

Machine Learning-Assisted Validation of Experimental VWAP Trading Signals Under Changing Market Conditions

Venkata Maheswara Rao Atchula*, Dr. Pawan Kumar Pareek

Nirwan University, Jaipur, Rajasthan.

ABSTRACT

Financial markets exhibit changing trends, volatility, and market regimes, making deterministic trading signals difficult to evaluate consistently across time. This study investigates a machine-learning-assisted validation framework applied to an experimental VWAP trading strategy on BTCUSD. The rule-based strategy generates candidate BUY and SELL signals when price crosses the session VWAP, subject to a moving-average trend filter and an ATR-based expansion filter. A machine-learning validator is then used as an additional acceptance layer, with multiple confidence thresholds evaluated across several timeframes. The experimental design uses seven years of historical data, with four years allocated to training/development and three years reserved for testing. The complete Experimental VWAP registry contains 25 ML-test configurations covering 1-minute, 3-minute, 5-minute, 15-minute and 1-hour timeframes, with confidence thresholds of 0.50, 0.53, 0.55, 0.57 and 0.60, together with five training configurations. Across the 25 ML-test configurations, 53,818 orders are recorded; eight configurations report positive net realized results and seventeen report negative net realized results. The study therefore treats the registry as experimental evidence of configuration sensitivity rather than as evidence that a particular threshold or timeframe is universally optimal.

Keywords: VWAP, Machine Learning, Algorithmic Trading, Signal Validation, BTCUSD, Feature Engineering, Confidence Threshold, Backtesting

INTRODUCTION

Algorithmic trading systems commonly combine deterministic rules with statistical or machine-learning components. A deterministic strategy can provide a transparent mechanism for generating candidate trades, while a validation layer can be used to decide whether a candidate should proceed to execution. This separation is useful when the research question concerns signal quality rather than direct price forecasting.

VWAP is widely used as a reference price and execution benchmark. Academic work has examined VWAP tracking, optimal execution, market volume dynamics and the relationship between algorithmic trading and VWAP. [1–4] The present study uses VWAP differently: it is the primary signal-generation reference in a rule-based trading strategy, after which an ML validator evaluates the candidate trade.

The study is referred to as the Experimental VWAP experiment throughout this paper. The objective is not to establish a universally optimal VWAP configuration. Instead, the objective is to document how an ML

validation layer behaves when the underlying VWAP strategy is evaluated across multiple timeframes and confidence thresholds.

The supplied strategy implementation maintains a daily VWAP accumulator, uses an ATR filter and a moving-average trend filter, supports time-based exits, and passes engineered entry features to an ML validation client before an entry order is sent.

2. Research Questions and Objectives

- RQ1: How does ML-based validation affect the observed performance characteristics of the VWAP signal strategy?
- RQ2: How does validation behaviour vary across 1m, 3m, 5m, 15m and 1h timeframes?
- RQ3: How does changing the ML confidence threshold affect trade frequency, profitability and drawdown?
- RQ4: Do the observed experimental results remain consistent across the available configurations?

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.



- RQ5: What methodological limitations should be considered before interpreting the backtest results as evidence of deployable trading performance?

3. Literature Review

3.1 VWAP as a Trading and Execution Reference

VWAP is commonly defined as an average transaction price weighted by traded volume over a specified horizon. Research on VWAP strategies has focused heavily on execution, market impact, volume forecasting and tracking error. Dynamic-volume approaches emphasize that the intraday volume profile is an important component of VWAP execution. [1] Optimal-execution studies formulate VWAP tracking as a risk and cost minimization problem rather than as a directional trading rule. [2,3]

3.2 VWAP and Algorithmic Trading

Empirical research has also examined the interaction between algorithmic trading and VWAP. Evidence from equity markets indicates that algorithmic execution can be sensitive to prevailing VWAP conditions. [4] Recent research has extended VWAP execution into machine-learning and cryptocurrency settings, illustrating the difficulty of modelling volume and price dynamics in volatile digital-asset markets. [5]

3.3 Machine Learning in Financial Applications

Machine learning has been widely studied across financial forecasting, portfolio management, cryptocurrency and foreign-exchange applications. Reviews emphasize the diversity of datasets, features, validation procedures and evaluation measures used across financial ML research. [6,7] The present study adopts a narrower use of ML: the model is a validation layer applied after a deterministic VWAP signal rather than a standalone price predictor.

