International Journal of Web of Multidisciplinary Studies
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: *https://ijwos.com*
Volume 03, Issue 11,November 2026
OPEN ACCESS
E-ISSN : 3049-2424
DOI: https://doi.org/10.71366/ijwos

CRYPTO CURRENCY ALPHA GENERATOR USING MACHINE LEARNING

Aditya Pawar[1], Anish Zuting[2], Hardik Jain[3], Arjun Shedage[4], Mrs. Ashvini Shirke[5]
[1,2,3,4,5] *Department of Artificial Intelligence and Data Science, DR D. Y. PATIL COLLEGE OF ENGINEERING AND INNOVATION, VARALE , TALEGAON*

Article Info

Article History:
Published: 07 Nov 2026

Publication Issue:
Volume 03, Issue 11
November 2026

Page Number:
113-120

Corresponding Author:
Aditya Pawar

Abstract:
Cryptocurrency spot markets operate continuously and exhibit noisy price movements, volatility shifts, and transaction frictions that can weaken rules-based strategies. This paper presents an empirical study of a research platform that combines two technical strategies—Z-score mean reversion and Donchian-channel trend following—with a machine-learning meta-filter. Instead of predicting every future price directly, the system scores candidate entries using a walk-forward Logistic Regression classifier trained on technical and statistical features. A K-Means model classifies volatility regimes and blocks entries in the high-volatility cluster, while a Random Forest regressor estimates next-bar volatility for position sizing. The evaluation uses historical Binance spot OHLCV data for BTC/USDT, ETH/USDT, SOL/USDT, BNB/USDT, and XRP/USDT over six timeframes from 5 minutes to one day, with next-bar-open execution and a fixed modeled round-trip transaction drag of 0.30%. According to the supplied project results, the meta-filter improves several configurations substantially: SOL/USDT 15-minute mean reversion moves from a baseline Sharpe ratio of −1.13 to 1.00, and XRP/USDT 4-hour trend following moves from 0.81 to 0.98. The best reported configuration is SOL/USDT 15-minute mean reversion, with Sharpe 1.00 across 253 trades and a 69.17% win rate. These results are historical backtest findings, not evidence of guaranteed live profitability. The study also identifies limitations arising from fixed transaction-cost assumptions, testing many strategy–asset–timeframe combinations, low trade counts on daily bars, and the absence of automated live-stream retraining. The contribution is an integrated, cost-aware evaluation of entry filtering, volatility-regime gating, position sizing, and risk controls rather than a claim that any single model independently predicts cryptocurrency prices.

Keywords: Cryptocurrency trading; meta-labeling; Logistic Regression; mean reversion; trend following; volatility regimes; transaction costs; walk-forward validation

1 Introduction

Cryptocurrency trading presents a difficult setting for systematic decision-making because markets operate 24 hours a day and price series can contain rapid changes, noisy movements, and periods with different volatility characteristics. A signal that appears useful in one interval may become ineffective when transaction costs, execution timing, or a change in market conditions are considered. The supplied project therefore frames the problem as a complete trading-system evaluation problem: generate candidate entries, decide which entries to accept, control risk, model trading frictions, and evaluate the resulting equity curve.

Prior research has explored several ways of approaching cryptocurrency prediction and trading. Omole and Enke compare deep-learning models for Bitcoin direction prediction and assess trading strategies through backtesting, including the effect of feature selection [1]. John, Binnewies, and Stantic survey cryptocurrency price-prediction algorithms and discuss the range of technical, sentiment, and other factors examined in the literature [2]. Fu et al. investigate a machine-learning pair-trading approach with an enhanced triple-barrier labelifng method, illustrating the

Copyright © 2026 The Author(s) : This is an open access article under the CC BY license (http://creativecommons.org/licenses/by-nc/4.0/)

importance of how trading opportunities are labeled [3]. Otabek and Choi study multi-level deep Q-networks for Bitcoin trading strategies, representing a reinforcement-learning approach to automated trading [4]. These studies differ in data, model families, and trading objectives; they are related research context rather than direct replications of the system described here.

This study adopts a different design from direct price-direction prediction and reinforcement-learning policy optimization. Two deterministic technical strategies first generate candidate long entries. A Logistic Regression meta-classifier then estimates whether a candidate entry is likely to meet a net-return threshold after modeled fee and slippage costs. The architecture also includes K-Means volatility-regime detection, a Random Forest volatility estimator used for position sizing, and explicit portfolio-level and trade-level risk controls.

