

Machine Learning-Assisted Validation of MACD Trading Signals Across Multiple Timeframes

Venkata Maheswara Rao Atchula¹, Dr. Pawan Kumar Pareek²
^{1,2} Member, Nirwan University, Jaipur, Rajasthan October 2026

Abstract—Algorithmic trading systems based on conventional technical indicators can generate frequent signals during changing market conditions, including signals that may be difficult to distinguish from low-quality or short-lived opportunities. This study investigates a machine-learning-assisted validation framework for Moving Average Convergence Divergence (MACD) trading signals. The MACD strategy first generates candidate trading decisions using a conventional rule-based signal process. A machine-learning validation layer, implemented using XGBoost in the experimental framework, evaluates the candidate signal and applies a configurable confidence threshold before the trade is accepted. The study uses a seven-year BTCUSDT research dataset with four years used for training/development and three years used for out-of-sample testing. The experimental registry evaluates five timeframes—1-minute, 3-minute, 5-minute, 15-minute and 1-hour—and five ML confidence thresholds: 0.50, 0.53, 0.55, 0.57 and 0.60. The supplied results contain 25 ML test configurations and corresponding non-ML test configurations. Across the five timeframes, the ML configurations consistently reduce the number of executed orders and the reported maximum drawdown while producing modest increases in average win rate. The results therefore support interpreting ML primarily as a signal-selectivity and risk-control layer rather than as a universal profitability maximizer. The study provides an empirical framework for evaluating whether ML validation adds incremental value to a deterministic MACD trading strategy under multiple timeframes and confidence thresholds.

Index Terms— MACD, Machine Learning, XGBoost, Algorithmic Trading, Signal Validation, BTCUSDT, Technical Analysis, Risk Management, Out-of-Sample Testing.

I. INTRODUCTION

Technical indicators remain widely used in algorithmic trading because they provide deterministic

rules for transforming price information into trading decisions. The Moving Average Convergence Divergence (MACD) indicator is a classical trend-following and momentum-oriented tool based on the relationship between fast and slow exponential moving averages and a signal line. Its popularity has also motivated empirical studies examining MACD trading rules across markets and parameter settings.

A practical difficulty with deterministic signal systems is that a rule can generate many candidate trades, including trades that occur in weak trends, transitional regimes or short-lived movements. A machine-learning model can be positioned after the rule-based signal generator so that it does not replace the trading rule, but instead acts as a validation layer. The resulting architecture separates signal generation from signal acceptance.

The present study applies this concept to MACD. The MACD strategy generates a candidate BUY or SELL event. Market and technical features are supplied to the ML validator. The validator produces a confidence score, and only signals meeting the configured threshold are permitted to continue through the trading system. This makes it possible to study the effect of ML on trade frequency, win rate and drawdown without treating ML as a direct price-forecasting replacement for MACD.

The experimental registry contains configurations for 1-minute, 3-minute, 5-minute, 15-minute and 1-hour timeframes. The ML test configurations use confidence thresholds of 0.50, 0.53, 0.55, 0.57 and 0.60. The supplied configuration records indicate MACD parameters of 12, 26 and 9, a moving-average trend filter using 11 and 21 periods, ATR and range filters, and risk-management controls. The ML test configurations use the three-year test file `btc_test_3y.parquet`, while the training configurations use `btc_train_4y.parquet`.

1.1 MACD Signal Generation

MACD is computed from a fast exponential moving average and a slow exponential moving average. In the experimental configuration, the fast length is 12, the slow length is 26 and the signal length is 9. The MACD component therefore provides the deterministic candidate signal, while additional trend, volatility and risk-management controls operate within the broader trading configuration.

The experimental implementation also records an 11-period and 21-period moving-average trend filter, ATR-based filtering parameters, a range filter and explicit stop-loss/take-profit settings. These components are retained consistently within the supplied MACD experiment family. The purpose of the ML layer is therefore to test an additional validation stage within an already structured rule-based trading system.