3.4 Research Gap

The specific experimental question addressed here is whether a machine-learning acceptance layer changes the observed behaviour of a rule-based VWAP strategy across timeframes and confidence thresholds. The focus is therefore on configuration sensitivity, trade filtering and out-of-sample behaviour rather than on predicting the future price directly.

RESEARCH METHODOLOGY

The study follows a computational experimental methodology with a development/training period and an unseen test period. The seven-year BTCUSD dataset is divided into four years for training/development and three years for testing, as specified for the Experimental VWAP study.

The experimental pipeline is:

VWAP SIGNAL GENERATION → FEATURE ENGINEERING → ML VALIDATION → SIGNAL ACCEPTANCE/REJECTION → TRADE MANAGEMENT

Stage	Implementation in Experimental VWAP
1. Market data	Seven years of BTCUSD historical data; four years training/development and three years testing.
2. Candle construction	The backtest runner routes VWAP as a candle strategy and builds candles according to the selected timeframe.
3. Signal generation	Price crossing the daily-reset VWAP generates a candidate BUY/SELL signal, subject to trend and ATR filters.
4. Feature engineering	Base, moving-average and engineered contextual features are constructed before validation.
5. ML validation	The MLValidatorClient evaluates the feature dictionary and returns an allowed flag, score, model and prediction information.
6. Trade management	Accepted trades are passed to risk management and exit handling; the strategy supports SL/TP/TSL through RiskEngine and a time exit.
7. Evaluation	The supplied registry reports orders, wins, losses, win percentage, total profit, total loss, net realized and maximum drawdown.

4.1 VWAP Signal Generation

The strategy resets its cumulative VWAP state when the calendar date changes. It then updates cumulative price and cumulative volume and computes VWAP from the running accumulators. The implementation uses unit volume for each candle, explicitly described in the source code as $\text{volume} = 1$ (tick-normalised).

A BUY candidate is generated when the previous close is at or below VWAP and the current close moves above VWAP. A SELL candidate is generated when the previous close is at or above VWAP and the current close moves below VWAP. Both directions are additionally filtered by the configured moving-average trend mode and ATR-expansion condition.

This implementation detail is important for interpretation: because cumulative volume is incremented by one for every candle, the experimental indicator is computationally a session-reset cumulative average of candle prices rather than a conventional exchange-volume-weighted VWAP. The paper retains the Experimental VWAP name used by the study while making this implementation characteristic explicit.

4.2 Moving-Average and ATR Filters

The strategy uses a moving-average engine with configurable short and long lengths. The MA feature module computes short MA, long MA, MA slope, MA bias and crossover state.

The ATR filter computes a rolling true-range-based ATR and compares the current ATR with a recent ATR baseline. Trading is permitted when the configured ATR expansion condition does not indicate excessive expansion. The supplied configuration defaults include ATR length 14, ATR baseline length 20 and an expansion multiplier of 1.5.

4.3 Feature Engineering

The feature builder combines strategy metadata, trade context, base features and moving-average features. Base features include MA slope, absolute MA slope, market regime and risk/reward quantities; the MA module adds short MA, long MA, slope, bias and crossover.

The engineered feature layer derives trend strength, momentum, volatility, risk pressure, reward pressure, signal quality and market regime. The feature builder obtains these values from runtime context or computes defaults when they are not supplied.

The strategy runtime context contains VWAP and distance-from-VWAP values, but the supplied feature builder does not explicitly add those two fields to the returned feature dictionary. Therefore, this paper does not claim that the ML model directly consumed VWAP distance as a model feature. This is a reproducibility point that should be addressed in a future controlled experiment.

4.4 ML Validation Layer

For each candidate entry, the strategy constructs the feature dictionary and calls `MLValidatorClient.validate_trade()`. The returned result contains an allowed decision, score, reason, model and prediction. If the validator does not allow the trade, the strategy returns without sending the entry order.