The paper makes three bounded contributions. First, it documents an integrated, reproducible architecture for combining mean-reversion and trend-following signals with a trade-level classifier. Second, it reports the supplied project's historical backtest metrics across five spot pairs and six timeframes, emphasizing the differences between unfiltered and ML-filtered configurations. Third, it discusses the limits of the reported evidence, including fixed cost modeling and selection bias, instead of interpreting the highest backtest Sharpe ratio as a guarantee of future performance.

2 Related Work

Omole and Enke evaluate CNN–LSTM, LSTNet, TCN, and ARIMA-based Bitcoin direction prediction, alongside feature-selection approaches and strategy backtests [1]. Their work is relevant because it connects predictive modeling to the performance of a trading rule rather than treating classification accuracy as the only endpoint. The present platform shares the use of historical market data, technical features, and backtesting, but differs by using technical strategies to propose entries and applying Logistic Regression as a second-stage filter. It does not implement the deep-learning models or on-chain feature set studied in [1].

The survey by John et al. organizes cryptocurrency prediction research and describes the variety of model families and feature categories used in the field [2]. It provides broad context for the problem of selecting an appropriate modeling design in a market affected by multiple types of information. The current work is narrower: it evaluates a specific rules-plus-meta-classifier system using OHLCV-derived features. It does not incorporate sentiment or legal-event variables, and it does not claim to be a comprehensive comparison of cryptocurrency forecasting algorithms.

Fu et al. study pair trading with machine learning and an enhanced triple-barrier labeling approach [3]. Their focus on labeling is pertinent to this work because the quality of a learning target determines what the classifier is trained to recognize. The supplied project uses a simpler binary meta-label: for candidate strategy entries, the forward return over four bars is labeled positive when it exceeds the modeled 0.30% round-trip cost threshold. This is not the triple-barrier method and is not a pair-trading strategy; the connection is methodological, not an assertion of equivalence.

Otabek and Choi apply multi-level deep Q-networks to Bitcoin trading [4]. Their work represents a policy-learning approach in which the agent learns trading actions, whereas the current system combines fixed technical entry/exit rules with supervised trade filtering and deterministic risk controls. The two approaches share the objective of evaluating automated trading decisions but use different learning paradigms. No reinforcement-learning component is included in the supplied project.

Taken together, the four papers motivate examining model choice, trade labels, and the relationship between prediction and simulated trading outcomes. They do not establish that the current architecture is universally superior. The present experiment should instead be interpreted as a project-specific, multi-asset backtest with explicit limitations.

3 Research Problem and Objectives

The research problem is whether a supervised meta-filter, combined with regime and risk controls, can improve the historical risk-adjusted performance of two baseline technical strategies across crypto spot assets and timeframes when transaction costs are included.

The objectives are: (i) implement mean-reversion and trend-following entry logic; (ii) train a Logistic Regression model to filter candidate entries using walk-forward evaluation; (iii) suppress entries in a high-volatility regime identified by K-Means; (iv) use predicted volatility for inverse-volatility position sizing; and (v) compare baseline and ML-filtered backtest metrics while explicitly modeling fees and slippage.

The central research question is: under the supplied historical data and cost assumptions, how does ML meta-filtering change strategy performance across assets and timeframes, and what risks remain before live deployment? The analysis is descriptive and empirical. It does not claim causal proof that the classifier alone explains every observed change, because the full ML-filtered path also includes regime gating and the project's backtest configuration.

4 Data and Feature Engineering

The project obtains historical OHLCV (open, high, low, close, and volume) data from Binance public REST endpoints through CCXT. The configured universe contains BTC/USDT, ETH/USDT, SOL/USDT, BNB/USDT, and XRP/USDT. The configured timeframes are 5 minutes, 15 minutes, 30 minutes, 1 hour, 4 hours, and 1 day. The project reference describes the historical range as beginning on 1 January 2018 and extending to the current UTC date used by the data configuration.

Feature engineering uses only historical lookbacks. Features include log return, 24-bar simple moving average, 24-bar rolling standard deviation, rolling price Z-score, RSI-14, 24-bar return volatility, volume Z-score, UTC hour, a prior 20-bar Donchian upper channel, a prior 10-bar Donchian lower channel, and a 14-bar average high–low range proxy. The Donchian limits are shifted by one bar to avoid using the current bar's high or low to define the prior channel boundary. The supplied test suite includes a feature-level check for look-ahead behavior.