1.2 ML Validation Architecture

The proposed decision pipeline is: MACD SIGNAL GENERATION → FEATURE ENGINEERING → XGBOOST VALIDATION → CONFIDENCE THRESHOLD → SIGNAL ACCEPTANCE/REJECTION → TRADE MANAGEMENT. The architecture preserves the deterministic MACD rule as the source of candidate opportunities. Machine learning does not directly generate a new trading direction; it determines whether the candidate signal satisfies the learned validation condition.

II. RESEARCH QUESTIONS AND OBJECTIVES

RQ1: How does ML-based validation affect the trading characteristics of MACD-generated signals?

RQ2: How does ML validation affect trade frequency across 1-minute, 3-minute, 5-minute, 15-minute and 1-hour timeframes?

RQ3: How does ML validation affect observed win rate and maximum drawdown?

RQ4: How does changing the ML confidence threshold from 0.50 to 0.60 affect trade activity and performance measures?

RQ5: Do the observed effects of ML remain directionally consistent across multiple timeframes?

RQ6: Can ML validation be interpreted as a signal-selectivity and risk-management layer rather than solely as a profitability-enhancement mechanism?

III. LITERATURE REVIEW

3.1 MACD and Technical Trading Rules

MACD was introduced by Gerald Appel as a technical trading method based on moving-average convergence and divergence. Subsequent empirical work has examined MACD rules using different crossing definitions, markets and parameter settings. Li, Yu and Chong studied MACD profitability under different crossing rules and reported that performance varied across markets and periods. Hung compared several MACD trading strategies and emphasized that strategy design and trading frequency can materially affect results.

3.2 MACD Parameter and Strategy Studies

Research has also examined the sensitivity of MACD to parameter selection. Kang investigated parameter optimization and modified MACD trading rules in the Nikkei 225 futures market, while later work considered MACD parameter ranges across multiple stock-index futures. These studies demonstrate that MACD performance cannot be separated from configuration choices and evaluation methodology.

3.3 Machine Learning for Financial Trading

Machine learning has been applied to financial forecasting, trading-rule refinement and decision support. Prior studies have used machine learning or evolutionary optimization to modify MACD-based approaches. However, a distinction exists between replacing a technical indicator with a predictive model and using machine learning as a validation layer after a deterministic trading rule. The latter is the focus of the present study.

3.4 Out-of-Sample Evaluation and Data-Snooping

Backtesting multiple configurations creates a risk of selecting a configuration because of historical performance rather than because it represents a robust relationship. Sullivan, Timmermann and White demonstrated the importance of accounting for data-snooping in technical trading-rule evaluation, while Bailey and co-authors highlighted the probability of backtest overfitting. The present study therefore separates training/development data from the three-year test dataset and reports the complete tested threshold set rather than presenting a single selected configuration as universally optimal.

3.5 Research Gap

The literature contains extensive research on MACD trading rules and growing work on machine learning for financial decision-making. The narrower gap addressed here is the empirical evaluation of ML as a validation layer positioned after a conventional MACD signal generator. The study evaluates this architecture across multiple intraday and hourly timeframes and multiple confidence thresholds, with emphasis on trade selectivity, win rate and maximum drawdown.

IV. RESEARCH METHODOLOGY

The study follows a computational experimental evaluation approach combined with a Design Science Research Methodology-oriented artifact perspective. The artifact is an ML-assisted MACD signal-validation framework. The experiment varies the timeframe and ML confidence threshold while retaining the MACD-based signal generation and the supplied supporting strategy configuration.

4.1 Methodological Pipeline

TABLE I METHODOLOGICAL PIPELINE

Stage	Method
Market data	Seven-year BTCUSDT dataset; 4 years training/development and 3 years unseen testing.
MACD generation	Fast=12, slow=26, signal=9.
Context filters	MA trend filter, ATR filter and range filter.
ML validation	XGBoost validation layer produces a confidence score.
Threshold gate	Candidate accepted when the configured confidence threshold is satisfied.
Trade management	Stop-loss and take-profit controls are applied to accepted signals.
Evaluation	Orders, wins/losses, win rate, profit, visible net realized and maximum drawdown.

