA nor pure sentiment can teach you alone.

I'd love it if my data showed that contrarian-to-the-crowd trades produced edge. The Williams %R finding hints that something like that might exist in narrow conditions. But I'm not going to claim it does until I have many more trades' worth of data on the new positioning. The user gets the context. The user decides.

What I'm asking from you

If you're a crypto trader, indie founder, journalist, academic, or skeptic — I'd love your take on this data. The audit is reproducible. The methodology is documented. The app is on the App Store and Play Store. The new positioning ships this week.

If you find a methodological error in this audit, I want to know. I'd rather correct the record than be wrong publicly. If you've run a similar audit on your own engine and arrived at different conclusions, I want to see your data. If you think the divergence pivot is wrong, tell me why.

This is the part of building in public that hurts. Publishing your failure with the receipts.

But it's also the only honest version of the product I can build. The choice was: ship the broken signals product and take customers' money, or ship the honest divergence product and let users decide. **I picked the honest one.**

The 12-indicator engine doesn't predict prices. The academic literature has been telling us that for 30 years. My data confirms it on 3,767 trades. The product I'm shipping next week is the one that doesn't pretend otherwise.

— Murat