INSTITUTO POLITÉCNICO DE SANTARÉM

Escola Superior de Gestão e Tecnologia de Santarém

A Leakage-Aware Benchmark of Hourly Bitcoin Return Forecasting: Naive, Statistical, and Machine Learning Models under Walk-Forward Validation

Dissertation

Mestrado em Informática Aplicada

Vasile Timotin

Supervisors:

Pedro Nuno de Alexandre Sobreiro

Filipe Montez Coelho Madeira

May 2026

POLITÉCNICO
DE SANTARÉM

Dedication

To my family, whose encouragement, values, and sacrifices made this journey possible, and to my girlfriend, whose patience and affection have been a constant source of strength.

Acknowledgements

I express my sincere gratitude to my friends, whose companionship and support provided strength and balance during the most demanding stages of this research. I also thank my professors, and especially my supervisors, Pedro Nuno de Alexandre Sobreiro and Filipe Montez Coelho Madeira, whose guidance, critical insight, and commitment to academic rigor were fundamental to the development of this dissertation.

Acronyms

Table 1. Acronyms used in the dissertation.

ACRONYM	FULL TERM	CATEGORY
AI	Artificial Intelligence	General
AIC	Akaike Information Criterion	Model selection
APA	American Psychological Association (7th edition style)	Formatting
API	Application Programming Interface	Data / systems
ARIMA	AutoRegressive Integrated Moving Average	Statistical model
ARIMAX	AutoRegressive Integrated Moving Average with eXogenous inputs	Statistical model
ARMA	AutoRegressive Moving Average	Statistical model
BIC	Bayesian Information Criterion	Model selection
BTC	Bitcoin	Market
BTC-USD	Bitcoin / United States Dollar pair	Market
CCXT	CryptoCurrency eXchange Trading library	Data / systems
CSV	Comma-Separated Values	Data format
CV	Cross-Validation	Evaluation
DA	Directional Accuracy	Evaluation metric
DM	Diebold–Mariano test	Statistical test
DNN	Deep Neural Network	Deep learning model
EDA	Exploratory Data Analysis	Data analysis
ETF	Exchange-Traded Fund	Market
ETL	Extract, Transform, Load	Data engineering
GRU	Gated Recurrent Unit	Deep learning model
HLN	Harvey–Leybourne–Newbold correction	Statistical test
HMM	Hidden Markov Model	Regime-aware model
LIGHTGBM	Light Gradient Boosting Machine library	Data / systems
LSTM	Long Short-Term Memory	Deep learning model
MACD	Moving Average Convergence Divergence	Technical indicator
MAE	Mean Absolute Error	Evaluation metric
MAPE	Mean Absolute Percentage Error	Evaluation metric
MASE	Mean Absolute Scaled Error	Evaluation metric
ML	Machine Learning	Modelling approach
NAN	Not a Number (missing value)	Data format
NLP	Natural Language Processing	Modelling approach
OBV	On-Balance Volume	Technical indicator
OHLCV	Open–High–Low–Close–Volume	Market data
OLS	Ordinary Least Squares	Regression model
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses	Literature review
RF	Random Forest	Machine learning model
RMSE	Root Mean Squared Error	Evaluation metric

ACRONYM	FULL TERM	CATEGORY
RSI	Relative Strength Index	Technical indicator
SARIMA	Seasonal AutoRegressive Integrated Moving Average	Statistical model
SEC	United States Securities and Exchange Commission	Market
SHAP	SHapley Additive exPlanations	Model interpretation
SMA	Simple Moving Average	Technical indicator
SMAPE	symmetric Mean Absolute Percentage Error	Evaluation metric
USDT	Tether (US dollar-pegged stablecoin)	Market
UTC	Coordinated Universal Time	Time standard

POLITÉCNICO DE SANTARÉM

Resumo

Uma Avaliação Comparativa Atenta à Fuga de Informação na Previsão de Retornos Horários do Bitcoin: Modelos Ingénuos, Estatísticos e de Aprendizagem Automática sob Validação Preditiva Sequencial (Walk-Forward).

Esta dissertação avalia se a aprendizagem automática prevê de forma fiável os retornos horários do Bitcoin sob um protocolo reprodutível, imune à contaminação do treino por informação futura. Os dados reúnem preços, volumes e indicadores técnicos horários do Bitcoin na Binance, de agosto de 2017 a janeiro de 2025, com variáveis externas onde a cobertura o permitiu. Compararam-se sete modelos — dois ingénuos, uma regressão linear, uma floresta aleatória, dois modelos clássicos de séries temporais e uma rede neuronal — em horizontes de 1, 6 e 24 horas, com validação preditiva sequencial e purga de 24 horas em 53 segmentos do último ano. Nenhum modelo superou a previsão de retorno nulo em erro absoluto médio; a diferença para o melhor concorrente, a regressão linear, não foi estatisticamente significativa. O contributo é metodológico: propõe um protocolo reprodutível e mostra que os ganhos preditivos obtidos sob validações permissivas não resistem a um teste rigoroso.

Palavras-chave: Bitcoin, previsão de séries temporais, aprendizagem automática, validação preditiva sequencial (walk-forward), fuga de informação, Diebold–Mariano

Abstract

This dissertation evaluates whether machine-learning models reliably forecast hourly Bitcoin returns under a reproducible protocol that prevents future information from leaking into training. The data comprise hourly Bitcoin prices, trading volumes and technical indicators from the Binance exchange, from August 2017 to January 2025, together with external variables for the periods with sufficient coverage. Seven models were compared — two naive benchmarks, a linear regression, a random forest, two classical time-series models and a recurrent neural network — at horizons of 1, 6 and 24 hours, using walk-forward validation with 24-hour purge gaps across 53 segments of the final year of the sample. No model outperformed a zero-return forecast in mean absolute error, and the gap to the closest challenger, the linear regression, was not statistically significant. The contribution is methodological: it proposes a reproducible evaluation protocol and shows that the predictive gains reported under permissive validation do not survive rigorous testing.

Keywords: Bitcoin, time-series forecasting, machine learning, walk-forward validation, data leakage, Diebold–Mariano

POLITÉCNICO DE SANTARÉM

Contents

List of Figures

List of Tables

1 Introduction

Bitcoin started in 2008 as a peer-to-peer electronic cash system (Nakamoto, 2008) and has since become a widely traded financial asset with growing academic and practical interest. Its liquidity, global reach, and rapid adoption made it a natural subject for quantitative finance, data science, and machine learning. Forecasting Bitcoin, however, is hard: prices are highly volatile, the series is structurally unstable, and market conditions change abruptly, especially at short intraday horizons (Fama, 1970; McNally et al., 2018; Mudassir et al., 2020). These features challenge both standard econometric assumptions and the practical value of predictive models in live trading (Box et al., 2015; López de Prado, 2018).

However, these findings should be interpreted with caution. The existing evidence is highly heterogeneous in terms of datasets, sampling frequencies, target definitions, feature sets, and validation procedures. Some studies rely on random or weakly constrained train-test splits, while others evaluate model performance on transformed or log-scaled targets, which can make forecasting results appear stronger than they are in an economically meaningful price space. Mudassir et al. (2020), for instance, note that evaluations based on log-normalised prices may understate the true magnitude of forecasting error. Moreover, several studies focus primarily on daily data, where smoother dynamics may create an impression of stronger predictability than that achievable in more demanding hourly or intraday settings (Chen et al., 2020; Mudassir et al., 2020). Consequently, strong headline metrics do not necessarily imply robust out-of-sample predictive values under realistic market conditions.

A central methodological issue is that Bitcoin forecasting is particularly vulnerable to look-ahead bias and temporal leakage. In financial time series, even subtle contamination between the training and test periods can produce overly optimistic conclusions. This is especially problematic when predictors are derived from rolling indicators, mixed-frequency exogenous variables, or preprocessing steps applied globally rather than separately within each training split (López de Prado, 2018). The problem becomes even more acute at high frequencies, where the signal-to-noise ratio is low and where daily sentiment or network variables, even when causally aligned, may be too slow-moving to explain hourly return fluctuations (Fama, 1970; Serafini et al., 2020). Recent work that enforces walk-forward validation with purge gaps and training-only preprocessing reports that most standard models fail to achieve stable superiority over naive benchmarks at the hourly horizon (Jaquart et al., 2021).