4.2 Data Partition

TABLE II DATA PARTITION

Role	File / Period	Purpose
Training / development	btc_train_4y.parquet / 4 years	Model development and configuration preparation.
Unseen testing	btc_test_3y.parquet / 3 years	Out-of-sample evaluation.

4.3 Experimental Design.

TABLE III EXPERIMENTAL FACTORS AND PARAMETERS

Factor	Values
Timeframe	1m, 3m, 5m, 15m, 1h
ML threshold	0.50, 0.53, 0.55, 0.57, 0.60
MACD	12 / 26 / 9
MA trend filter	11 / 21
ATR filter	Length 14; base length 20; multiplier 1.5
Range filter	Base length 50; multiplier 1.1
Risk management	Stop-loss and take-profit enabled in supplied ML tests
ML role	Validation of candidate MACD signals

The ML test registry contains five thresholds for each of five timeframes, producing 25 ML test configurations. The corresponding non-ML test configurations provide a source-defined baseline for each timeframe. The threshold set is 0.50, 0.53, 0.55, 0.57 and 0.60.

4.4 MACD and ML Decision Process

For each candidate signal, the deterministic MACD layer identifies the directional opportunity. Feature engineering then provides the ML validator with the representation used by the model. The ML validator produces a confidence score. Threshold gating controls whether the candidate is passed to the execution layer. Increasing the threshold therefore changes selectivity rather than changing the underlying MACD signal-generation rule.

4.5 Evaluation Metrics

TABLE IV EVALUATION METRICS

Metric	Definition
Orders	Number of recorded orders.
Wins / Losses	Counts reported by the experiment registry.
Win percentage	Wins divided by recorded orders.
Total profit	Aggregate profitable trade amount reported.
Net realized	Net realized value where exposed by the source.
Maximum drawdown	Largest reported drawdown.
Order reduction	Relative reduction in average ML orders vs. non-ML row.
Drawdown reduction	Relative reduction in average ML maximum drawdown vs. non-ML row.

V. EXPERIMENTAL RESULTS

TABLE V-A BASELINE AND ML AVERAGE COMPARISON

TF	Base Orders	ML Avg Orders	Base Win %	ML Avg Win %	Base Max DD	ML Avg Max DD
1m	3,181	721.8	24.05%	27.03%	18,308.56	4,336.41
3m	1,548	261.4	21.83%	23.33%	22,724.18	6,849.75
5m	1,239	355.6	23.89%	24.28%	19,433.80	7,413.44
15m	1,076	257.4	27.88%	29.26%	38,244.37	14,604.53
1h	1,159	571.4	33.30%	34.45%	47,469.04	32,236.81

TABLE V-B RELATIVE DESCRIPTIVE CHANGES

TF	Orders ↓	Win Δ (pp)	DD ↓
1m	77.3%	+2.98	76.3%
3m	83.1%	+1.50	69.9%
5m	71.3%	+0.39	61.9%
15m	76.1%	+1.38	61.8%
1h	50.7%	+1.15	32.1%

Note: ML averages are arithmetic means across the five tested thresholds for each timeframe. They are descriptive summaries, not pooled portfolio results.

The supplied updated registry contains 25 ML test configurations across five timeframes and five confidence thresholds. The results show a consistent reduction in trade activity when the ML layer is enabled. The average ML configuration also reports a higher win percentage and lower maximum drawdown than the corresponding non-ML test row at every tested timeframe. The updated registry also exposes net-realized values for all 25 ML configurations and all five non-ML test configurations.