For the meta-model, the documented feature set is [Z-score, RSI-14, 24-bar volatility, volume Z-score, hour, and log return]. Candidate training rows are restricted to bars where the baseline strategy generated an entry signal. The binary label uses a four-bar forward-return horizon: a candidate is positive if its forward return exceeds 0.30%, the project's modeled round-trip fee-plus-slippage threshold; otherwise it is negative. The label is intended to estimate the quality of a proposed entry, not to forecast every bar's exact price.

5 Methodology

5.1 Baseline strategies

Mean reversion generates a long entry when the rolling price Z-score falls below −2.0 and exits when the Z-score rises above 0.0. The state machine uses a discrete position state and does not pyramid positions. Trend following enters long when the close breaks above the prior 20-bar Donchian high and exits when the close falls below the prior 10-bar Donchian low. Channel boundaries use historical values shifted by one bar.

5.2 Logistic Regression meta-filter

The supervised model pipeline consists of StandardScaler followed by Logistic Regression with balanced class weights and a maximum of 1,000 iterations. Walk-forward evaluation uses TimeSeriesSplit, with the universe pipeline documented as using three splits (the training utility also supports five). Timestamp-overlap checks reject train/test splits that share timestamps. When the available meta-labeled dataset contains fewer than 50 entries, training is skipped and probabilities default to 1.0, as specified in the project. Candidate entries pass the model gate only when the predicted positive-class probability is greater than 0.55.

5.3 Volatility-regime gate and volatility model

K-Means with three clusters groups observations using return, 24-bar volatility, and volume Z-score. The clusters are ordered by mean 24-bar volatility and named low-, medium-, and high-volatility regimes. The system blocks strategy entries in the high-volatility regime. Separately, a Random Forest regressor with 200 trees predicts next-bar 24-bar return volatility from return, RSI-14, volume Z-score, and hour. The predicted volatility is used for inverse-volatility position sizing; it is not used to generate the entry signal. The three regimes are illustrated in Fig. 1.

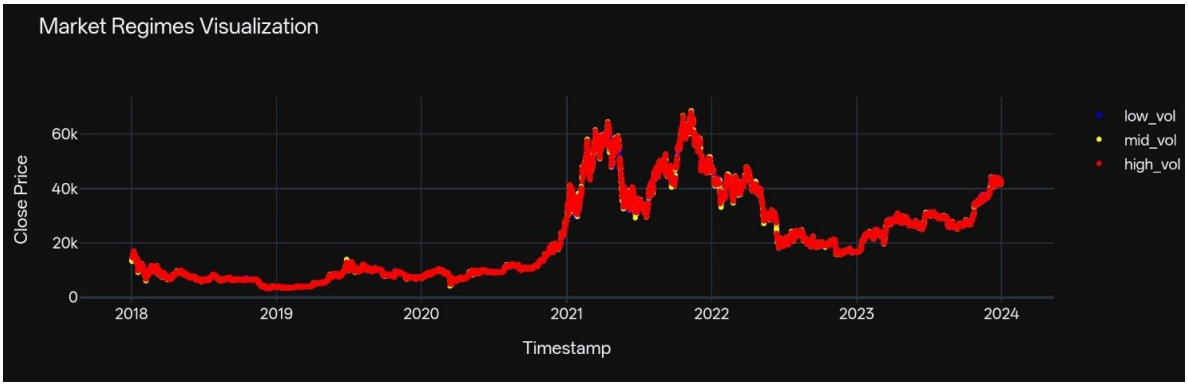


Fig. 1. Market-regime visualization showing Bitcoin closing prices classified into low-, medium-, and high-volatility regimes (2018–2024).

5.4 Execution, transaction costs, and risk controls

The backtester processes price series in a single event-driven pass. A signal from bar i−1 is executed at the open of bar i, avoiding same-bar signal execution. The modeled fee is 0.10% per leg and slippage is 0.05% per leg, giving a nominal round-trip drag of 0.30%. The risk manager caps position allocation at 5% by default, targets 1% volatility for sizing, stops an individual long trade at a 3% loss, and halts trading when portfolio drawdown reaches 20%. These values describe the project configuration, not universal recommendations.

