Bitcasting: A Proposal of Multi-Horizon Bitcoin Direction Prediction Using Machine Learning

Hugo Leonardo B. Fernandes
Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife-PE, Brazil
hlbf@cin.ufpe.br

Daniel C. Cunha
Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife-PE, Brazil
dcunha@cin.ufpe.br

***Abstract*—This paper investigates the efficacy of machine learning models in predicting Bitcoin's price direction across multiple time horizons. We train models to classify price movements as buy, sell, or hold signals using technical, fundamental, and time-series features. Our methodology integrates feature engineering, data from Binance and CoinMetrics, and a comparative evaluation of XGBoost, support vector machine, logistic regression, and random forest. The results show that time series and technical indicators are especially effective at shorter horizons. Among the models, XGBoost consistently achieved superior performance in both classification accuracy and trading profitability. Simulated trading strategies based on the model's predictions notably outperformed a buy-and-hold benchmark, reaching cumulative returns of 347% and 607% for 3-day and 30-day horizons, respectively, while maintaining competitive Sharpe ratios.**

***Index Terms*—Bitcoin, cryptocurrency, machine learning, forecasting, time series analysis.**

I. Introduction

The emergence of Bitcoin in 2008, introduced by the pseudonymous figure Satoshi Nakamoto, marked a significant milestone in the evolution of digital currencies [1]. Its fundamental concept revolves around a decentralized digital currency that leverages cryptographic principles to secure transactions and preserve value. This innovative approach facilitates peer-to-peer transactions, eliminating the need for traditional financial intermediaries, such as central banks.

Bitcoin is created through mining, where miners are rewarded for solving complex mathematical problems. The supply of Bitcoin is inherently limited; new issuances are expected to cease around 2140, capping the total circulation at 21 million coins [2].

Bitcoin's widespread appeal among global investors has been driven by several key attributes, leading to a market capitalization of $1.65 trillion by 2025 and solidifying its dominance in the cryptocurrency market [3]. However, this rapid growth has also been marked by significant volatility, characterized by sharp price fluctuations over short periods. This inherent volatility presents a considerable challenge for price prediction and creates an environment conducive to various analytical techniques.

Unlike traditional financial markets, cryptocurrency exchanges operate continuously, 24 hours a day, further complicating prediction efforts. As a result, there is considerable interest in using cryptocurrency forecasting tools, especially those based on machine learning (ML) algorithms, to generate potential profits, especially utilizing emerging technologies such as Transformers and hybrid approaches [4]. This interest has been noted by both academic researchers and market participants alike.

The efficient market hypothesis (EMH) asserts that all relevant information influencing asset prices is already incorporated into those prices. According to this theory, it is impossible to consistently achieve profits that exceed market averages [5]. While the EMH indicates limitations in predicting asset prices, the unique characteristics of cryptocurrency markets—such as their emerging nature and the fact that they operate on a continuous trading basis—provide a compelling context to examine the relevance of this hypothesis and to explore the potential of advanced forecasting methods.

This study aims to develop ML models capable of predicting Bitcoin's price direction across multiple time horizons, ranging from one to 30 days. We evaluate the efficacy of different modeling approaches and predictor variable types. The findings of this work contribute to the understanding of Bitcoin price predictability and provide insights into the effectiveness of various data sources and ML techniques in this highly volatile market.

The specific objectives of this work are to explore various data sources, including technical, fundamental, and time-series metrics. It involves the implementation and comparison of several machine learning models, with performance assessed over different prediction periods. Evaluation metrics such as precision, recall, and F1-score are analyzed to measure the effectiveness of the models. Additionally, the impact of market volatility and conditions on prediction accuracy is thoroughly investigated.

The experimental results demonstrate that technical features are particularly effective for short-term forecasting when paired with tree-based models, while time series features offer more consistent performance over medium and long-term periods. Among the evaluated models, XGBoost consistently achieved superior F1-scores and trading returns. These findings underscore the importance of tailored feature engineering and model selection in developing profitable and robust cryptocurrency trading strategies.

The structure of the paper is as follows. Section II presents the related works, while the methodology adopted in this research is shown in Section III. Section IV presents and discusses the results. Finally, conclusions are drawn in Section V.

II. Related Works

This section offers a clear summary of the relevant literature, examining the methodological approaches and assessing the effectiveness of various data types and models in predicting asset prices.

Several studies have explored the use of ML for predicting stock market trends. In one study [6], the authors introduced the ensemble of a boosted hybrid of deep learning models and technical analysis for forecasting stock prices algorithm. This algorithm combines convolutional neural networks with long short-term memory (LSTM) networks and demonstrates strong performance. However, it also highlighted the risk of overfitting in neural networks.