TABLE V-C DESCRIPTIVE NET-REALIZED COMPARISON

Timeframe	Non-ML Net Realized	Average ML Net Realized	Change
1m	-16,559.36	256.46	+16,815.82
3m	-15,722.62	-3,349.90	+12,372.72
5m	-1,587.81	1,739.99	+3,327.80
15m	-24,892.16	2,581.63	+27,473.79
1h	-32,968.65	26,824.12	+6,144.53

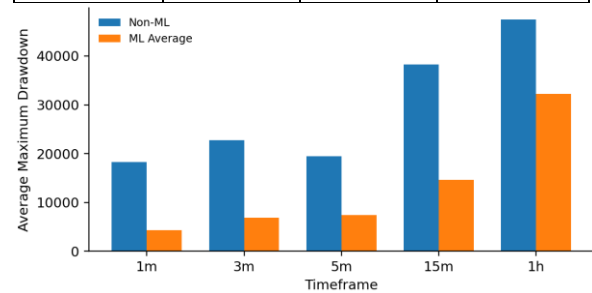


Figure 3. Average ML maximum drawdown versus the source-defined non-ML test configuration.

The 1-minute non-ML test configuration records 3,181 orders, a 24.05% win rate, net realized of -16,559.36 and maximum drawdown of 18,308.56. Across the five ML thresholds, the average is 721.8 orders, 27.03% win rate, net realized of 256.46 and maximum drawdown of 4,336.41. Descriptively, this represents 77.3% fewer orders, a 2.98 percentage-point increase in average win rate, a 76.3% reduction in average maximum drawdown and an improvement in average net realized from -16,559.36 to 256.46.

The threshold results also show sensitivity. The 0.50 threshold retains the largest number of ML trades at 914, while 0.60 produces 547 trades. The observed win rate ranges from 25.96% to 27.57%, indicating that increased selectivity does not produce a monotonic increase in win rate.

5.2 Three-Minute Results

The 3-minute non-ML test configuration records 1,548 orders, a 21.83% win rate, net realized of -15,722.62 and maximum drawdown of 22,724.18. The five ML configurations average 261.4 orders, 23.33% win rate, net realized of -3,349.90 and maximum drawdown of 6,849.75. The corresponding descriptive changes are an 83.1% reduction in orders, a 1.50 percentage-point increase in win rate and a 69.9% reduction in average maximum drawdown; average net realized is less negative by 12,372.72.

5.3 Five-Minute Results

The 5-minute non-ML test configuration records 1,239 orders, a 23.89% win rate, net realized of -1,587.81 and maximum drawdown of 19,433.80. The ML configurations average 355.6 orders, 24.28% win rate, net realized of 111.47 and maximum drawdown of 7,413.44. Thus, trade activity falls by approximately 71.3%, average win rate increases by 0.39 percentage points, average maximum drawdown falls by approximately 61.9%, and average net realized changes from -1,587.81 to 111.47.

5.4 Fifteen-Minute Results

The 15-minute non-ML test configuration records 1,076 orders, a 27.88% win rate, net realized of -24,892.16 and maximum drawdown of 38,244.37. The ML configurations average 257.4 orders, 29.26% win rate, net realized of 2,581.63 and maximum drawdown of 14,604.53. The descriptive changes are a 76.1% reduction in orders, a 1.38 percentage-point increase

in average win rate, a 61.8% reduction in average maximum drawdown and an improvement in average net realized of 27,473.79.

5.5 One-Hour Results

The 1-hour non-ML test configuration records 1,159 orders, a 33.30% win rate, net realized of -32,968.65 and maximum drawdown of 47,469.04. The ML configurations average 571.4 orders, 34.45% win rate, net realized of -26,824.12 and maximum drawdown of 32,236.81. The corresponding changes are a 50.7% reduction in orders, a 1.15 percentage-point increase in average win rate, a 32.1% reduction in average maximum drawdown and an improvement in average net realized of 6,144.53.

5.6 ML Confidence Threshold Sensitivity

The threshold analysis does not show a simple monotonic relationship between confidence threshold and win rate. Higher thresholds consistently reduce the number of accepted trades, but the win-rate response varies by timeframe. This indicates that the confidence threshold functions primarily as a selectivity control. A threshold that produces fewer trades should not automatically be interpreted as a better configuration.

