Determining Multi-Class Trading Signals for Bitcoin: A Comparative Study of XGBoost, LightGBM, and Random Forest

Marcin Stawarz
Nicolaus Copernicus University in Torun, Faculty of Economic Sciences and Management,
Torun, Poland *marcin.stawarz@umk.pl*

Michał Dominik Stasiak
Poznan University of Economics and Business, Department of Investment and Real Estate,
Poznan, Poland *michal.stasiak@ue.poznan.pl*

Abstract

We investigate a multi-class machine learning (ML) framework to generate daily Bitcoin trading signals—Buy, Sell, or Hold. Three algorithms—XGBoost, LightGBM, and Random Forest—are compared with a naive buy-and-hold strategy. Using BTC/USD daily data (2015–2024), we apply a range of technical indicators across trend, momentum, volatility, and volume, later pruned by correlation analysis. A ±1% threshold defines the "Hold" zone to avoid minor fluctuations. Empirical tests show that LightGBM outperforms other models and even surpasses buy-and-hold in final portfolio value. Our findings support the design of tri-class ML strategies tailored for high-volatility markets like cryptocurrency.

Keywords: Bitcoin, trading signals, XGBoost, LightGBM, Random Forest.

1. Introduction

Bitcoin's persistent volatility, lack of intrinsic valuation, and 24/7 trading dynamics expose the limitations of traditional forecasting models. Machine learning (ML) offers new tools to address such challenges [1,2], but naive binary strategies often suffer from overtrading, excessive drawdowns, and weak generalization.

Recent literature suggests that adding a third "Hold" class can enhance model stability by reducing the frequency of low-confidence trades [3]. We adopt a tri-class framework, labeling days as Buy, Sell, or Hold based on a ±1% return threshold. While static, this threshold reflects typical daily volatility in BTC markets and enables a clean experimental design. Future iterations should explore volatility-adaptive alternatives.

We benchmark three ensemble classifiers—Random Forest [4], XGBoost [5], and LightGBM [8]—on real daily BTC/USD data (2015–2024). These models represent distinct ensemble paradigms (bagging vs. boosting), allowing us to test how learning architecture interacts with the proposed label structure and feature set.

Advanced ML has shown promise across asset pricing [7], volatility modeling [6], and execution systems [9]. Our aim is to bridge academic insight with real-world implementation by assessing model performance not only in accuracy, but also in practical investment terms. Deep learning methods such as CNNs, LSTMs, and transformers have also shown effectiveness in capturing complex, multi-horizon dynamics in financial time series [10,11].

2. Data and Methodology

2.1. Data Sources and Indicator Selection

We source daily Bitcoin price and volume data from Stooq.pl, covering January 1, 2015, through June 30, 2024. The dataset includes open, high, low, close, and adjusted close prices, as well as trading volume. To explore diverse predictors, we calculate numerous technical indicators spanning four principal categories: Trend (e.g., moving averages, ADX, Aroon Oscillator), Momentum (e.g., RSI, stochastic oscillators, MACD, ROC), Volatility (e.g., ATR, Bollinger metrics), Volume (e.g., OBV, CMF, PVT).

Observing many correlation coefficients above 0.9, we eliminate strongly overlapping indicators to avoid duplication, ultimately retaining EMA_20 (trend), RSI, Stoch_K, MACD, ROC (momentum), ATR (volatility), and OBV, CMF (volume).

2.2. Multi-Class Target Definition and Lagging

$$\mathrm{r}_{\mathrm{t+1}} = \frac{C_{t+1}-C_t}{C_t} \tag{1}$$

where C_t is the closing price on day t. If r_{t+1} exceeds +1%, day *t* is labeled Buy; if it falls below –1%, day *t* is Sell; otherwise, day *t* is Hold. All indicators are lagged by one trading day to ensure they reflect only the information available up to day t.

We chose ±1% to reflect typical daily volatility in Bitcoin. Many days see fluctuations over 1%, so restricting signals to moves beyond ±1% helps sidestep minor oscillations.

Importantly, the target labels are not predicted from price alone, but from a rich set of lagged technical indicators. These include features capturing trend, momentum, volatility, and volume. By incorporating such indicators, the models can detect complex, nonlinear relationships and anticipate meaningful market movements more effectively than raw returns alone.