In another study [7], researchers incorporated stock prices and news sentiment into their analysis. They found that neural networks produced the best results in trading simulations. Lastly, a hybrid model was developed in [8] using LightGBM and XGBoost, specifically targeting emerging markets. This model yielded mixed results across different market conditions.

Comparative analyses of technical and fundamental approaches to stock prediction have been conducted. One study [9] found that a hybrid model, which utilized both technical and fundamental data, outperformed a model that relied solely on technical data. Another research [10] indicated that while technical analysis tends to perform better in the short term, combining both types of analysis produces the most effective results. Additionally, in [11], the authors compared classification and regression methods, concluding that classification models typically demonstrate superior performance in algorithmic trading strategies.

In cryptocurrency, LSTM networks are often employed for price prediction because of their capability to manage long-term dependencies. A study by [12] introduced an LSTM model enhanced with Bayesian optimization for predicting Bitcoin prices. While it reported promising results, it did not include comparisons with other models. In another study [13], the authors compared the performance of LSTM, gated recurrent unit (GRU), and bidirectional LSTM (Bi-LSTM) models, finding that Bi-LSTMs yielded the best results. Furthermore, a hybrid model combining 1D convolutional neural networks and GRU was presented in [14], outperforming the other tested methods.

More recently, transformer-based models have gained prominence in financial forecasting. In [16], the authors proposed a deep learning approach for asset price and directional movement prediction using Vision Transformers (ViT), Swin Transformers, and ConvMixer architectures. By converting time series data into 2D images of technical indicators, the study demonstrated that transformer-based models significantly outperform traditional CNNs and common baselines such as LSTM and MLP. Notably, the ViT model achieved the best financial performance in trading simulations, surpassing even enhanced CNN-based methods.

Similarly, [17] presented a model that integrates 1D convolutional neural networks and Transformers to forecast intraday stock price direction for S&P 500 constituents. Their approach, called CTTS, captures both short-term and long-term dependencies in the data. Experimental results showed that CTTS outperformed traditional statistical models like ARIMA and EMA, as well as the DeepAR deep learning model. The method also achieved high classification accuracy, especially under a confidence threshold, indicating the reliability of its probabilistic outputs for trading decisions.

These works demonstrate the application of various ML techniques to both stock and cryptocurrency markets. It emphasizes the importance of feature engineering, model selection, and hyperparameter tuning for effective price prediction, while also highlighting the growing relevance of transformer architectures in financial time series forecasting.

III. Methodology

This section details the data sources, feature engineering, label construction, preprocessing, modeling, and evaluation steps used in the study. The goal is to predict the directional movement of Bitcoin prices over multiple future horizons by comparing different modeling strategies and sets of predictive variables. Figure 1 illustrates the complete framework of the Bitcasting proposal.

We examined two primary data sources covering the period from July 17, 2017, to March 5, 2024. The first data source was obtained from Binance, and it includes daily information on open, high, low, and close prices, as well as the volume traded in BTC and USDT, and the number of trades. The second database was acquired from CoinMetrics and has 146 on-chain blockchain metrics reflecting network activity, valuation, fees, mining difficulty, and wallet behavior. Data were merged based on matching dates. The Binance dataset had no missing values. Two variables in the CoinMetrics dataset, which had over 20% missing data, were discarded due to negligible correlation with the target. The remaining gaps were filled using forward-filling.

Based on the initial data acquired, we defined three distinct modeling approaches, each built upon a specific set of features that were specifically engineered for each approach. The first one is the **technical approach**, where the feature set is based on well-established technical indicators that are commonly used in financial market analysis [15]. Each indicator is expanded into multiple features by adjusting parameters such as time windows and breaking them down into components (e.g., signal, distance from price, or normalized scores). The technical indicators considered include:

(a) *Simple Moving Average (SMA)* and *Exponential Moving Average (EMA)*: These indicators are calculated over