5.5 Evaluation metrics

The project calculates total return, CAGR, annualized Sharpe and Sortino ratios, Calmar ratio, maximum drawdown, drawdown duration, win rate, profit factor, payoff ratio, average win/loss, tail ratio, time in market, trade count, and average trade PnL. Sharpe, Sortino, and Calmar are set to NaN when fewer than 30 completed trades are available. This low-sample guard is important when interpreting daily-timeframe results.

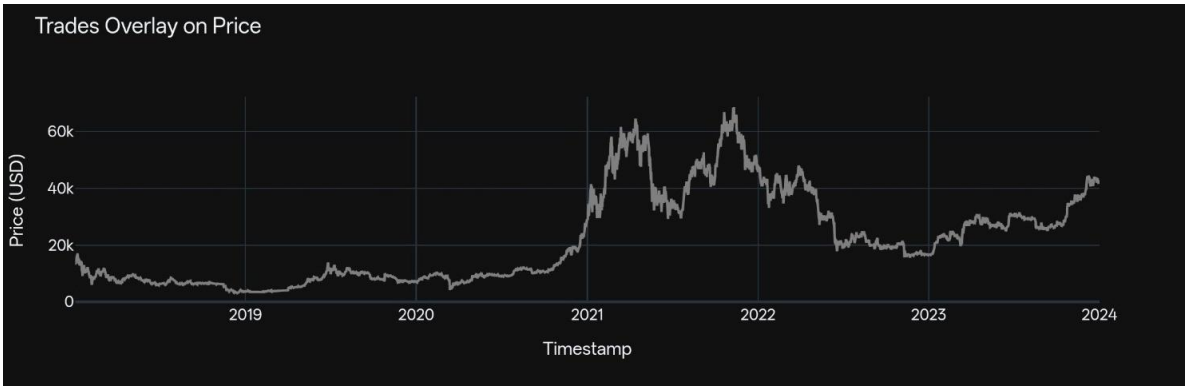


Fig. 2. Historical price-series visualization supplied for the trade-overlay analysis (2018–2024). Individual entry and exit markers are not discernible in the supplied rendering.

6 Experimental Results

The figures below are transcribed from the supplied project reference, which states that the metrics were extracted from the project's universe_results.parquet report. The tables summarize selected configurations rather than reproducing every asset–timeframe cell. All values are historical backtest outputs under the configured 0.30% round-trip transaction-drag assumption. The supplied visualizations are presented in Figs. 2–4.

6.1 Selected Sharpe-ratio changes

Strategy / Pair / Timeframe	Baseline Sharpe	ML-filtered Sharpe	Change
Mean reversion / SOL-USDT / 15m	−1.13	1.00	+2.13

Mean reversion / BTC-USDT / 15m	−3.05	0.54	+3.59
Mean reversion / BNB-USDT / 15m	−2.07	0.58	+2.65
Mean reversion / SOL-USDT / 30m	−0.56	0.95	+1.51
Trend following / XRP-USDT / 4h	0.81	0.98	+0.17
Trend following / ETH-USDT / 4h	0.52	0.91	+0.39
Trend following / BTC-USDT / 1h	−0.32	0.52	+0.84