The ML configuration is externally supplied through the strategy definition. The backtest runner reads `enable_ml`, `ml_model`, `ml_endpoint`, `ml_threshold`, `timeout` and `fail-mode` settings before constructing the validator. Because the supplied files do not establish a single fixed model name for every Experimental VWAP run, the paper refers to the component generically as the ML validator rather than assuming a specific algorithm.

4.5 Data Partition and Experimental Dimensions

Dimension	Experimental setting
Asset	BTCUSD
Total historical period	7 years
Training/development	4 years
Unseen testing	3 years
Timeframes represented	1m, 3m, 5m, 15m, 1h
ML thresholds	0.50, 0.53, 0.55, 0.57, 0.60
Reported metrics	Orders, wins, losses, win %, profit, loss, net realized, maximum drawdown
Experiment name	Experimental VWAP

5. Experimental Data Integrity

The complete Experimental VWAP markdown export supplied with this study contains the full experimental registry, including all 30 reported rows: five training configurations and 25 ML-test configurations. The ML-test grid is complete across five timeframes and five confidence thresholds.

The complete registry therefore supports direct descriptive aggregation by timeframe and by ML confidence threshold. No missing order counts, profits, losses, net results or drawdowns have been reconstructed or estimated.

All numerical summaries in the following sections are calculated directly from the complete supplied registry. Aggregations across configurations are descriptive and should not be interpreted as the result of one pooled trading account.

6. Experimental Results

The complete ML-test portion contains 25 configurations and 53,818 recorded orders, comprising 18,560 wins and 35,238 losses. The order-weighted win rate across these independent configurations is 34.49%. The arithmetic sum of reported net realized values is -364,846.31, while the largest individual reported maximum drawdown is 52,289.84. These are descriptive aggregates across independent configurations and must not be interpreted as the result of one pooled trading account.

6.1 Results by Timeframe

Timeframe	Runs*	Orders*	Mean Win %	Net Realized*	Mean Max DD*	Positive Runs*
1m	5	11,501	31.55%	22,754.44	4,957.71	5/5
3m	5	7,873	33.32%	-10,202.29	10,569.29	2/5
5m	5	9,906	33.08%	-53,581.20	16,207.15	1/5
15m	5	14,320	36.40%	-148,080.80	37,830.43	0/5
1h	5	10,218	37.19%	-175,736.46	43,206.67	0/5

* Complete ML-test registry: all five threshold configurations are represented at each of the five timeframes.

6.2 Results by ML Confidence Threshold

Threshold	Runs*	Orders*	Mean Win %	Net Realized*	Mean Max DD*	Positive Runs*
0.50	5	13,962	34.75%	-71,747.45	25,113.17	1/5
0.53	5	11,920	34.38%	-65,482.90	23,290.26	2/5
0.55	5	10,632	34.09%	-71,236.21	20,537.54	2/5
0.57	5	9,505	34.44%	-65,764.97	19,563.96	1/5
0.60	5	7,799	33.88%	-90,614.78	24,266.33	2/5

* Threshold totals use all five timeframes for each threshold; each threshold therefore contains five ML-test configurations.

6.3 Positive Recoverable ML Configurations

Configuration	Orders	Win %	Net Realized	Max DD
56 / 1m	1,966	31.89%	7,494.20	1,998.32
53 / 1m	2,540	31.46%	6,846.43	4,808.46
55 / 1m	2,226	31.18%	4,736.33	3,591.33
60 / 3m	943	34.04%	3,625.55	4,969.64
50 / 1m	3,200	31.16%	3,168.95	9,737.36
53 / 5m	2,217	34.06%	1,031.24	12,462.50
55 / 3m	1,580	33.48%	624.77	8,924.60
60 / 1m	1,569	32.06%	508.53	4,653.08

Eight of the 25 complete ML-test configurations report positive net realized values. The positive observations occur across the 1m, 3m and 5m groups. The largest positive reported net is 7,494.20 for the 1m

configuration at threshold 0.57. These are observations within the supplied experiment registry, not claims of universal superiority.