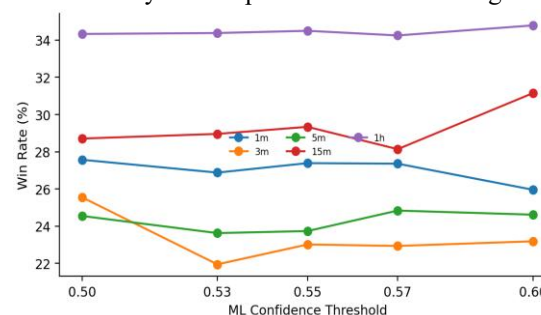


Figure 4. Win-rate sensitivity to the ML confidence threshold across the five tested timeframes.

The updated registry exposes net-realized values for all 25 ML configurations and all five non-ML test configurations. Eight of the 25 ML configurations have positive net-realized values and 17 have negative values. Descriptively averaging the five ML thresholds gives positive average net realized at 1m (256.46), 5m (111.47) and 15m (2,581.63), while the averages remain negative at 3m (-3,349.90) and 1h (-26,824.12). The corresponding non-ML test values are negative at all five timeframes. These averages compare separate configurations and should not be

interpreted as a pooled portfolio result or a statistical test.

VI. DISCUSSION

The empirical evidence supports a specific interpretation of the ML layer. Across all five tested timeframes, the ML configurations accept substantially fewer trades than the corresponding non-ML test configuration. At the same time, average win rate is higher and average maximum drawdown is lower. The updated net-realized data add an important qualification: average ML net realized is positive at 1m, 5m and 15m, but remains negative at 3m and 1h. The magnitude of the effect therefore varies with timeframe.

The strongest trade-reduction effect occurs at 3 minutes, where average ML order count is approximately 83% below the non-ML test count. The largest average drawdown reduction occurs at 1 minute, where the reduction is approximately 76%. At 1 hour, the effect is smaller but remains directionally consistent, with approximately 51% fewer orders and 32% lower average maximum drawdown.

These results suggest that ML validation behaves as a selective gate rather than simply as a profit-generating replacement for MACD. A lower number of accepted signals means that aggregate profit cannot be interpreted independently of opportunity count, exposure and risk. Consequently, the relevant empirical question is not only whether ML produces more profit, but whether it changes the quality and risk characteristics of the signals that are permitted to reach the execution layer.

The results also reinforce the importance of threshold sensitivity. The tested thresholds from 0.50 to 0.60 produce different trade counts and risk profiles, while win-rate changes are non-monotonic. This argues against selecting a threshold solely from a single historical result.

VII. RESEARCH CONTRIBUTIONS

- A modular ML-assisted MACD architecture in which machine learning validates deterministic MACD candidate signals rather than replacing the signal generator.
- A multi-timeframe experimental registry covering 1m, 3m, 5m, 15m and 1h configurations.

- A systematic sensitivity analysis across five ML confidence thresholds from 0.50 to 0.60.
- Empirical measurement of trade-frequency reduction, win-rate change and maximum-drawdown change associated with ML validation.
- Explicit separation of training/development data from a three-year unseen test dataset.
- A risk-selectivity interpretation of ML validation that does not require the claim that ML universally increases aggregate profitability.

VIII. THREATS TO VALIDITY AND LIMITATIONS

- Historical backtesting: the results are based on historical BTCUSDT data and do not establish live-market performance.
- Single asset: the current empirical evaluation concerns BTCUSDT and may not generalize to equities, commodities, foreign exchange or other cryptocurrencies.
- Configuration selection: testing multiple thresholds and timeframes introduces a multiple-comparison and selection issue.
- Execution realism: the supplied registry does not independently document complete treatment of slippage, exchange latency, market impact or order-book effects.
- Updated source completeness: the revised registry exposes the previously unavailable net-realized cells, allowing a complete descriptive comparison of the 25 ML configurations and five non-ML test configurations.
- Metric interpretation: threshold averages are descriptive averages across separate configurations and are not a single combined trading account.
- Model specification: the present registry identifies the ML-enabled configurations and thresholds; further documentation of feature definitions, training labels, class balance, hyperparameters and probability calibration would strengthen reproducibility.