Top 15 Strategy Configurations by Sharpe Ratio — ML-Filtered

Strategy	Symbol	Timeframe	Sharpe	Sortino	Calmar	Max Drawdown	Win Rate	Payoff Ratio	Time in Market	Num Trades
Mean Reversion	SOL/USDT	15m	1.00	1.48	0.90	-35.4%	69.2%	0.70	2.1%	253
Mean Reversion	SOL/USDT	30m	0.95	1.41	0.53	-53.3%	72.2%	0.69	2.7%	158
Mean Reversion	BNB/USDT	15m	0.58	0.83	0.30	-47.6%	64.1%	0.65	2.6%	429
Mean Reversion	XRP/USDT	15m	0.58	0.80	0.32	-55.5%	63.5%	0.64	4.9%	778
Mean Reversion	BTC/USDT	15m	0.54	0.81	0.42	-25.9%	64.0%	0.69	3.0%	511
Mean Reversion	XRP/USDT	4h	0.47	0.72	0.19	-61.0%	67.5%	0.81	7.4%	77
Mean Reversion	ETH/USDT	30m	0.44	0.62	0.20	-47.3%	63.9%	0.65	3.8%	332
Mean Reversion	BNB/USDT	30m	0.40	0.56	0.17	-49.8%	69.5%	0.53	3.1%	266
Mean Reversion	BTC/USDT	30m	0.38	0.55	0.17	-38.7%	65.6%	0.63	3.0%	256
Mean Reversion	BTC/USDT	1h	0.36	0.50	0.17	-33.3%	67.4%	0.58	4.7%	218
Mean Reversion	BNB/USDT	1h	0.31	0.42	0.11	-41.9%	61.8%	0.81	1.6%	68
Mean Reversion	SOL/USDT	4h	0.21	0.29	0.02	-65.4%	68.8%	0.49	4.4%	82
Mean Reversion	XRP/USDT	30m	0.20	0.28	-0.02	-72.6%	64.0%	0.55	5.5%	430
Mean Reversion	BNB/USDT	4h	0.14	0.20	0.00	-55.4%	65.6%	0.53	3.4%	82
Mean Reversion	SOL/USDT	1h	0.14	0.20	-0.02	-59.7%	57.6%	0.71	4.1%	99
Trend Following	XRP/USDT	4h	0.98	1.60	0.84	-51.0%	43.1%	2.69	10.0%	51
Trend Following	ETH/USDT	4h	0.91	1.47	0.65	-42.5%	39.7%	2.29	19.5%	121
Trend Following	ETH/USDT	1h	0.82	1.23	0.50	-41.3%	38.1%	1.91	14.2%	370
Trend Following	XRP/USDT	1h	0.77	1.19	0.38	-68.9%	38.7%	2.44	5.2%	111
Trend Following	BTC/USDT	4h	0.63	0.92	0.30	-40.6%	39.1%	2.13	15.9%	92
Trend Following	BNB/USDT	4h	0.54	0.84	0.20	-64.3%	41.0%	1.91	10.0%	61
Trend Following	BTC/USDT	1h	0.52	0.76	0.30	-27.6%	41.5%	1.74	8.6%	207
Trend Following	XRP/USDT	30m	0.51	0.76	0.20	-68.5%	38.1%	1.92	6.9%	320
Trend Following	SOL/USDT	4h	0.50	0.76	0.26	-48.6%	50.0%	1.35	14.8%	74
Trend Following	BNB/USDT	30m	0.25	0.36	0.05	-65.0%	34.4%	1.97	6.2%	337
Trend Following	BNB/USDT	1h	0.22	0.33	0.06	-42.7%	32.5%	2.21	6.7%	169
Trend Following	SOL/USDT	1h	0.08	0.12	-0.03	-68.7%	40.0%	1.41	8.7%	160
Trend Following	ETH/USDT	30m	0.03	0.04	-0.05	-67.1%	37.4%	1.61	12.7%	644
Trend Following	BNB/USDT	15m	-0.30	-0.43	-0.14	-77.5%	34.9%	1.64	4.2%	470
Trend Following	XRP/USDT	15m	-0.35	-0.53	-0.24	-95.4%	32.9%	1.69	6.7%	752

Fig. 3. Expanded ML-filtered strategy-configuration metrics, including risk-adjusted returns, drawdown, win rate, payoff ratio, market exposure, and trade count.

The selected rows show that the change is not uniform across every configuration. The largest highlighted improvements occur in sub-hourly mean-reversion examples, where the baseline strategy is strongly negative and the ML-filtered result becomes positive. Trend-following improvements are also visible at some hourly and four-hour intervals, while several other ML-filtered cells remain negative. The results therefore support a conditional finding: filtering can improve selected strategy–asset–timeframe combinations, but it does not make every configuration profitable.

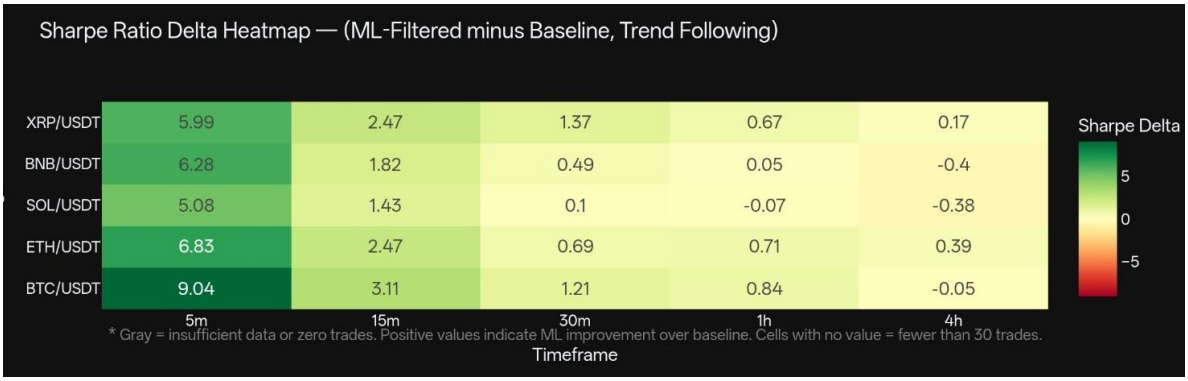


Fig. 4. Sharpe-ratio delta heatmap for trend following (ML-filtered minus baseline) across five cryptocurrency pairs and five intraday-to-multi-hour timeframes.