A central tension runs through the Bitcoin forecasting literature: parts of the field report optimistic results, while stricter out-of-sample evaluations produce far more cautious evidence. Several prior studies have reported encouraging results for machine learning and deep learning models, particularly in settings focused on daily price-level prediction or directional classification (Chen, 2023; Kashyap et al., 2024; McNally et al., 2018; Mudassir et al., 2020; Nagula & Alexakis, 2022). However, more recent research suggests that predictive gains in cryptocurrency markets are often substantially weaker when assessed using more rigorous temporal validation, stronger benchmark comparisons, and greater sensitivity to regime dependence (Bouri et al., 2022; Cakici et al., 2024; Cortese et al., 2023; Yae & Tian, 2022).

In light of this tension, the central research problem in short-horizon Bitcoin forecasting is no longer merely whether returns can be predicted using machine learning methods, but whether any apparent predictive gains remain statistically and methodologically credible after controlling for non-stationarity, temporal dependence, mixed-frequency feature alignment, and leakage risk (Diebold & Mariano, 1995; López de Prado, 2018).

1.1 Research Objective, Specific Objectives, and Research Question

The general objective of this dissertation is to evaluate whether machine learning models can achieve consistent and methodologically credible out-of-sample improvements over strong naive benchmarks in hourly Bitcoin return forecasting under a reproducible and leakage-aware validation design.

To achieve this general objective, this study pursues the following specific objectives:

1. To construct a reproducible hourly dataset for Bitcoin forecasting built on causally aligned market and technical variables, incorporating sentiment and network-related variables only where their effective intraday coverage permitted, and explicitly documenting feature-domain coverage at the hourly grid.
2. To define leakage-safe forecasting targets and preprocessing procedures that preserve temporal ordering and prevent the use of future information.
3. To compare naive baselines (Naive0, NaiveLast), a classical statistical baseline (ARIMA), a linear multivariate reference (OLS), and a non-linear ensemble learner (Random Forest) under a common rolling-origin walk-forward protocol with 24-hour purge gaps at the 1-hour, 6-hour, and 24-hour horizons, with an auxiliary LSTM track reported separately as configuration-specific evidence.
4. To evaluate whether model flexibility and exogenous inputs provide robust incremental predictive value relative to strong naive benchmarks across horizons.

5. To interpret the empirical findings in terms of methodological robustness, practical limitations, and implications for future cryptocurrency forecasting research.

This dissertation is guided by the following research question:

Under a reproducible and leakage-aware forecasting design, do machine learning models provide consistent out-of-sample improvements over strong naive benchmarks in short-horizon Bitcoin return forecasting at the 1-hour, 6-hour, and 24-hour horizons?

1.2 Scope and Delimitation of the Study

The empirical scope of this study is delimited as a reproducible, leakage-aware benchmark of hourly Bitcoin return forecasting. The core like-for-like reproducible archive — i.e., the set of models with complete, harmonised split-level outputs preserved under the same protocol — comprises Naive0, NaiveLast, OLS, Random Forest, ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24), with the auxiliary LSTM track reported on 52 valid splits per horizon. All seven models are ranked against the Naive0 and NaiveLast baselines on the same metrics and the same statistical tests. The ARIMA and SARIMA results are interpreted in Section 4.2 in the light of a level-to-return reconversion artefact of the univariate log-price specification; whether a correctly specified ARIMA (estimated in return space, or as an ARIMAX with a calibrated reconversion) would alter the ranking is identified as a focused next step in Table 9.

The defended benchmark is, in practice, a market-and-technical evaluation. The original dissertation proposal envisaged a broader multimodal and regime-sensitive design; however, the clustering component was not retained, sentiment coverage was only partial on the hourly grid, and the usable hash-rate field contributed no non-missing hourly observations to the final benchmark table. These scope adjustments reflect what could be executed and documented to a reproducible standard, not a reassessment of the conceptual importance of such variables. The multimodal directions identified in Chapter 2 are therefore carried forward as priority extensions in Table 9 rather than as elements of the defended empirical comparison. Methodological justification for these decisions is provided in Section 3.1.

The three forecast horizons — 1-hour, 6-hour, and 24-hour — were retained to test robustness across difficulty levels, from the strictest and noisiest intraday setting to a comparatively less noisy intermediate horizon. Chapter 2 reviews the relevant literature; Chapter 3 details the leakage-aware methodology; Chapter 4 reports the empirical results and discussion; Chapter 5 presents the conclusions and future directions.

2 Literature Review

Bitcoin price forecasting has attracted substantial academic interest because the market combines continuous trading, high volatility, structural instability and rapid changes in investor sentiment. These characteristics have motivated the use of statistical, machine learning, and deep learning methods, particularly through the incorporation of technical indicators, behavioral signals, and nonlinear model structures (McNally et al., 2018; Phaladisailoed & Numnonda, 2018; Mudassir et al., 2020; Chen, 2023), as comprehensively surveyed in Khedr et al. (2021) and the PRISMA-based bibliometric mapping of Pečiulis et al. (2024). The review below covers the broader research agenda; the subset operationalised in the defended benchmark is delimited in Section 1.2 and Chapter 3. A comprehensive survey of deep learning applications to financial time series over 2005–2019 by Sezer et al. (2020) documents the rapid expansion of these methods across asset classes and identifies recurrent and convolutional architectures, together with attention-based variants, as the dominant model families in recent work.

A recurrent finding in the literature is that more flexible model classes can outperform simpler benchmarks under particular experimental settings, although such results are usually conditional on the forecast horizon, target definition, and evaluation design. Recurrent neural networks, especially LSTM-based models (Hochreiter & Schmidhuber, 1997), have been used to capture temporal dependence in the Bitcoin series, while tree-based and ensemble methods have been used to model non-linear relations among technical and exogenous variables (McNally et al., 2018; Wu et al., 2018; Mudassir et al., 2020). Ji et al. (2019) provide a systematic comparison of deep learning architectures for Bitcoin forecasting and find that LSTM-based models marginally outperform other architectures on the regression task while DNN-based models are preferred for directional classification, suggesting that no single architecture dominates across all task formulations.

Other studies have examined GRU models, decomposition-based approaches, and hybrid combinations of statistical and neural methods, which together suggest that Bitcoin forecasting may benefit from models that can represent non-linearity, temporal structure, and changing market conditions (Chung et al., 2014; Mittal & Geetha, 2022; Da Silva et al., 2020; Hashish et al., 2019). More broadly, hybrid forecasting designs that combine machine learning with optimisation-oriented search procedures have also been explored in adjacent financial prediction settings, suggesting that any apparent performance gain may depend not only on the learner itself but also on feature selection and tuning strategy (Sedighi et al., 2019). More recent work has extended this line of research through broader predictor sets and alternative machine learning designs, again reporting improved performance in

study-specific settings rather than under a single shared benchmark (Chen, 2023; Kashyap et al., 2024).

Hybrid, ensemble, and regime-aware approaches form another important strand in the literature. Da Silva et al. (2020) combine decomposition methods with ensemble learning for multi-step Bitcoin forecasting, while Wu et al. (2018) propose a hybrid LSTM-autoregressive framework intended to improve predictive stability. More broadly, hybrid designs are motivated by the view that a single model may be insufficient for a market characterised by abrupt shifts in volatility, trends, and momentum.

Related master's-level studies comparing ARIMA with LSTM or contrasting LSTM and GRU architectures likewise report context-specific improvements, but again under data and evaluation choices that are not directly comparable to a shared leakage-aware hourly benchmark (Mendes, 2019; Meijer, 2020; Xu, 2020). The Box–Jenkins ARIMA family and its seasonal extension (SARIMA) remain canonical references for univariate time-series forecasting (Box et al., 2015); at hourly frequency, the 24-hour daily cycle motivates a seasonal extension of the form SARIMA(p,d,q)(P,D,Q,24), which has been used as a baseline in intraday financial and energy forecasting work to absorb diurnal regularities that a non-seasonal ARIMA cannot represent. Peer-reviewed evidence on short-term Bitcoin prediction reaches a comparable conclusion: Munim et al. (2019) forecast next-day Bitcoin prices using statistical and machine-learning models and report that predictive accuracy depends substantially on the choice of model and feature set, while Jaquart et al. (2021) report that machine-learning gains over naive benchmarks in short-horizon Bitcoin forecasting are modest and depend strongly on the evaluation protocol, which reinforces the case for leakage-aware testing rather than weakly-validated optimism.