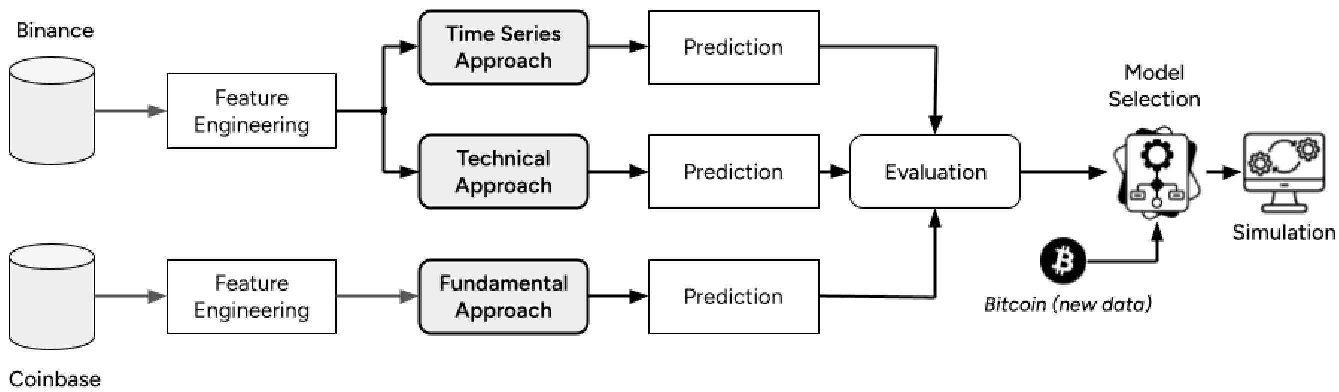


Fig. 1: Framework of the Bitcasting proposal.

multiple window lengths (3, 5, 7, 20, and 50 days) to capture both short-term and long-term trends.

(b) *Relative Strength Index (RSI)*: A momentum oscillator computed with a 14-day window (RSI-14) that quantifies the magnitude and velocity of recent price movements.

(c) *Bollinger Bands*: A volatility indicator constructed using a moving average and standard deviations, which helps assess potential price overextensions or contractions.

(d) *On Balance Volume (OBV)*: A cumulative, volume-based indicator that relates trading volume to price movement direction, capturing buying or selling pressure.

(e) *Pivot Points*: These indicators are key levels of potential support and resistance derived from previous high, low, and closing prices, useful for forecasting potential price reversals.

The second proposition outlines a **fundamental approach** in which we utilize on-chain metrics extracted from the Coin-Metrics database. These metrics are grouped into categories, including active addresses, market capitalization, transaction count, miner revenue, wallet balances, and hash rate. To enhance the clarity of our analysis, we removed highly correlated variables (where the correlation coefficient $r > 0.95$) using hierarchical clustering.

The final approach involves using a **time series analysis** where we generate lagged versions of price and volume, as well as their logarithmic and first differences. To assess stationarity, we utilized the Augmented Dickey-Fuller test and applied log-differencing as necessary. Additionally, we incorporated lag features ranging from 1 to 20 days to capture temporal dependencies effectively.

A. Target Labeling

The target variable is defined as a 3-class label that indicates the expected price movement over a future horizon of $p \in \{1, 3, 7, 14, 30\}$ days, where p represents the number of days ahead for the return forecast.

To begin, we calculate the return over p days from time t, denoted as $R_{t,p}$, using the formula

$$R_{t,p} = \frac{P_{t+p} - P_t}{P_t}\,, \tag{1}$$

where P_t and P_{t+p} are the closing prices at times t and $t+p$, respectively.

After that, we compute the volatility using the 20-day rolling standard deviation of $R_{t,p}$, which we denote as $\sigma_{t,p}$. This parameter reflects the estimated volatility of the p-day return at time t, computed over a 20-day window to help smooth out fluctuations.

To establish thresholds, we utilize symmetric thresholds set at $\pm\alpha \cdot \sigma_{t,p}$, where $\alpha = 0.75$. This scaling factor defines a neutral zone around a zero return, allowing for clearer interpretation of the data.

Given the above, the target class label at time t for horizon p, denoted as $y_{t,p}$, can be obtained as follows

$$y_{t,p} = \begin{cases} -1, & \text{if } R_{t,p} < -\alpha \cdot \sigma_{t,p} \\ 0, & \text{if } |R_{t,p}| \leq \alpha \cdot \sigma_{t,p} \\ +1, & \text{if } R_{t,p} > \alpha \cdot \sigma_{t,p} \end{cases} \tag{2}$$

with -1, 0, and $+1$ indicating, respectively, a *sell* signal, a *hold* signal, and a *buy* signal.

B. Preprocessing

All features are standardized using z-score normalization. In addition, we analyze outliers in price differences using the interquartile range method. Despite identifying several outliers, we retain them due to their high incidence and potential informational value in financial contexts.