6.2 Top reported ML-filtered configurations

Rank	Strategy	Pair	Timeframe	Sharpe	Trades	Win rate	Payoff ratio
1	Mean reversion	SOL/USDT	15m	1.00	253	69.17%	0.70
2	Trend following	XRP/USDT	4h	0.98	51	43.14%	2.69
3	Mean reversion	SOL/USDT	30m	0.95	158	72.15%	0.69
4	Trend following	ETH/USDT	4h	0.91	121	39.67%	2.29
5	Trend following	ETH/USDT	1h	0.82	370	38.11%	1.91

The top rows illustrate different trade profiles. The mean-reversion examples have higher win rates but payoff ratios below one, whereas the listed trend-following examples have lower win rates and higher payoff ratios. This is consistent with different return distributions across strategy types; the table alone does not establish that these profiles will persist out of sample. The expanded configuration-level metrics are shown in Fig. 3.

6.3 Timeframe and cost effects

The project reports heavily negative baseline Sharpe ratios on the 5-minute timeframe for both strategy families. For mean reversion, baseline Sharpe values range from −5.69 to −10.08 across the five pairs; for trend following, the range is −5.85 to −13.07. The project attributes the poor high-frequency baseline behavior in part to frequent trading under a fixed 0.30% round-trip cost model. At the 15- and 30-minute intervals, several mean-reversion configurations improve after filtering. Trend-following results are strongest among the reported configurations at 1-hour and 4-hour intervals.

Daily-timeframe ML results are not ranked because all daily ML configurations have fewer than 30 trades and the metric guard assigns NaN to Sharpe, Sortino, and Calmar. This is a reporting safeguard, not evidence that daily trading is necessarily ineffective. It means the available sample does not support those particular risk-adjusted comparisons. The trend-following Sharpe-ratio changes across pairs and timeframes are visualized in Fig. 4.

7 Discussion

The results are compatible with the design rationale that a trade-level filter may be useful when candidate signals contain many weak entries. The classifier does not replace the technical strategies: it conditions their entry decisions

on a probability score trained against a net-return label. This differs from the direct next-day direction prediction setting in [1] and the policy-learning setting in [4]. The pair-trading and triple-barrier labeling study [3] is relevant to the general importance of labeling, but the current system uses a fixed-horizon binary threshold rather than a triple-barrier label. The survey in [2] situates this design within a broad range of cryptocurrency prediction approaches, but the present results do not compare against every approach covered in that survey.

A practical implication is that performance should be examined at the strategy–asset–timeframe level rather than averaged into a single claim. Mean reversion and trend following have different entry logic and payoff profiles, and transaction drag affects high-frequency strategies disproportionately when gross edge per trade is small. A classifier that reduces trade frequency can change both the win rate and the distribution of gains and losses. Consequently, Sharpe ratio, drawdown, trade count, win rate, and payoff ratio should be read together.

The system also layers controls rather than relying on a single model: the high-volatility regime gate suppresses some entries, predicted volatility scales position size, a per-trade stop-loss limits individual losses, and a portfolio drawdown kill switch halts trading after a threshold. These controls are part of the implemented platform. The summary results do not isolate the marginal contribution of each control, so attribution to the Logistic Regression model alone would be too strong.

The strongest reported Sharpe ratio is 1.00, not an extraordinary value, and it arises from a selected configuration among 60 strategy–pair–timeframe combinations. The project itself identifies selection bias and multiple testing as limitations. A ranking selected after testing many combinations can be optimistic; the highest observed value should therefore be treated as a candidate for further validation rather than as a final estimate of expected future performance.

8 Limitations and Future Work

First, transaction costs are modeled as constants: a 0.10% fee and 0.05% slippage per leg. Actual slippage can vary with order size, order-book depth, volatility, and execution conditions. This limitation matters especially for short timeframes, where small gross returns can be dominated by friction.