Four master's-level dissertations share the broad topic of Bitcoin forecasting and illustrate the heterogeneity of design choices in this literature. Mendes (2019) compares ARIMA against an LSTM on daily Bitcoin prices and concludes that the LSTM achieves lower error in the configurations evaluated, with the comparison conducted on a daily price-level target under a conventional train–test partition. Meijer (2020) and Xu (2020) focus on daily price or directional targets with LSTM and GRU specifications and rely on hold-out evaluation rather than rolling-origin testing with explicit purge gaps. Daş (2022) extends the feature space to network metrics and other financial assets, also operating at the daily frequency and on a price-level target. Across the four, target definition, data frequency, and out-of-sample validation design vary substantially, which limits the comparability of their reported error magnitudes.

Earlier exploratory evidence also includes relatively simple machine-learning applications to Bitcoin and broader cryptocurrency prediction, such as Raghava-Raju (2018) and Rathan et al. (2019). These studies are useful mainly as historical comparators, because their model scope and evaluation settings are less stringent than those used in leakage-aware hourly benchmarking.

Alongside these non-linear approaches, ordinary least squares (OLS) regression remains a transparent linear benchmark in financial forecasting and a natural reference under the Diebold and Mariano (1995) framework, since deviations from a multivariate linear specification can be tested directly against richer comparators. Although OLS is rarely the model of interest in recent Bitcoin forecasting work, its inclusion as a baseline is consistent with the forecasting tradition that recommends evaluating gains over a simple linear reference before attributing predictive power to more flexible learners (Hyndman & Koehler, 2006).

Beyond the comparator set operationalised in the executed benchmark, the literature has expanded to attention-based and Transformer-style architectures that represent a priority future direction for this research programme (Table 9). Lim et al. (2021) introduced the Temporal Fusion Transformer as an interpretable multi-horizon forecasting architecture that combines recurrent processing with attention mechanisms. In cryptocurrency applications, Farooq et al. (2024) and Kehinde et al. (2025) reported promising results for attention-based models that aimed to capture longer-range dependencies and complex temporal patterns.

These studies show that the methodological frontier has moved beyond conventional recurrent architectures. The evidence base for Transformer-style models in cryptocurrency forecasting nevertheless remains limited, since the underlying studies differ in targets, frequencies, features, and evaluation protocols, which constrains direct comparability across reported gains.

Also beyond the executed benchmark's model universe, regime-aware research reinforces the signal-conditionality interpretation and motivates a further priority extension (Table 9). Hashish et al. (2019) model Bitcoin price dynamics through Hidden Markov Models combined with LSTM networks; Bouri et al. (2022) show that regime-switching factor models can improve the forecasting of major cryptocurrency returns; and Cortese et al. (2023) find that cryptocurrency return dynamics are better described by a sparse, state-dependent structure than by a single stable process. Taken together, these studies support the interpretation that the predictive structure, where present, may be intermittent and

regime-dependent rather than constant across time — a finding that the present benchmark's single-window evaluation design cannot test but that future work extending the rolling-origin design to multiple regime windows should address.

The literature also highlights the importance of feature design and selection. Technical indicators such as moving averages, RSI, MACD, Bollinger Bands, and OBV are widely used to represent trends, momentum, volatility, and volume effects. Sentiment measures derived from social media or news are used to capture behavioral responses that may not be visible in price series alone (Mudassir et al., 2020; Serafini et al., 2020).

Some studies also incorporate blockchain-related variables, including hash rate, mining difficulty, and other network measures as proxies for system activity and market confidence (Saad & Mohaisen, 2018). More recent work has broadened this exogenous perspective by combining financial, macroeconomic, and explainable modelling approaches. For example, Goodell et al. (2023) show that explainable artificial intelligence can be used to identify which determinants appear most relevant in forecasting Bitcoin prices. Daş (2022) explored the role of network metrics and other financial assets in Bitcoin forecasting, reporting that the inclusion of such exogenous variables can shift predictive accuracy but that the temporal alignment of mixed-frequency inputs remains methodologically delicate. In general, the literature indicates that predictive performance depends not only on the model choice but also on the selection, transformation, and temporal alignment of explanatory variables.

Simultaneously, the inclusion of exogenous variables creates an important comparability problem. Many sentiment, macroeconomic, or network variables are available only at daily or irregular frequencies, whereas Bitcoin market data are often modelled at hourly or finer resolutions. Aligning slower-moving exogenous variables to higher-frequency market targets requires lagging, resampling, or forward-filling decisions that can materially affect both the leakage risk and apparent predictive power. This issue is especially important in short-horizon Bitcoin forecasting, where the signal-to-noise ratio is low and daily features may have limited explanatory value for hourly return variation (Fama, 1970; López de Prado, 2018; Serafini et al., 2020).

A further limitation concerns the methodological comparability across studies. The literature is highly heterogeneous in terms of sampling frequency, target definition, feature space, preprocessing strategy, and validation design; Kumbure et al. (2022), in a systematic review of machine learning methods applied to financial market forecasting, confirm that this diversity of design choices makes isolated performance claims across studies difficult

to generalise directly. Some studies forecast daily prices, others forecast returns, and others formulate the task as a directional classification. Some evaluate model performance in the transformed price space, while others report results in more directly interpretable error metrics. Validation procedures also vary from static or weakly constrained splits to more demanding rolling or walk-forward designs.

Consequently, reported improvements in metrics such as RMSE, MAPE, or directional accuracy are not directly comparable across studies and should not be automatically interpreted as robust evidence of out-of-sample predictive superiority (Chen, 2023; Mudassir et al., 2020; Yae & Tian, 2022). This caution is especially important when benchmark-relative claims are made without a shared loss definition or common out-of-sample evaluation framework (Diebold & Mariano, 1995).

This issue is particularly important in financial time series because even subtle forms of temporal leakage can yield optimistic results. In Bitcoin forecasting, leakage can arise from globally fitted preprocessing, rolling indicators that are not handled causally, or mixed-frequency exogenous variables whose publication timing is not properly aligned with the prediction target. These concerns become more severe at hourly horizons, where short-term return predictability is weak and daily or irregular external variables may be too coarse to provide a reliable incremental signal (Fama, 1970; López de Prado, 2018).

Recent evidence supports this cautious interpretation. Yae and Tian (2022) show that out-of-sample cryptocurrency forecasting performance depends materially on the choice of predictors and algorithms, whereas Cakici et al. (2024) find that, although machine learning can identify cross-sectional return predictability in cryptocurrencies, the incremental benefits from greater model complexity are limited.

Overall, the literature supports two conclusions. First, machine learning, deep learning, hybrid, and regime-aware methods can extract predictive structure from Bitcoin and broader cryptocurrency data under some settings (McNally et al., 2018; Mudassir et al., 2020; Hashish et al., 2019; Bouri et al., 2022; Cortese et al., 2023). Second, the strength and stability of these gains remain highly conditional on target specification, feature timing, horizon choice, market regime, and validation discipline (Jaquart et al., 2021; Yae & Tian, 2022; Cakici et al., 2024). Third, evidence from broader forecasting competitions reinforces this caution: the M4 competition demonstrated that simple and naive benchmarks remain extraordinarily difficult to outperform even by optimised machine-learning ensembles across diverse series types (Makridakis et al., 2020), a result that is consistent with the modest gains reported in short-horizon cryptocurrency forecasting. Isolated positive results should

therefore not be interpreted as general proof of persistent and robust predictability in the Bitcoin market.