To promote stationarity in the time series approach, two transformations are applied. The first is called first-differencing, which is used to eliminate linear trends. It is defined as

$$\Delta x_t = x_t - x_{t-1}\,, \tag{3}$$

where Δx_t represents the first difference of the time series. The second transformation is log-differencing, expressed as

$$\Delta \log x_t = \log(x_t) - \log(x_{t-1}) = \log\left(\frac{x_t}{x_{t-1}}\right). \tag{4}$$

This transformation approximates continuous returns and helps stabilize the variance of multiplicative processes.

No dimensionality reduction beyond correlation filtering was applied; features with high pairwise correlation (above a chosen threshold) were removed to decrease multicollinearity.

C. Evaluation

For each of the three approaches—technical, fundamental, and time series—we train four types of classifiers: `(a)` logistic regression (LR) with L2 regularization; `(b)` XGBoost, a gradient-boosted tree ensemble); `(c)` support vector machine (SVM) with a linear kernel; and `(d)` random forest (RF), a bagging-based ensemble of decision trees.

We perform hyperparameter tuning using grid search with 5-fold rolling-window cross-validation to preserve the temporal order of the data. For each forecast horizon p, models are trained and evaluated separately, allowing us to compare the performance of each approach across different timeframes, from short-term to long-term. Algorithm 1 illustrates each step of the modeling process outlined above.

Algorithm 1 Classification algorithm with hyperparameter tuning

Define models, horizons, and hyperparameters:
Models = {XGBoost, SVM, LR, RF}
periods = {1, 3, 7, 14, 30}
Define hyperparameter grids for each model

for model in Models **do**
  for horizon in periods **do**

    Data preparation:
    Remove rows with null values in the target for the given horizon
    Separate features (X) and target variable (y)
    Apply StandardScaler to standardize features

    Time series cross-validation:
    Create TimeSeriesSplit with 5 splits maintaining temporal order

    Randomized hyperparameter search:
    Use RandomizedSearchCV with 10 iterations
    Optimize using weighted F1-score
    Run in parallel using all available cores

    Result storage:
    Save best parameters for each model and horizon
    Generate a JSON file with the results
  end for
end for

Model performance is evaluated using ML-based metrics, including accuracy, precision, recall, and the macro-averaged F1-score. Additionally, a simplified trading simulation assessed maximum drawdown and the Sharpe ratio. In this trading setup, predictions are translated into position signals: long (+1), short (-1), or neutral (0). Returns were calculated to determine the economic viability of the model. For simplicity, transaction costs are set at 0.1% of the transaction value. The ML-based metrics used in this work are defined as follows:

1) Accuracy per class k (Ac_k):
$$Ac_k = \frac{TP_k + TN_k}{TP_k + TN_k + FP_k + FN_k}$$
where TP_k, TN_k, FP_k, and FN_k denote the true positives, true negatives, false positives, and false negatives for the k-th class, respectively.
2) Precision per class k (Pr_k):
$$Pr_k = \frac{TP_k}{TP_k + FP_k}$$
3) Recall per class k (Re_k):
$$Re_k = \frac{TP_k}{TP_k + FN_k}$$
4) Macro-averaged F1-score ($F1_{\text{macro}}$):
$$F1_{\text{macro}} = \frac{1}{K}\sum_{k=1}^{K} \frac{2 \cdot Pr_k \cdot Re_k}{Pr_k + Re_k}$$
where K is the number of classes.

In our trading simulation, we assess the portfolio return at time t, denoted as r_t, which is calculated using the formula
$$r_t = s_t \cdot r_{t,1}\,,$$
where $s_t \in \{-1, 0, +1\}$ is the trading signal derived from the model prediction, and $r_{t,1}$ is the actual next-day return. We also obtained the Sharpe ratio using the formula
$$Sharpe\ Ratio = \frac{\mathbb{E}[r_t - R_f]}{\sigma(r_t)}\,, \tag{5}$$
where R_f is the risk-free rate (assumed to be zero for simplicity), and $\sigma(r_t)$ is the standard deviation of returns, and $\mathbb{E}[\cdot]$ represents the expected value operator. This metric evaluates the risk-adjusted return of the strategy, capturing how much excess return is generated per unit of volatility. Finally, we calculated the maximum drawdown, represented as MDD, such that
$$MDD = \max_{t\in[0,T]} \left(\frac{\max_{\tau\in[0,t]} P_\tau - P_t}{\max_{\tau\in[0,t]} P_\tau} \right), \tag{6}$$
where P_t is the cumulative portfolio value at time t, T is the total number of periods in the investment horizon, and $\max_{\tau\in[0,t]} P_\tau$ represents the historical peak value of the portfolio up to τ. The expression computes the largest percentage drop from a historical peak to a subsequent trough over the entire investment period. Maximum drawdown quantifies the largest observed loss from a peak to a trough, reflecting downside risk and helping assess capital preservation under adverse conditions.