Second, 60 strategy–pair–timeframe combinations were tested. The best configurations are exposed to selection bias and multiple testing. They should be validated using a genuinely untouched out-of-sample period and forward paper trading before any claim of generalization.

Third, daily intervals have fewer than 30 trades in all ML configurations, making the corresponding Sharpe, Sortino, and Calmar metrics unavailable under the project's low-sample guard. More data or a different evaluation design would be required before drawing strong daily-timeframe conclusions.

Fourth, walk-forward retraining is implemented for historical evaluation but is not automated for live WebSocket streaming. A production-oriented extension would need scheduled or event-driven retraining, monitoring for data drift, operational failure handling, and an evaluation of live fills against the backtest assumptions. Such work is proposed as future work; it has not been completed or evaluated in the supplied project.

Fifth, the current classifier uses a fixed four-bar forward-return label and a 0.30% threshold. Future research could compare alternative labeling horizons and barrier-based labels, inspired by the methodological question examined in [3], without assuming that a different labeling method will necessarily improve results. The model family could also be compared with the predictive models discussed in [1] or the reinforcement-learning direction represented by [4], but these would be new experiments rather than findings of this study.

Finally, the reported study uses OHLCV-derived features and does not include news sentiment or on-chain variables. The literature represented by [1] and [2] discusses other feature sources and prediction approaches. Adding such inputs would require separate data collection, careful timestamp alignment, leakage checks, and a new out-of-sample evaluation. The current paper does not make claims about the performance of these unimplemented extensions.

9 Conclusion

This paper documented a cost-aware cryptocurrency spot trading research platform that combines Z-score mean reversion and Donchian-channel trend following with Logistic Regression meta-labeling, K-Means volatility-regime gating, Random Forest volatility prediction, and explicit risk controls. The evaluation covers five Binance spot pairs and six timeframes, using next-bar-open execution and a fixed 0.30% modeled round-trip transaction drag.

In the supplied historical results, ML filtering materially improves selected configurations. SOL/USDT 15-minute mean reversion improves from a baseline Sharpe ratio of −1.13 to 1.00, while XRP/USDT 4-hour trend following improves from 0.81 to 0.98. The highest-ranked configuration reports 253 trades and a 69.17% win rate. At the same time, many filtered configurations remain negative, and the daily timeframe lacks enough trades to report risk-adjusted metrics under the project's threshold.

The main conclusion is therefore limited but useful: meta-filtering can improve historical results for selected strategy–asset–timeframe combinations in this implementation, but the results remain sensitive to transaction-cost assumptions, configuration selection, sample size, and the gap between offline backtesting and live execution. More realistic variable-cost modeling and untouched out-of-sample forward testing are necessary before assessing operational robustness.

Declarations

Funding: No funding information was provided in the supplied project materials.

Competing interests: No competing-interest statement was provided; the author should confirm and edit this declaration before submission.

Data availability: The project reference describes data acquisition from public Binance REST endpoints and local project reports. The exact report dataset should accompany any reproducibility package intended for publication.

Code availability: The supplied project reference documents a Python implementation and commands for fetching data, engineering features, running the backtest, generating reports, and running tests. Availability of a public repository was not specified.

Ethics and financial disclaimer: This work is a quantitative research and educational exercise. Historical backtest performance is not financial advice and does not guarantee future profitability.

References

[1] Omole, O., Enke, D.: Deep learning for Bitcoin price direction prediction: models and trading strategies empirically compared. Financial Innovation 10, Article 117 (2024). https://doi.org/10.1186/s40854-024-00643-1

[2] John, D.L., Binnewies, S., Stantic, B.: Cryptocurrency Price Prediction Algorithms: A Survey and Future Directions. Forecasting 6(3), 637–671 (2024). https://doi.org/10.3390/forecast6030034

[3] Fu, N., Kang, M., Hong, J., Kim, S.: Enhanced Genetic-Algorithm-Driven Triple Barrier Labeling Method and Machine Learning Approach for Pair Trading Strategy in Cryptocurrency Markets. Mathematics 12(5), 780 (2024). https://doi.org/10.3390/math12050780

[4] Otabek, S., Choi, J.: Multi-level deep Q-networks for Bitcoin trading strategies. Scientific Reports 14, Article 771 (2024). https://doi.org/10.1038/s41598-024-51408-w