The methods surveyed above represent the full frontier of the field. Of these, the executed benchmark operationalises a deliberately conservative subset — naive baselines, OLS, Random Forest, ARIMA(1,1,1), SARIMA(1,1,1)(1,0,1,24), and an auxiliary LSTM path — rather than the Transformer-based architectures (Lim et al., 2021; Farooq et al., 2024; Kehinde et al., 2025), Hidden Markov and regime-switching specifications (Hashish et al., 2019; Bouri et al., 2022; Cortese et al., 2023), or full multimodal feature sets incorporating intraday sentiment and on-chain data reviewed in the preceding sections. This restriction is intentional: the present study's contribution lies in the reproducibility and leakage-discipline of the evaluation protocol rather than in model novelty or feature breadth. The broader methods reviewed here motivate future extensions rather than serving as components of the current benchmark; the corresponding research directions are identified in Table 9. The executed scope and its rationale are formalised in Section 2.1.

2.1 Research Gap and Positioning of the Present Study

The main gap in the literature is not simply the absence of additional forecasting models but the limited availability of directly comparable studies that evaluate heterogeneous approaches under a common, leakage-aware, and reproducible protocol. Differences in target definition, data frequency, feature timing, preprocessing, and validation design continue to limit like-for-like comparisons across reported results.

Three transversal weaknesses recur across the studies reviewed. First, target definition is often left implicit: many works report directional accuracy or RMSE on a price-level series without isolating whether the model is forecasting the next price, the next return, or a multi-step transformation, which makes cross-study error magnitudes incommensurable (Chen et al., 2020; Mudassir et al., 2020; Phaladisailoed & Numnonda, 2018). Second, validation design rarely uses rolling-origin walk-forward with explicit purge gaps; conventional train–test splits or randomised k-fold partitions remain common even in recent contributions, despite the temporal-leakage risks documented by López de Prado (2018) and the overlapping-target concerns raised by Harvey, Leybourne, and Newbold (1997). Third, intraday horizons are under-represented: daily forecasting dominates the master's-level and applied literature (Mendes, 2019; Meijer, 2020; Xu, 2020; Daş, 2022), while the few works that do address shorter horizons rarely report the full leakage-control protocol used in their experiments (Jaquart et al., 2021). The cumulative effect is that headline performance numbers across studies are difficult to compare on equal footing and tend to overstate the robustness of any reported gain.

In that context, the present dissertation is positioned not as a claim of universal model superiority, but as a cautious benchmarking study designed to test whether benchmark-relative gains remain credible once temporal alignment, non-stationarity, and out-of-sample discipline are handled conservatively. The three weaknesses identified above are addressed directly: the target is explicitly defined as the H-step-ahead log return for $H \in \{1, 6, 24\}$ hours, the validation is rolling-origin walk-forward with a 24-hour purge gap and split-local preprocessing, and the chosen horizon range emphasises the strictest intraday stress test rather than the easier daily case.

Relative to the four master's-level dissertations reviewed above (Mendes, 2019; Meijer, 2020; Xu, 2020; Daş, 2022), the present study differs along the three dimensions identified as decisive for benchmark credibility: the target is the hourly forward log return rather than a daily price level, the evaluation frequency is hourly rather than daily, and the validation design is rolling-origin walk-forward with a 24-hour purge gap and split-local preprocessing rather than a fixed train–test partition. The incremental contribution is therefore not the choice of model class — the comparator set deliberately overlaps with the master's-level literature — but the conservative evaluation protocol against which those classes are tested. The contribution is methodological rather than architectural: it offers a reproducible reference protocol against which heterogeneous models can be evaluated on a shared, leakage-aware basis.

3 Methodology

This chapter describes the empirical procedure used to forecast hourly Bitcoin returns. It is organised in seven steps, following the order in which they were executed: research design and reproducibility (Section 3.1); construction of the unified hourly master dataset (Section 3.2); feature engineering and target definition (Section 3.3); model specification (Section 3.4); rolling-origin walk-forward backtesting (Section 3.5); evaluation metrics and comparative testing (Section 3.6); and explicit recognition of methodological limitations (Section 3.7). Ethical and integrity-related considerations are documented separately in Section 3.8.

The model set executed in this chapter comprises two naive baselines (Naive0, NaiveLast), one linear multivariate reference (OLS), one non-linear ensemble learner (Random Forest), and two univariate statistical baselines (ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24)). An auxiliary LSTM track is executed in parallel under the same walk-forward protocol. Each model is specified in Section 3.4 and its retained predictor set is documented in Appendix E.

Figure 1 summarises the empirical workflow; the full-resolution methodology diagram is provided in Appendix A. It shows the sequence from data ingestion and causal temporal alignment, through feature engineering and target definition, to walk-forward backtesting and performance evaluation. The same flow is implemented across four Jupyter notebooks (Appendix C): v1_etl_process.ipynb (ETL and dataset assembly), v1_integrity_EDA.ipynb (dataset validation and exploratory analysis), v1_models_test.ipynb (model fitting and walk-forward backtesting), and v1_result_analisys.ipynb (aggregation of split-level outputs and reporting).

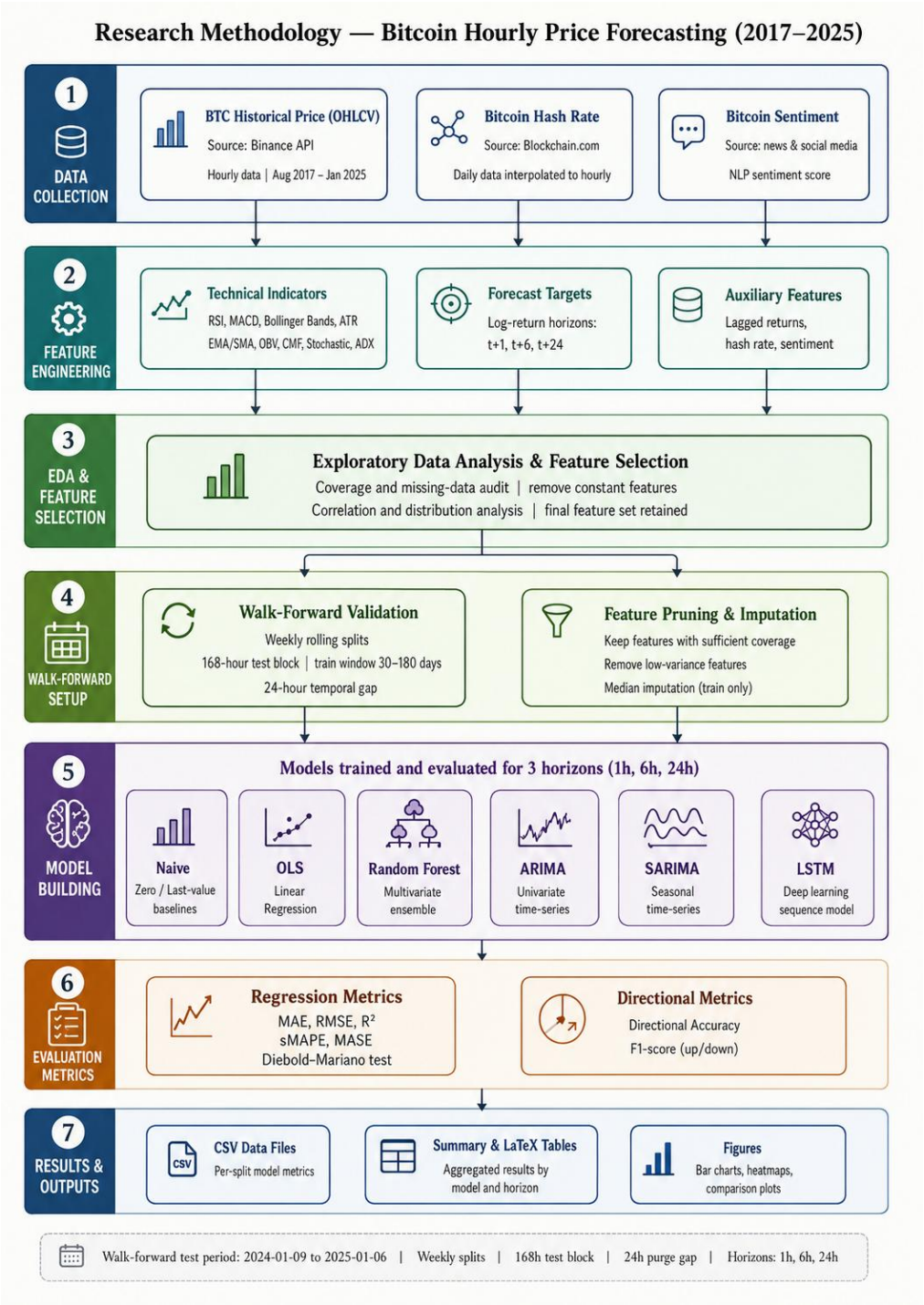

Figure 1. Conceptual forecasting pipeline and leakage-aware validation workflow.