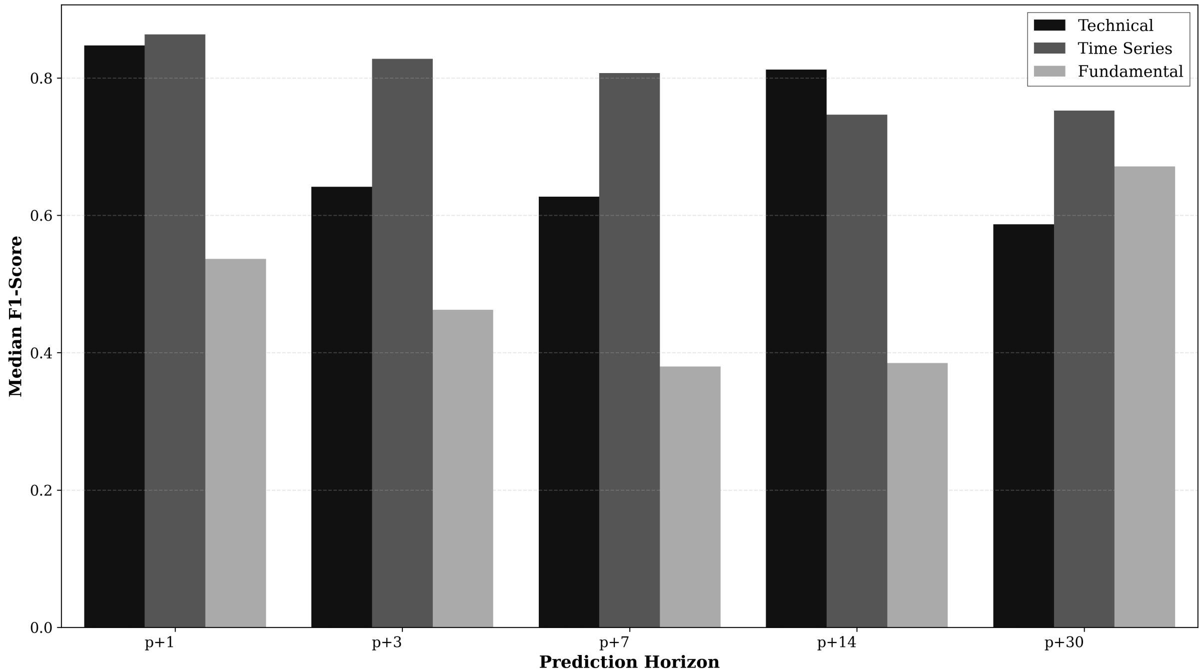


Fig. 2: Median F1-scores for the evaluated models (XGBoost, RF, LR, and SVM) are presented for each approach, categorized by time horizon p. Short-term horizon: $p + 1$, $p + 3$. Mid-term horizon: $p + 7$, $p + 14$. Long-term horizon: $p + 30$.

All simulation experiments are implemented in Python 3.10 using `scikit-learn 1.2`, `XGBoost 1.7`, and custom scripts[1].

IV. Results

This section discusses the experimental results obtained by applying various modeling approaches, including technical, fundamental, and time series methods, along with classifiers such as LR, RF, XGBoost, and SVM. The analyses covered five daily horizons $p = \{1, 3, 7, 14, 30\}$. We evaluated performance using classification metrics and conducted a simplified trading simulation for comparison.

Table I presents the macro-averaged F1-scores for each model and feature approach combination, averaged across all daily horizons. Overall, SVM and LR achieved the highest scores, particularly when using time series features. The time series approach provided the best results across most models, with LR and SVM reaching a macro F1-score of 0.83. The technical features yielded competitive results, especially for SVM (0.75) and LR (0.74), while fundamental features underperformed in comparison, except for XGBoost, which showed a F1-score of 0.57 with this input type. These results suggest that time series data are more informative for the classification task, and linear models tend to generalize better in this context.

[1]Code and processed datasets are publicly available at https://github.com/hfernandescfc/bitcasting.