6.4 Training Rows Available in the Supplied Export

Training TF	Orders	Wins	Losses	Win %	Net Realized	Max DD
1m	12,083	4,134	7,921	34.21%	-16,323.09	23,356.60
3m	6,890	2,551	4,337	37.02%	7,794.29	13,188.50
5m	5,190	2,032	3,157	39.15%	-12,543.82	18,962.11
15m	2,989	1,308	1,681	43.76%	-26,750.66	35,733.25
1h	1,315	614	701	46.69%	17,314.81	10,789.83

All five training configurations are present in the complete registry. Their reported net realized values are positive for 1h and 3m and negative for 1m, 5m and 15m. Because these training results belong to the development period, they are not treated as out-of-sample evidence.

DISCUSSION

The recoverable experimental results show substantial configuration sensitivity. The observed net realized values range from positive to materially negative across the tested thresholds and timeframes. This dispersion is consistent with the study objective of evaluating validation behaviour rather than assuming that a higher confidence threshold necessarily produces better trading outcomes.

Timeframe behaviour is non-uniform. The complete 1m ML-test group has positive aggregate net realized across all five thresholds, while the 3m, 5m, 15m and 1h groups have negative aggregate net realized. The size of the dispersion also increases materially at the longer timeframes, where the reported maximum drawdowns are larger.

Threshold behaviour is similarly non-monotonic. Increasing the threshold does not produce a simple monotonic improvement in net realized value or drawdown. This is an important observation for a

validation-layer architecture: a stricter acceptance rule changes the population of trades, but stricter filtering alone does not guarantee better aggregate outcomes.

The strategy architecture provides a clear separation between signal generation and validation. The candidate signal originates from the VWAP crossing rule, while the ML layer decides whether the candidate is allowed to proceed. The source implementation also records ML score, prediction and model metadata in the entry feature payload, providing a useful basis for future audit and error analysis.

At the same time, the current feature pipeline reveals an important research-design issue. VWAP and distance-from-VWAP are present in runtime context, but the supplied feature builder does not explicitly place them into the returned feature dictionary. This means that the validation model's direct exposure to VWAP-specific information is not established by the supplied code. A controlled follow-up experiment should explicitly include VWAP level, normalized VWAP distance and possibly volume-related features if true volume data are available.

8. Threats to Validity and Limitations

- Historical backtest only: the experiment does not establish live execution performance.

- Execution effects: the supplied registry does not establish a complete model of transaction costs, slippage, spread, latency or market impact.
- Multiple configurations: repeated testing of thresholds and timeframes creates opportunities for selection bias and backtest overfitting. The observed results should therefore be treated as experimental evidence requiring further controlled validation.
- Perform forward paper-trading validation before considering any live deployment.

CONCLUSION

This study presents the Experimental VWAP framework as a rule-based VWAP signal generator combined with an ML validation layer. The seven-year BTCUSD study design separates four years for training/development and three years for unseen testing, while the experimental grid spans multiple intraday and hourly timeframes and ML confidence thresholds.

The recoverable experimental evidence demonstrates substantial variation across configurations. Among the 22 complete ML-test rows available in the supplied export, five report positive net realized values, while the remaining rows are negative. The results therefore show that ML threshold and timeframe materially affect observed trade activity and performance

Appendix A. Recoverable Experimental VWAP Registry

The following table lists the complete rows recoverable from the supplied markdown export. The source export contains rows 6–30; rows 1–5 are not fully present. Values are reproduced from the supplied data without normalization.