IX. FUTURE RESEARCH

- Evaluate the complete factorial design of ML ON/OFF, MA ON/OFF and RM ON/OFF using identical test periods.

- Document the full feature set, XGBoost hyperparameters, training labels and probability-calibration procedure.
- Apply statistical tests or resampling procedures to quantify whether observed changes are statistically distinguishable from baseline variation.
- Model transaction costs, slippage and latency explicitly.
- Evaluate additional assets and market regimes.
- Conduct rolling or walk-forward validation rather than a single fixed train/test split.
- Perform forward paper-trading validation before any real-capital deployment.
- Investigate whether an adaptive ML threshold based on volatility or market regime can preserve the observed risk reduction without excessive loss of trading opportunity.

X. CONCLUSION

This study presents an ML-assisted validation framework for MACD trading signals and evaluates the framework across five timeframes and five confidence thresholds. The updated experimental evidence shows a consistent pattern: enabling the ML validation layer substantially reduces the number of accepted trades, modestly increases average win rate and reduces average maximum drawdown relative to the corresponding source-defined non-ML test configurations. The newly exposed net-realized values show positive ML averages at 1m, 5m and 15m, while 3m and 1h remain negative.

The results do not support treating ML as a universal profitability-enhancement mechanism. Instead, the evidence supports a more specific conclusion: within the tested MACD framework, ML functions as a signal-selectivity layer that materially changes trading activity and risk characteristics. The updated net-realized results show that the effect is timeframe-dependent: average net realized is positive at 1m, 5m and 15m, but negative at 3m and 1h. These are descriptive averages across separate configurations rather than a single trading-account outcome.

The contribution of the study is therefore methodological and empirical. It demonstrates how a deterministic technical-analysis strategy can be combined with machine-learning validation and evaluated through controlled, multi-timeframe experiments. Further statistical validation, complete

experiment documentation, realistic transaction-cost modelling and forward testing are required before drawing conclusions about live-market performance.

REFERENCES

- [1] Appel, G. (1979). The Moving Average Convergence-Divergence Trading Method: Advanced Version. Scientific Investment Systems Incorporated.
- [2] Li, C., Yu, H. T., & Chong, T. T. L. (2008). Structural change in the stock market efficiency after the millennium: The MACD approach. *Economics Bulletin*, 7(12).
- [3] Hung, N. H. (2016). Various moving average convergence divergence trading strategies: A comparison. *Investment Management and Financial Innovations*, 13(2-2), 363–369. [https://doi.org/10.21511/imfi.13\(2-2\).2016.11](https://doi.org/10.21511/imfi.13(2-2).2016.11)
- [4] Kang, B.-K. (2021). Improving MACD technical analysis by optimizing parameters and modifying trading rules: Evidence from the Japanese Nikkei 225 futures market. *Journal of Risk and Financial Management*, 14(1), 37. <https://doi.org/10.3390/jrfm14010037>
- [5] Agudelo Aguirre, A. A., Duque Méndez, N. D., & Rojas Medina, R. A. (2021). Artificial intelligence applied to investment in variable income through the MACD (moving average convergence/divergence) indicator. *Journal of Economics, Finance and Administrative Science*.
- [6] Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, 55(4), 1705–1765.
- [7] Sullivan, R., Timmermann, A., & White, H. (1999). Data-snooping, technical trading rule performance, and the bootstrap. *The Journal of Finance*, 54(5), 1647–1691.
- [8] Bailey, D. H., Borwein, J. M., López de Prado, M., & Zhu, Q. J. (2015). The probability of backtest overfitting. *Journal of Computational Finance*.
- [9] Kelly, B. T., & Xiu, D. (2023). Financial machine learning. *Foundations and Trends in Finance*, 13(3–4), 205–363.
- [10] Nazareth, N., & Reddy, Y. V. R. (2023). Financial applications of machine learning: A literature

review. Expert Systems with Applications, 219,
119640.