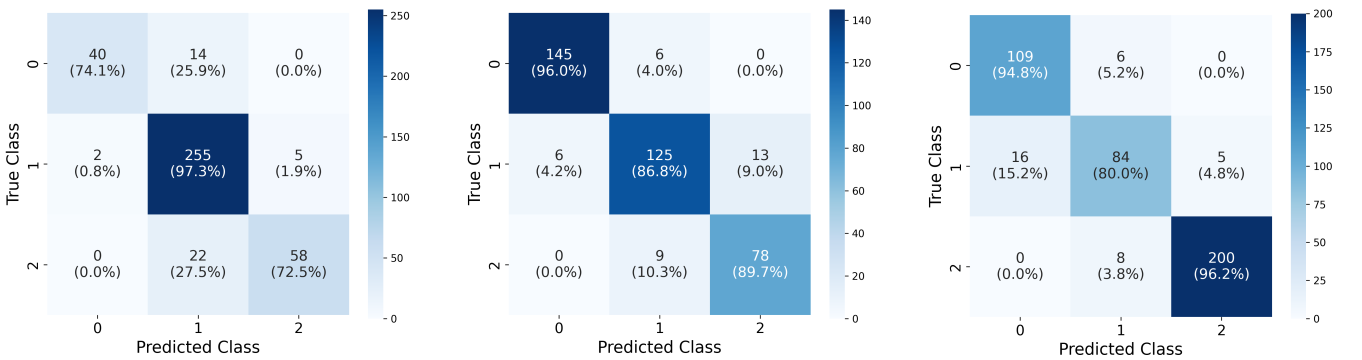
TABLE I: Macro-averaged F1-scores per model and feature approach.

Model	Technical	Fundamental	Time Series
LR	0.74	0.48	**0.83**
RF	0.58	0.47	0.74
XGBoost	0.67	**0.57**	0.77
SVM	**0.75**	0.48	**0.83**

Figure 2 illustrates the median F1-score trend across different prediction horizons for each feature approach. The time series approach consistently outperforms the others across all horizons, especially in the short term ($p + 1$ and $p + 3$), maintaining high performance even as the horizon increases. The technical approach shows strong short- to medium-term results, peaking at (p+14), but experiences a noticeable decline at the most extended horizon ($p + 30$). While consistently trailing the others, the fundamental approach shows its best relative performance at ($p + 30$) yet remains below the time series approach. These results suggest that time series features are more robust across different time frames, while technical features may be better suited for short to medium-term predictions.

For each horizon, confusion matrices were created to gain a deeper understanding of the model's behavior, as shown in Figure 3. In shorter horizons (e.g., $p = 1$), most models favored the "hold" class due to the high frequency of neutral returns. This trend shifts as we move to longer horizons; the "sell" class becomes more significant during medium periods, while the "buy" class predominates in longer horizons. This indicates that, over the long term, Bitcoin generally tends to increase in value.

A simplified trading simulation was conducted using the model's predictions to generate long positions ($\hat{y} = +1$), short



(a) Time series approach — short-term ($p = 1$). (b) Technical approach — mid-term ($p = 14$). (c) Fundamental approach — long-term ($p = 30$).

Fig. 3: Confusion matrices for different feature approaches and time horizons.

positions ($\hat{y} = -1$), and neutral positions ($\hat{y} = 0$). Figure 4 displays the cumulative return curves for the time series approach using XGBoost across three different prediction horizons ($p \in \{3, 7, 30\}$ days). These results are compared against a standard buy-and-hold (B&H) strategy.

The strategy for the $(t + 3)$ timeframe demonstrates the highest cumulative return over the whole period, significantly outpacing the corresponding B&H baseline. This suggests that the model effectively identifies meaningful short-term signals that can be leveraged for profit. The $(t + 7)$ horizon strategy delivers competitive returns but with greater volatility, sometimes underperforming compared to B&H during specific intervals. In contrast, the $(t + 30)$ strategy shows strong initial spikes, likely due to high-confidence signals, but later experiences instability and increased volatility, indicating a lack of robustness for long-term predictions. These results reinforce that shorter horizons produce more stable and profitable trading strategies when using this predictive setup. In contrast, longer-term strategies may be prone to overfitting or deteriorating signal quality.

Table II presents a comparison of the returns generated by the proposed strategy with those of the B&H strategy. The results indicate that the Time Series approach yields higher profitability, particularly for short ($p = 3$) and long ($p = 30$) investment horizons. In both instances, our strategy surpasses the B&H benchmark, achieving gains of 347% and 607%, respectively. Additionally, while the Bitcasting strategy produces competitive Sharpe ratios in the analyzed scenarios, it also demonstrates significantly higher maximum drawdown across all horizons, suggesting greater exposure to tail risk.