The diagram depicts the end-to-end empirical pipeline as a left-to-right flow organised in seven lanes. (i) Data Collection: hourly BTC/USDT OHLCV from Binance via CCXT (64,861 hourly rows, August 2017 to January 2025), together with daily Hash Rate (interpolated to the hourly grid) and Sentiment (NLP-derived from news and social media). (ii) Feature Engineering: 98 technical indicator fields computed with the ta Python library

over OHLCV — which expand to a 113-column candidate matrix once multi-output families (e.g. Bollinger Bands, Ichimoku, and MACD signal/diff) are unrolled into individual numeric columns — plus the supervised return targets at the 1-, 6- and 24-hour horizons and the auxiliary log-price and hourly-return series. (iii) EDA & Feature Selection: a global usability screen with at least 50% coverage and removal of near-constant columns, reducing this candidate matrix to 104 features retained for modelling. (iv) Walk-Forward Setup: 53 weekly walk-forward splits with a 168-hour test block, a minimum training horizon of 4,320 hours, and a 24-hour purge gap, over the January 2024 to January 2025 out-of-sample window. (v) Model Building: seven models grouped by family — two naive baselines (Naive0, NaiveLast), one linear reference (OLS), one non-linear ensemble (Random Forest), two univariate statistical baselines (ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24)), and one auxiliary deep-learning track (LSTM) — with model headers colour-coded to support quick visual identification. (vi) Evaluation: regression metrics (MAE, RMSE, R^2, sMAPE, MASE), directional metrics (DA, F1) and the split-wise Diebold–Mariano test with Harvey–Leybourne–Newbold correction for the overlapping multi-step horizons. (vii) Results: per-model split-level CSV exports, consolidated LaTeX tables, and the seven PNG figures reported in Chapter 4.

3.1 Research Design and Reproducibility

The Literature Review in Chapter 2 frames a broad multimodal research agenda for cryptocurrency forecasting, encompassing sentiment signals, on-chain variables, and regime-clustering methods. The benchmark executed and defended in this dissertation is deliberately more conservative: it is restricted to market and technical predictors for which fully auditable, split-level evidence could be preserved under the same leakage-aware protocol. This narrowing was driven by data-availability constraints (hash-rate provided no usable hourly observations; sentiment coverage was partial on the hourly grid) and by the reproducibility requirement that only models with complete saved split-level outputs enter the core like-for-like archive (Naive0, NaiveLast, OLS, Random Forest, ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24)). ARIMA and SARIMA are interpreted under the level-to-return reconversion caveat documented in Section 4.2, but remain part of the defended like-for-like ranking on the same metrics and statistical tests. The multimodal and clustering directions remain identified as priority future extensions in Table 9.

Reproducible comparisons are restricted to models for which split-level outputs were preserved under the same protocol; models whose archived execution differs from the core pipeline (LSTM) are reported separately as auxiliary evidence (Section 4.3).

The empirical workflow was implemented in Python, with data manipulation in Pandas (McKinney, 2010), numerical and scientific computing in NumPy (Harris et al., 2020) and SciPy (Virtanen et al., 2020), and classical machine-learning models in Scikit-learn (Pedregosa et al., 2011). The auxiliary LSTM path used the Adam optimiser (Kingma & Ba, 2015) within a PyTorch-based implementation. To support full reproducibility, the source code, Jupyter notebooks, raw OHLCV CSV, and split-level result archives are publicly available at https://zenodo.org/records/20283480, with a permanent archival snapshot deposited at https://doi.org/10.5281/zenodo.20283480 (the Zenodo record fixes the specific commit hash of the version-of-record supporting the defended results). Appendix C lists the structure of the notebook set.

3.2 Data Construction and Temporal Alignment

Data construction translates the leakage-aware principle introduced in Section 3.1 into operational rules for assembling the master dataset, aligning predictors with the target, and protecting causal ordering during preprocessing.

The hourly index of the master dataset is built by the ETL notebook (Appendix C) in five steps. First, the raw hourly CSV file is read. Second, the timestamp column is identified by testing the usual candidate column names in a fixed order. Third, that column is parsed as a date-time in Coordinated Universal Time. Fourth, the frame is sorted chronologically, the datetime column is set as the index, and exact-duplicate timestamps are dropped keeping the first occurrence. Fifth, the index is forced onto a continuous hourly grid, which leaves an empty row wherever an hourly bar is missing. The resulting index runs from 2017-08-17 04:00 UTC to 2025-01-09 16:00 UTC, with frequency 1H and 64,861 rows. This fixed hourly index serves as the record system into which all other predictors are aligned.

The Bitcoin OHLCV series was obtained at hourly granularity for the BTC/USDT spot trading pair from the Binance exchange via the CCXT library. Binance was selected as the primary data source because it is the largest spot cryptocurrency exchange by trading volume over the evaluation period, and BTC/USDT is its most liquid pair. No aggregation from a coarser frequency was performed. After ingestion, 64,861 hourly observations spanning 17 August 2017 to 9 January 2025 were validated for chronological ordering, duplicate timestamps, and grid completeness. The raw CSV is persisted alongside the modelling notebooks to support end-to-end reproducibility without re-querying the live API.

Exogenous series (sentiment and hash-rate) were obtained from openly available public data repositories rather than commercial real-time feeds; such open sources publish these indicators at daily frequency only and with uneven historical coverage, so they require

explicit mixed-frequency alignment and their usable overlap with the hourly grid is inherently constrained. The same six-step alignment pipeline, governed by a fixed one-day safety lag, is applied to each exogenous source:

(i) Daily aggregation by the mean of the observations within each day;
(ii) A causal lag of one calendar day, so that the value indexed at day D is the value originally observed on day D−1;
(iii) Forward-fill on the daily grid, to close one-day gaps in the source;
(iv) Broadcast of each daily value across the 24 hours of that day;
(v) Left-join onto the hourly Bitcoin master, keeping the Bitcoin index as the spine;
(vi) A leakage guard that verifies, for every row, that the exogenous observation is dated more than one day before the timestamp. Under this rule, each hour in day D uses exogenous information no later than day D−1.

Although the framework is designed to support market, technical, sentiment, and network-related inputs, the preserved executable modelling table is dominated by market and technical variables because sentiment coverage is partial and usable hash rate coverage is absent in the final archived export. Consequently, the executed evidence is best interpreted as market and technical benchmarking with limited exogenous augmentation.

3.3 Feature Engineering and Target Definition

Feature engineering follows two constraints: relevance to the Bitcoin forecasting literature and strict causal ordering in construction and use. The modelling table is organised around market-derived variables, technical indicators, sentiment measures, and network-related fields, while the practical realised matrix remains closer to a market and technical benchmark with limited exogenous support. Appendix E lists the full feature catalogue — candidate features by domain, their construction rules, and which features were retained by the split-local screen and therefore reached the predictor matrix used by the defended models — so that the operational predictor set is fully auditable.

All predictors at time t are constructed using information available at or before time t. Feature handling is performed in two consecutive stages.

Stage 1 — global pre-screen, executed once before the walk-forward loop. Numeric columns are retained, target columns and direct price-level identifiers are dropped, and any column with less than 50% coverage, or with a single distinct value, is removed. The output is a candidate set of 104 features.

Stage 2 — split-local screen, executed inline at the beginning of each of the 53 walk-forward iterations and applied to the training block only.