2.3. Machine Learning Models and Training

We compare three ensemble classifiers:

- Random Forest (RF) – bagging of decision trees,
- XGBoost – gradient boosting with regularization,
- LightGBM – boosting with histogram-based, leaf-wise splits.

Models were implemented in Python 3.9 using scikit-learn, XGBoost, and LightGBM libraries. Each model was trained and tuned on historical BTC/USD data from 2015–2022 via grid search and validation splits. We used an expanding-window forecasting scheme from January 2023 to June 2024, retraining daily on prior data to avoid look-ahead bias.

Models output one of three signals—Buy, Sell, or Hold—based on predicted daily return exceeding ±1%. These signals were used to generate classification metrics and were subsequently fed into a portfolio backtest for final evaluation.

2.4. Backtesting Strategy

We simulate daily rebalancing with $10,000 initial capital and a 0.1% transaction cost. Trade logic is:

- Buy: If no position, open a long; if short, close it and go long; if already long, do nothing.
- Sell: If no position, open a short; if long, close it and go short; if already short, do nothing.
- Hold: Keep the previous day's position (long, short, or none).

Portfolio value is updated at each close. We compute performance metrics including cumulative return, drawdown, Sharpe, Sortino, and compare each model's results against a naive buy-and-hold baseline.

3. Results and Discussion

Table 1 presents key backtest metrics for each strategy from January 2023 to June 2024. LightGBM achieved the highest final portfolio value (46,605 USD), the strongest Sharpe (2.64) and Sortino (3.86) ratios, and the lowest drawdown (–19%), using only 20 trades.

Buy-and-hold, while solid, fell behind with slightly higher volatility. Random Forest yielded modest gains but overtraded (50 trades), while XGBoost underperformed with poor timing and a 46% loss.

Table 1. Backtested performance metrics for each strategy.

Metric	XGBoost	Random Forest	LightGBM	Buy-and-Hold
Final Portfolio Value	$5422	$15098	$46605	$36387
Cumulative Return	-46%	+51%	+366%	264%
Sharpe Ratio	-0.70	0.85	2.64	2.20
Sortino Ratio	-0.81	1.21	3.86	2.79
Max Drawdown	-64%	-36%	-19%	-21%
Transactions	24	50	20	0

3.1. Key Observations

XGBoost concluded the test with the weakest outcome—final portfolio value of 5,422 USD (–46%), driven by poor signal timing around key market turns and persistent drawdowns. Random Forest performed better, reaching 15,098 USD (+51%), but its frequent trades (50 in total) inflated transaction costs, limiting net profitability despite partially capturing upward trends. LightGBM achieved the best result: highest final value (46,605 USD), strongest risk-adjusted metrics (Sharpe 2.64, Sortino 3.86), and the lowest drawdown (–19%). It outperformed even the buy-and-hold benchmark (36,387 USD) by capturing sustained bullish phases while avoiding volatile corrections.

Buy-and-hold remained effective in a generally rising market but was ultimately surpassed by the more selective LightGBM model.

3.2. Equity Curves

Figure 1 compares the equity curves for the four strategies. LightGBM (green line) shows a steady climb, especially capturing major surges from mid-2023 onward, while buy-and-hold (red line) trails slightly by the end despite also benefiting from the general Bitcoin uptrend. Meanwhile, Random Forest (orange line) struggles with overfitting to minor fluctuations, and XGBoost (blue line) remains deeply underwater.