V. Conclusion

This work presented a machine learning (ML) framework, named *Bitcasting*, designed to forecast Bitcoin's directional price movements across multiple time horizons using three distinct modeling approaches—technical indicators, on-chain fundamental metrics, and time series features—combined with four classification algorithms: logistic regression (LR), support vector machine (SVM), random forest (RF), and XGBoost.

TABLE II: Performance metrics of the Bitcasting proposal and buy-and-hold (B&H) strategy by investment horizon p.

Metric	$p = 3$	$p = 7$	$p = 30$
Simulated trades	400	170	40
Bitcasting return	347.98%	144.33%	607.07%
B&H return	263.94%	242.84%	320.79%
Bitcasting Sharpe (annual)	0.98	0.73	0.96
B&H Sharpe (annual)	0.93	0.89	0.93
Bitcasting max drawdown	-84.12%	-87.17%	-75.21%
B&H max drawdown	-77.21%	-74.43%	-71.66%

Experimental evaluation revealed that time series and technical features are the most effective for predicting short- to medium-term price movements. Despite their relative simplicity, LR and SVM achieved top-tier performance—particularly when paired with time series features—attaining macro-averaged F1-scores of 0.83. These findings indicate that lightweight linear models can provide reliable and interpretable forecasts under the right feature conditions. XGBoost consistently demonstrated strong classification performance and enhanced trading profitability across various time horizons. In contrast, fundamental features exhibited limited predictive power when analyzed in isolation, likely due to the delayed nature of blockchain-level metrics.

The conducted simulation validated the practical viability of the proposed models. The proposed strategy outperformed the buy-and-hold benchmark in cumulative returns and risk-adjusted performance. Notably, the Bitcasting framework delivered cumulative gains of 347% and 607% at the 3-day and 30-day horizons, respectively, while maintaining comparable or superior Sharpe ratios. These results highlight the potential of ML models to generate effective trading signals in volatile markets, although the observation of higher drawdowns emphasizes the importance of risk management.

Future research will focus on advancing the models through the integration of deep learning architectures (e.g., long short-term memory, Transformer) to enhance temporal dependency capture and by incorporating sentiment analysis from news and social media to enrich feature sets. Further developments include applying ensemble methods to combine predictions