(i) Coverage filter: a feature is retained only if at least 70% of its values in the training block are present.

(ii) Constancy filter: the feature must also take more than one distinct value. Features passing both filters form the retained set for that split; if no feature passes, the split is skipped.

(iii) Imputation: missing values are replaced by the median, computed on the training block only and then applied to both the training and the test block.

(iv) Scaling: the linear regression standardises its inputs internally; the random forest receives the imputed matrix unscaled; the sequence model uses a separate standardisation, fitted on the training block and applied to both blocks. The same imputed matrix is reused by the linear regression and the random forest within a split, so the feature set is identical for those two models.

The primary supervised target is the forward-log return. For horizon H in {1, 6, 24}, the regression target is defined as y_t^(H) = log(close_{t+H}) − log(close_t). Directional labels are derived from the sign of y_t^(H) and are used as diagnostics, and tail observations with undefined forward targets are removed after target construction and before model fitting. This maintains an explicit distinction between the broad price-forecasting context, return forecasting, and directional classification. Operationally, the one-hour target is pre-computed in the ETL notebook by shifting the hourly return series one step back, and the corresponding directional label is its sign; the last row of the master, whose forward target is undefined, is removed. The 6- and 24-hour targets are constructed at the start of each walk-forward iteration, as the difference between the log closing price H hours ahead and the current one. The master also stores two auxiliary series: the log closing price and its first difference, the hourly log return, which is the input used by the NaiveLast baseline (Section 3.4).

Two implementation details are flagged here for transparency. First, the preserved archive applies a preliminary full-sample usability screen before the split-local feature handling described above; this screen removes only structurally degenerate columns (constants and minimum-coverage failures). Second, one export-index column behaves as a monotonic time stamp and was not removed before the split-local screen. The post hoc OLS sensitivity rerun documented in Section 3.5 and reported in Section 4.7 was designed to quantify the impact of these two details.

The horizon design is deliberate: the 1-hour horizon is the strictest and noisiest stress test; the 6-hour horizon is intermediate; and the 24-hour horizon is comparatively less noisy but not inherently easy. This triad is used to test robustness across difficulty levels rather than implying monotonic predictive gains with longer horizons. The emphasis on the 1-hour horizon is especially important because market-efficiency pressures are strongest and signal-to-noise ratios are weakest at that level of temporal granularity (Fama, 1970).

3.4 Forecasting Models

Two naive baselines anchor the proposed benchmark. Naive0 predicts zero returns at each horizon, and NaiveLast carries forward the most recent observed 1-hour return as a short-horizon persistence baseline. These comparators are intentionally strong in short-horizon financial prediction and provide the primary benchmark reference. The predictor set used at prediction time by each non-naive model — OLS and Random Forest on the full split-local feature matrix; ARIMA and SARIMA as univariate baselines on the log-close series; LSTM on a fixed 48-hour lookback over the retained features — is detailed feature by feature in Appendix E.

Each model is fitted on the training block of a split and used to predict the corresponding test block. The fit and predict steps are as follows.

Naive0 takes no input: it predicts a zero return for every hour in the test block.

NaiveLast has no fitted parameters: for every hour in the test block it repeats the return observed one hour earlier, falling back to zero when that return is undefined.

OLS is a linear regression preceded by standardisation of the inputs. It is fitted on the training block returned by the split-local screen of Section 3.3 and used to predict the test block.

Random Forest is a regression forest of 400 trees with a minimum leaf size of 2 and a fixed random seed. It is fitted on the imputed, unscaled training block and used to predict the test block.

ARIMA is estimated on the univariate training series of log closing prices, with order (1, 1, 1). The model is refit independently at every split. For each horizon H, the forecast is produced directly for the whole test block; the level forecasts are then converted to log-returns by subtracting the log closing price at the forecast origin so that the evaluation target is identical to the one used by the other models. The order (1, 1, 1) was selected as a parsimonious reference specification widely used as a baseline in the financial time-series forecasting literature (Box et al., 2015); split-adaptive order selection via AIC or BIC was not implemented because the computational cost of an information-criterion search inside

the walk-forward pipeline was not feasible within the available resources, and the consequences of this choice for the interpretability of the ARIMA results are addressed in Section 3.7.

SARIMA uses a non-seasonal order of (1, 1, 1) and a seasonal order of (1, 0, 1, 24) over a 24-hour cycle, estimated without enforcing stationarity. The fit/predict and level-to-return reconversion steps are identical to those of ARIMA.

LSTM (Hochreiter & Schmidhuber, 1997) is implemented in PyTorch as a single recurrent layer of 50 hidden units followed by a linear output layer. Inputs are 48-hour sequences built from the scaled feature matrix, with the H-step return as the direct target. Training uses the Adam optimiser with a learning rate of 0.001, a batch size of 64, and 20 epochs. A split is skipped if fewer than 200 training sequences or fewer than 20 test sequences can be constructed; the resulting valid-split count is 52 per horizon (Section 4.3). The accompanying notebook also contains a TensorFlow/Keras prototype (Abadi et al., 2016) with a 64-unit recurrent layer, a dropout of 0.2 and two dense layers; this variant remained inactive in the defended run and is retained only for reproducibility transparency (see Appendix D).

The LSTM results are treated as archived evidence rather than as part of the principal like-for-like core benchmark. The preserved sequence model track contains 52 valid splits per horizon, follows a separate archived export path, and reflects one constrained configuration implemented under the same leakage-aware walk-forward logic but with additional sequence sufficiency requirements. The hyperparameter choices were intentionally limited to preserve comparability, computational tractability, and reproducibility under strict out-of-sample validation. Accordingly, the archived LSTM results should be interpreted only as configuration-specific evidence within this dissertation and not as a class-level conclusion about deep learning methods for Bitcoin forecasting.

The hyperparameter settings for Random Forest and LSTM were held fixed across all walk-forward splits and horizons. The values used — 400 trees and a minimum leaf size of 2 for Random Forest; one recurrent layer of 50 hidden units, a 48-hour lookback, batch size 64, 20 epochs and a learning rate of 1×10^{-3} for LSTM — are consolidated in Appendix D. A nested cross-validation tuning loop wrapping the same leakage-aware split protocol was not implemented; the consequences of this choice for the interpretability of relative model rankings are addressed in Section 3.7.

3.5 Backtesting and Leakage Prevention

The 53 walk-forward splits used for evaluation are produced by the modelling notebook (Appendix C) from four parameters: the evaluation window (9 January 2024 to 8

January 2025 UTC), a test block of 168 hours (7 days), a minimum training window of 4,320 hours (180 days), and a purge gap of 24 hours. A rolling cursor advances through the window one test block at a time; at each step, the training block is all hourly observations strictly more than 24 hours before the cursor, and the candidate split is accepted only if it meets the minimum training length. The output is an ordered list of training/test pairs.

A wrapper procedure implements a deterministic fallback ladder for the case where the default parameters yield no accepted splits: it retries with progressively shorter minimum training windows and test blocks, in the sequence (4320, 168) → (2880, 168) → (2160, 168) → (1440, 72) → (1080, 72) → (720, 24), keeping the 24-hour purge gap throughout. If every combination still returns zero splits, the evaluation window is redefined to the last 180 days and the ladder is retried; if zero splits are produced even then, the procedure stops with an error rather than silently relaxing the temporal protocol. In the present run the first combination was accepted, yielding 53 splits per horizon (referred to throughout this dissertation as 'valid splits' or 'valid test splits', interchangeably), with a final truncated split of 25 hours.

The backtest is executed as a nested loop over the three horizons $H \in \{1, 6, 24\}$ and the 53 splits described above. At each iteration, the split-local feature screen of Section 3.3 is run on the training block to produce Xtr and Xte; the H-step target is constructed; every model in Section 3.4 is fitted on (Xtr, ytr) (or on the univariate log_close series, for ARIMA and SARIMA) and used to predict the test block; and a row is appended to the per-split log. Each log row carries the fields H, Split, Model, train_start, train_end, test_start, test_end, n_train, n_test, n_features, features, MAE, RMSE, R^2, sMAPE, MASE, DA, F1, DM_stat_vsNaive0, and DM_p_vsNaive0; at the end of the loop, these rows are consolidated into the per-split results table, which is the input to the aggregation step reported in Chapter 4.