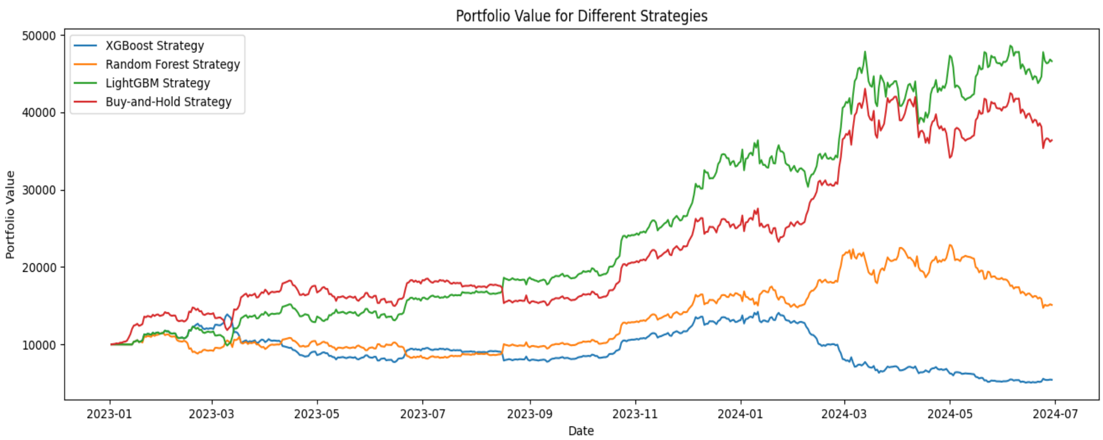

Fig. 1. Final equity curves for XGBoost, Random Forest, and LightGBM from January 2023 to July 2024.

4. Conclusion and Outlook

This study presented a multi-class machine learning framework for daily Bitcoin trading, applying Random Forest, XGBoost, and LightGBM to generate Buy, Sell, or Hold signals using a ±1% threshold. The approach included lagged indicators and correlation-based feature selection to ensure robust input data.

The results confirm that such a setup can improve both returns and risk control. LightGBM outperformed all models, delivering the highest cumulative return (+366%), best risk-adjusted ratios (Sharpe 2.64, Sortino 3.86), and the lowest drawdown (–19%) with relatively few trades. Random Forest showed moderate effectiveness but suffered from high turnover, while XGBoost performed poorly due to weak signal timing.

The Hold class helped reduce overtrading and protected against uncertainty—especially valuable in volatile markets. The tri-class structure proved more flexible than binary classification by allowing the model to remain inactive during ambiguous periods.

Despite these positive results, the approach has some limitations. The fixed ±1% threshold may not adapt well to changing volatility regimes. The dataset focuses only on Bitcoin at daily frequency, and the backtest does not account for market frictions such as slippage or liquidity constraints.

Future research should explore:

- Adaptive thresholds based on volatility measures (e.g., ATR),
- Multi-asset extensions and intraday data,
- Integration of risk management techniques like stop-losses and position sizing,
- Real-world simulation with capital and execution constraints.

In summary, multi-class ML strategies—when carefully designed and tuned—can outperform both binary models and passive investing, offering a promising path for systematic trading in maturing cryptocurrency markets.

Acknowledgements

This work was supported by the National Science Centre under Grant 2019/35/B/HS4/00642.

References

1. Abraham, J., Higdon, D., Nelson, J., Ibarra, J.: Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review* 1(3), 1–22 (2018)
2. Alessandretti, L., ElBahrawy, A., Aiello, L.M., Baronchelli, A.: Anticipating cryptocurrency prices using machine learning. *Complexity* 2018, 8983590 (2018)
3. Atsalakis, G.S., Valavanis, K.P.: Forecasting stock market short-term trends using a neuro-fuzzy methodology. *Expert Syst. Appl.* 36(7), 10696–10707 (2009)
4. Breiman, L.: Random Forests. *Mach. Learn.* 45, 5–32 (2001)
5. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pp. 785–794. ACM, New York (2016)
6. Fiszeder, P., Małecka, M.K., Molnár, P.: Robust estimation of the range-based GARCH model: forecasting volatility, value at risk and expected shortfall of cryptocurrencies. *Econ. Model.* 141, 106887 (2024)
7. Gu, S., Kelly, B., Xiu, D.: Autoencoder asset pricing models. *J. Econometrics* 218(1), 121–143 (2020)
8. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Liu, T.Y.: LightGBM: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* 30 (2017)
9. McNally, S., Roche, J., Caton, S.: Predicting the price of bitcoin using machine learning. In: *2018 Euromicro Conf. on Parallel, Distributed and Network-based Processing*, pp. 339–343. IEEE, Cambridge (2018)
10. Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning: A review: 2005–2019. *Appl. Soft Comput.* 90, 106181 (2020)
11. Wang, Y., Luo, H., Chen, Y.: Multi-horizon financial time series forecasting using transformer. *Expert Syst. Appl.* 208, 118192 (2022)