Row	Configuration	Orders	Wins	Losses	Win %	Net	Max DD
1	train_1h	1,315	614	701	46.69%	17,314.81	10,789.83
2	train_3m	6,890	2,551	4,337	37.02%	7,794.29	13,188.50
3	test_ml57_1m	1,966	627	1,338	31.89%	7,494.20	1,998.32
4	test_ml53_1m	2,540	799	1,740	31.46%	6,846.43	4,808.46
5	test_ml55_1m	2,226	694	1,531	31.18%	4,736.33	3,591.33
6	test_ml60_3m	943	321	621	34.04%	3,625.55	4,969.64
7	test_ml50_1m	3,200	997	2,201	31.16%	3,168.95	9,737.36
8	test_ml53_5m	2,217	755	1,460	34.06%	1,031.24	12,462.50
9	test_ml55_3m	1,580	529	1,051	33.48%	624.77	8,924.60
10	test_ml60_1m	1,569	503	1,066	32.06%	508.53	4,653.08
11	test_ml57_3m	1,324	455	869	34.37%	-2,018.82	9,265.81
12	test_ml53_3m	1,830	589	1,240	32.19%	-2,493.14	12,271.48
13	test_ml50_3m	2,196	714	1,480	32.51%	-9,940.65	17,414.92
14	test_ml60_5m	1,388	439	948	31.63%	-11,271.36	13,702.75
15	train_5m	5,190	2,032	3,157	39.15%	-12,543.82	18,962.11
16	test_ml50_5m	2,613	896	1,716	34.29%	-12,701.62	18,217.63

17	test_ml57_5m	1,709	562	1,146	32.88%	-13,827.35	16,585.40
18	train_1m	12,083	4,134	7,921	34.21%	-16,323.09	23,356.60
19	test_ml55_5m	1,979	644	1,334	32.54%	-16,812.11	20,067.49
20	test_ml57_15m	2,660	959	1,700	36.05%	-21,357.28	27,511.35
21	test_ml50_1h	2,528	970	1,558	38.37%	-22,179.98	40,410.18
22	test_ml55_15m	2,841	1,030	1,810	36.25%	-23,568.31	29,547.44
23	train_15m	2,989	1,308	1,681	43.76%	-26,750.66	35,733.25
24	test_ml50_15m	3,425	1,281	2,143	37.40%	-30,094.15	39,785.74
25	test_ml53_15m	3,111	1,144	1,966	36.77%	-30,680.67	40,017.79
26	test_ml57_1h	1,846	683	1,163	37.00%	-36,055.72	42,458.92
27	test_ml55_1h	2,006	742	1,264	36.99%	-36,216.89	40,556.83
28	test_ml53_1h	2,222	832	1,390	37.44%	-40,186.76	46,891.09
29	test_ml60_1h	1,616	584	1,032	36.14%	-41,097.11	45,716.33
30	test_ml60_15m	2,283	811	1,471	35.52%	-42,380.39	52,289.84

Note: the first line of the supplied export is a continuation fragment containing 3,591.33 and does not provide enough fields to reconstruct the preceding row. It is therefore not treated as a complete observation.

REFERENCES

- Białkowski, J., Darolles, S., & Le Fol, G. (2008). Improving VWAP strategies: A dynamic volume approach. *Journal of Banking & Finance*, 32 (9), 1709–1722. <https://doi.org/10.1016/j.jbankfin.2007.09.023>
- Busseti, E., & Boyd, S. (2015). Volume Weighted Average Price Optimal Execution. *arXiv:1509.08503*.
- Cartea, Á., & Jaimungal, S. (2016). A Closed-Form Execution Strategy to Target Volume Weighted Average Price. *SIAM Journal on Financial Mathematics*, 7, 760–785.
- Zhou, H., Kalev, P. S., & Frino, A. (2020). Algorithmic trading in turbulent markets. *Pacific-Basin Finance Journal*, 62, 101358.
- Genet, R. (2025). Deep Learning for VWAP Execution in Crypto Markets: Beyond the Volume Curve. *arXiv:2502.13722*.
- Nazareth, N., & Reddy, Y. V. R. (2023). Financial applications of machine learning: A literature review. *Expert Systems with Applications*, 219, 119640. <https://doi.org/10.1016/j.eswa.2023.119640>
- Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197, 116659.
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

HOW TO CITE: Venkata Maheswara Rao Atchula, Dr. Pawan Kumar Pareek, Machine Learning-Assisted Validation of Experimental VWAP Trading Signals Under Changing Market Conditions, *Int. J. Sci. R. Tech.*, 2026, 3 (10), 563-569. <https://doi.org/10.5281/zenodo.23256717>