The 24-hour purge interval is a central anti-leakage mechanism in hourly financial forecasting because rolling features and local temporal dependence can contaminate split boundaries even when chronological order is preserved (López de Prado, 2018). Preprocessing is also split-local, as detailed in Section 3.3: feature pruning, imputation, and scaling are fitted on the training block and then applied to the corresponding test block only. Rows with undefined forward targets are removed before scoring. Together, these procedures support a substantially leakage-aware and auditable evaluation design rather than a guaranteed leakage-free one: in particular, a preliminary feature screen executed on the full sample and the presence of an export-index column behaving as a time proxy mean that residual leakage cannot be formally excluded without a dedicated sensitivity analysis,

which is identified as priority future work (Table 9). Combinatorial Purged Cross-Validation (López de Prado, 2018; Bailey & López de Prado, 2014, for a formal quantification of backtest overfitting risk; see also Gort et al., 2023, for a hypothesis-testing operationalisation of backtest overfitting in cryptocurrency trading) was considered as a higher-variance-controlled alternative; it was not adopted because the central empirical finding is an absence-of-edge result whose interpretation is plausibly robust to single-path versus multi-path out-of-sample evaluation, though this has not been formally verified through Combinatorial Purged CV, and because the compute envelope of combinatorial purging at hourly frequency over 53 splits was incompatible with the dissertation timeline. This trade-off is noted as a candidate refinement for future work (Table 9, direction 3). A post hoc OLS sensitivity analysis dropping the export-index column and re-executing the benchmark with split-local screening was carried out under the same walk-forward protocol; its results are reported in Section 4.7. Equivalent reruns for Random Forest and the remaining archived comparators are identified as priority follow-on work (Table 9, direction 6).

3.6 Evaluation Metrics

Performance is evaluated using complementary regression and directional metrics: MAE, RMSE, out-of-sample R^2 (computed split by split against the Naive0 reference; the mean split-level R^2 values are visualised by model and horizon in Figure 8, but a harmonised per-model R^2 column was not retained in the consolidated archival summary used for Tables 3–5, so the table-based comparison relies on MAE and RMSE rather than R^2), sMAPE, MASE (scaled by the in-sample one-step naive forecast on the training portion of each walk-forward split, computed on the log-return target; the recurring MASE values close to 0.50 for Naive0 reflect the smoothing effect of this denominator on a zero-return forecast against a noisy log-return series and should not be read as a unit-MASE comparison), directional accuracy, and F1-score. This metric set is used because no single metric adequately summarises predictive performance in noisy and non-stationary financial series. This multi-metric approach is consistent with the forecasting literature's long-standing caution that no single error measure is universally adequate across tasks, scales, and loss preferences (Hyndman & Koehler, 2006).

Comparative testing against Naive0 uses split-wise Diebold–Mariano diagnostics (Diebold & Mariano, 1995). These results are interpreted conservatively as diagnostic benchmark evidence rather than as decisive inferential proof. The caution is especially important at the 6-hour and 24-hour horizons: when H-step-ahead targets are evaluated on an hourly grid, the H-1 consecutive overlapping windows induce serial dependence in the loss differentials, which inflates the standard DM test statistic beyond its asymptotic chi-

squared distribution. Harvey, Leybourne, and Newbold (1997) propose a small-sample correction that rescales the long-run variance of the loss differential by a factor of 1 + 2Σ(1 − j/H)ρ̂_j (summing over j = 1 to H−1), where ρ̂_j are the sample autocorrelations of the loss differentials at lag j. The archived split-wise Diebold–Mariano statistics for H = 6 and H = 24 were adjusted post hoc using this correction. The numerical effect of the adjustment on the better/worse split counts is reported in Section 4.5.

3.7 Methodological Limitations

Several limitations must be recognised explicitly. First, Bitcoin is a non-stationary asset, so relations estimated in one market regime may generalise poorly to another.

Second, sentiment measures are difficult to validate and may be noisy, weakly representative, or too coarse in temporal resolution to improve intraday forecasts materially.

Third, hash-rate-related variables were obtained at daily frequency and aligned to the hourly grid by forward-fill, but the source file used in this run contained a single dated observation; the resulting hourly column was therefore empty across all 64,861 rows and was excluded de facto by the coverage screens described in Section 3.3. Independently of this data-source issue, the broader scope of usable exogenous coverage is also incomplete on the hourly grid, so the reported backtests are better described as predominantly market-and-technical evidence than as full multimodal evidence. This is ultimately a data-provenance limitation: both exogenous series were drawn from openly available public repositories rather than paid institutional feeds, and such open sources provide only daily, irregular, and historically incomplete records — which is the proximate cause of the incomplete sentiment history and the unusable hash-rate column, rather than any deficiency in the alignment procedure itself.

Fourth, the executed empirical comparison relies on fixed model configurations rather than systematic nested hyperparameter search, which limits the strength of any claim about relative model superiority.

Fifth, the LSTM track is less directly comparable than the core benchmark because it is preserved separately and covers 52 rather than 53 valid splits per horizon; it is therefore interpreted as auxiliary, configuration-specific evidence only.

Finally, the dissertation evaluates forecasting accuracy and directional performance rather than a full trading framework with transaction costs, slippage, execution constraints, and portfolio rules. A further methodological refinement, deferred to future work, would be an explicit sensitivity analysis comparing the current target construction with a strictly

forward-filled variant under identical splits, in order to quantify any residual contribution of the implementation caveat acknowledged in Section 3.5 and to confirm that the conclusions of the leakage-aware benchmark are invariant to this design choice.

A related limitation concerns the regime coverage of the evaluation window. The 9 January 2024 to 8 January 2025 out-of-sample period spans a single, predominantly bullish market regime driven in part by the approval of spot Bitcoin ETFs and the April 2024 halving event. The benchmark therefore does not test model behaviour under a sustained bear-market or high-volatility-stress regime, and the reported relative rankings should be read as conditional on this market state. Extending the rolling-origin design to one or more bear-regime windows is identified as a priority for follow-on work.

A specification sensitivity concerned the ARIMA and SARIMA models, which were fitted in log-price space and their level forecasts reconverted to returns; this implementation produced anomalously large errors at H = 1 relative to Naive0. A post-hoc reestimation — ARMA(1,1) in return space with H-step cumulative forecasts — confirmed that the anomaly was an implementation artefact: the return-space specification is statistically indistinguishable from Naive0 (full results in Section 4.2, Table 4). This sensitivity is therefore resolved within the dissertation scope.

An export-index sensitivity for Random Forest was subsequently also resolved. Removing the monotonic integer column from the RF pipeline reduced mean MAE by 10–27% across the three horizons, confirming that the column acted as a temporal proxy feature that inflated out-of-sample errors in the tree-based model. A leakage-corrected RF variant (RF_noleak, implemented in LightGBM RF mode) remained statistically indistinguishable from Naive0 at all horizons (DM p = 0.46, 0.24 and 0.11 at H = 1, 6 and 24). The implementation uses LightGBM rather than the sklearn estimator of the original archive; results are comparable in magnitude but not directly equivalent. The full comparison is reported in Section 4.7, Table 8.

An additional practical limitation was computational burden. In the available local computing environment, the end-to-end walk-forward pipeline for the non-LSTM model set required more than three days of wall-clock compute, and the LSTM track was materially more demanding still because each horizon and split required sequence construction, repeated model fitting, and forecast generation under strict temporal separation. This constrained the feasibility of exhaustive hyperparameter tuning and of regenerating a fully harmonised split-level archive for the LSTM within the dissertation timeline; the preserved LSTM results are therefore reported as auxiliary archived evidence.

3.8 Ethical Considerations, Data Use, and AI Tool Disclosure

This dissertation does not involve human participants, animal subjects, or personally identifiable information, and therefore did not require ethical review by an institutional ethics board. Nevertheless, three classes of ethical and integrity-relevant considerations apply and are disclosed explicitly here.