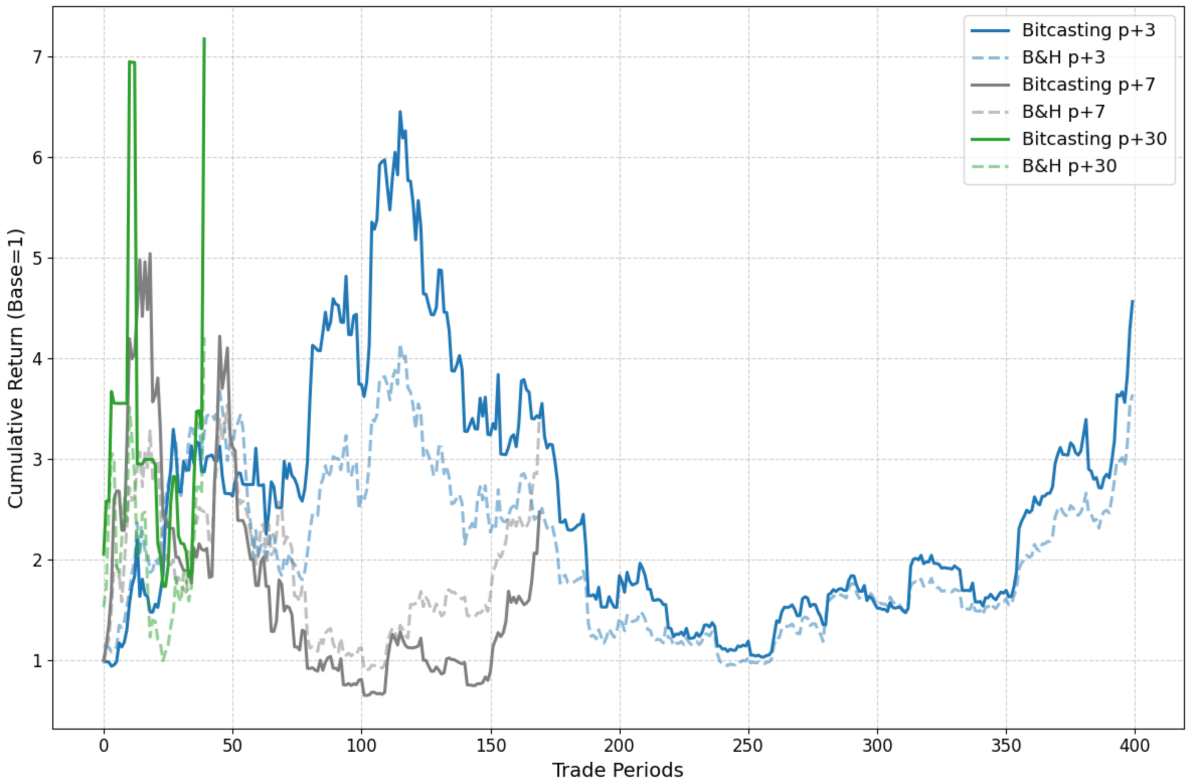


Fig. 4: Simulated cumulative returns using time series approach and buy-and-hold (B&H) strategy simulated for different time horizons.

across diverse approaches and horizons, and extending the framework to encompass additional crypto assets and facilitate cross-asset learning. Crucially, continuous model updating and online learning strategies are essential for maintaining robust real-time prediction capabilities within the highly volatile and rapidly evolving crypto markets.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available at https://bitcoin.org/bitcoin.pdf. Accessed: 02 May 2025.

[2] Quartz, "By reading this page, you are mining bitcoins," 2014. [Online]. Available: https://qz.com/154877/by-reading-this-page-you-are-mining-bitcoins. Accessed: 02 May 2025.

[3] CoinMarketCap, "Bitcoin (BTC) Preço Hoje, Gráfico, Informações do Mercado," 2025. [Online]. Available: https://coinmarketcap.com/pt-br/currencies/bitcoin/?utm_source=chatgpt.com. Accessed: 10 Mar 2025.

[4] D. L. John, S. Binnewies, and B. Stantic, "Cryptocurrency price prediction algorithms: A survey and future directions," *Forecasting*, vol. 6, no. 3, pp. 637–671, 2024.

[5] E. F. Fama, "Efficient capital markets," *J. Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[6] A. F. Kamara, E. Chen, and Z. Pan, "An ensemble of a boosted hybrid of deep learning models and technical analysis for forecasting stock prices," *Information Sciences*, vol. 594, pp. 1–19, 2022.

[7] A. Picasso, S. Merello, Y. Ma, L. Oneto, and E. Cambria, "Technical analysis and sentiment embeddings for market trend prediction," *Expert Systems with Applications*, vol. 135, pp. 60–70, 2019.

[8] O. Guennioui, D. Chiadmi, and M. Amghar, "Machine learning-driven stock price prediction for enhanced investment strategy," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 5, pp. 5884–5893, 2024.

[9] A. Adebiyi, C. Ayo, M. Adebiyi, and O. S. Otokiti, "Stock price prediction using neural network with hybridized market indicators," *J. Emerg. Trends Comput. Inf. Sci.*, vol. 3, pp. 1–9, 2012.

[10] E. Beyaz, F. Tekiner, X. Zeng, and J. Keane, "Comparing technical and fundamental indicators in stock price forecasting," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun. (HPCC)*, 2018, pp. 1607–1613.

[11] H. Wang and D. Xie, "Optimal profit-making strategies in stock market with algorithmic trading," *Quantitative Finance and Economics*, vol. 8, pp. 546–572, 2024.

[12] E. S. Pour, H. Jafari, A. Lashgari, E. Rabiee, and A. Ahmadisharaf, "Cryptocurrency price prediction with neural networks of LSTM and Bayesian optimization," *Eur. J. Bus. Manag. Res.*, vol. 7, no. 2, pp. 20–27, 2022.

[13] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza, "Forecasting cryptocurrency prices using LSTM, GRU, and bi-directional LSTM: A deep learning approach," *Fractal Fract.*, vol. 7, no. 2, Art. no. 203, 2023.

[14] C. Kang, C. P. Lee, and K. Lim, "Cryptocurrency price prediction with convolutional neural network and stacked gated recurrent unit," *Data*, vol. 7, no. 11, Art. no. 149, 2022.

[15] F. Lemos, Technical Analysis of Financial Markets (in Portuguese), 528 p., 2nd ed., Saraiva Uni, 2018.

[16] A. H. B. Gezici and E. Sefer, "Deep Transformer-Based Asset Price and Direction Prediction," *IEEE Access*, vol. 12, pp. 24164–24175, Jan. 2024, doi: 10.1109/ACCESS.2024.3358452.

[17] Z. Zeng, R. Kaur, S. Siddagangappa, S. Rahimi, T. Balch, and M. Veloso, "Financial Time Series Forecasting using CNN and Transformer," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2023. [Online]. Available: https://arxiv.org/abs/2304.04912