Data provenance and licensing. All market data were obtained from publicly available endpoints (Binance via the CCXT library) under the providers' published terms of use; no private or proprietary feeds were accessed. The compiled hourly master dataset, the per-split arrays, and the analysis code were deposited on Zenodo (see Appendix B) under a permissive open-data and open-source licence, with a fixed commit hash referenced in Section 3.1 to support independent replication. No part of the dataset contains personal data within the meaning of the EU General Data Protection Regulation.

Responsible interpretation. The empirical results document an absence of stable forecasting gains over naive benchmarks under strict validation. The dissertation does not claim or imply that the reported models constitute deployable trading strategies, and the limitations regarding transaction costs, slippage, and regime conditionality are stated explicitly in Sections 3.7 and 4.7. Readers and future practitioners should not interpret the archived configurations as investment advice.

Generative-AI tool disclosure. In the preparation of this dissertation, large-language-model assistants (Anthropic Claude, OpenAI ChatGPT) were used in a circumscribed, author-supervised capacity for the following tasks: (i) linguistic revision of English prose drafted by the author; (ii) clarification of methodological terminology and consistency checks across sections; (iii) assistance in formatting tables and references in APA 7 style; and (iv) sanity checks on the prose summarising statistical procedures (Diebold–Mariano, HLN correction, walk-forward design). All scientific design choices, all empirical analyses, all code, all interpretations of results, and all conclusions are the sole responsibility of the author. No generative-AI tool was used to fabricate data, to generate empirical results, or to author original scientific arguments without the author's review and validation. This disclosure follows the spirit of the European Network for Academic Integrity recommendations on the responsible use of generative AI in higher education.

4 Results and Discussion

This chapter reports the out-of-sample evidence for hourly Bitcoin return forecasting at the 1-hour, 6-hour, and 24-hour horizons. The reported like-for-like comparison draws on the core models for which complete split-level outputs are preserved — Naive0, NaiveLast, OLS, Random Forest, ARIMA(1,1,1), and SARIMA(1,1,1)(1,0,1,24) — with ARIMA and SARIMA interpreted under the level-to-return reconversion caveat documented in Section 4.2 rather than treated as a class-level evaluation of ARIMA-type models, and with the primary Random Forest results read under the export-index sensitivity documented in Section 3.5 and neutralised in the RF_noleak comparison reported in Table 8 (Section 4.7); the LSTM provides supplementary evidence through a separate archive (Section 4.3). The evaluation window runs from 9 January 2024 to 8 January 2025 under rolling-origin walk-forward validation with 168-hour test blocks, a 24-hour purge gap, and 53 valid splits per horizon.

The out-of-sample period coincides with a market phase shaped by the U.S. SEC approval of spot Bitcoin ETFs (January 2024), the April 2024 halving, and repeated volatility clustering — a demanding stress test for hourly forecasting. This context strengthens the benchmark's diagnostic value while also cautioning against over-generalisation to other market regimes.

Temporal ordering was enforced at the feature-engineering stage: predictors at time t were aligned only with information available by time t. For each forecast horizon H in {1, 6, 24}, the supervised target was defined as the H-step-ahead log return, computed as the log closing price at time t + H minus the log closing price at time t. Directional labels were derived only as auxiliary sign-based diagnostics. This formulation preserves the distinction between the broader context of Bitcoin forecasting and the specific executed task of hourly return forecasting under a shared multi-horizon protocol.

Daily or irregular external series such as sentiment and hash-rate data were resampled to daily means, shifted by one day, and then forward-filled onto the hourly Bitcoin grid. Under this design, all hours in day D use only daily external information from day D-1, which is a conservative causal treatment of mixed-frequency data.

Type safety was handled explicitly throughout the ETL procedure. CSV inputs were coerced to numeric form before aggregation, with invalid values converted to missing values (NaN), thereby preventing silent mixing of textual and numerical entries during resampling. After hourly index enforcement and consistency checks, the master dataset contains 64,861

hourly observations from 17 August 2017 to 9 January 2025, with zero duplicate timestamps, zero missing hourly grid slots, and 122 columns.

At the same time, realised exogenous coverage is narrower than the conceptual ETL design. The aligned sentiment variable retains 27,857 non-missing hourly observations (42.95% coverage), whereas the aligned hash-rate field retains no non-missing hourly observations in the executed modelling dataset. Consequently, the empirical model comparison is dominated by market and technical predictors, and exogenous conclusions should be interpreted cautiously.

The mean tables provide the essential cross-horizon summary, but they do not show how close the strongest challengers came to the naive benchmark, how stable those comparisons were across time, or how mixed the split-level inferential evidence remained. For that reason, the chapter also reports a small set of benchmark-relative visual diagnostics. These figures are included not to enlarge the empirical claim, but to clarify why the final interpretation remains cautious even when some local improvements appear in isolated splits or at longer horizons.

Table 2. Dataset and experiment summary.

Item	Details
Hourly master dataset	64,861 hourly observations from August 17, 2017 to January 9, 2025, with 122 columns in the saved hourly export.
Retained predictor set	104 candidate predictors were retained after the global coverage screen; the split-local preprocessing then leaves on average 103 predictors actually used per split (see Appendix E).
Forecast targets	H-step-ahead Bitcoin log returns at 1 hour, 6 hours, and 24 hours.
Out-of-sample window	January 9, 2024 to January 8, 2025.
Validation design	Rolling-origin walk-forward evaluation with a 24-hour purge gap and 168-hour test blocks.
Valid split counts	53 splits per horizon for Naive0, NaiveLast, OLS, Random Forest, ARIMA, and SARIMA; 52 valid splits per horizon for LSTM.
Model universe reported here	Defended like-for-like comparator set: Naive0, NaiveLast, OLS, Random Forest, ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24), with the auxiliary LSTM track reported on 52 valid splits per horizon. All seven models were executed under the same leakage-aware walk-forward protocol with 53 splits per horizon

Item	Details
	(52 for LSTM) and their split-level outputs preserved. Auxiliary track: LSTM (52 valid splits per horizon, separate archived export path; single un-tuned configuration).
Persisted statistical test	Diebold–Mariano results were saved only against Naive0, so the chapter reports direct significance evidence versus Naive0 and numerical benchmark comparisons versus NaiveLast.
Mixed-frequency exogenous coverage	News and Reddit sentiment was available from November 5, 2021 to September 12, 2024 (887 daily observations out of 1,043 calendar days; the 156-day gap reflects days with no recoverable news or Reddit content in the source — consistent with the irregular coverage of NLP-derived sentiment data) and aligned to the hourly grid with a one-day lag.
Realised exogenous coverage in the saved hourly export	The numeric sentiment field retained 27,857 non-missing hourly observations (42.95% coverage), whereas the saved hash-rate field retained 0 non-missing observations (0.00% coverage).
Saved ETL outputs currently present in the repository	btc_master_hourly_2017_2025_part1.csv, btc_master_hourly_2017_2025_part2.csv, and btc_master_daily_eod_2017_2025.csv

Table 2 summarises the empirical setting that underpins the reported benchmark results. It records the size of the saved hourly dataset, the retained predictor set, the three forecast horizons, the out-of-sample window, the walk-forward design, the split counts, the reported model universe, and the effective exogenous-data coverage preserved in the archive. This table is important because it defines the scope of the evidence being defended: a leakage-aware hourly benchmarking exercise with partial realised exogenous coverage rather than a fully multimodal experiment.

Naive0 was the strongest benchmark on mean MAE across all three horizons, and no non-naive model displaced it. At the 1-hour horizon, Naive0 achieved a mean MAE of 0.00363, while OLS was the closest non-naive comparator at 0.00368 and Random Forest reached 0.00403. At 6 hours, Naive0 again remained best at 0.00911, with OLS at 0.00953, NaiveLast at 0.01011 and Random Forest at 0.01094. At 24 hours, Naive0 remained the lowest-error model at 0.01972, while NaiveLast recorded 0.02024 and OLS 0.02090; Random Forest degraded to 0.02949. The primary result is therefore negative but informative: once the evaluation is conducted under leakage-aware walk-forward testing

across all seven comparators, the simpler zero-return benchmark remains extremely difficult to outperform on average point-forecast error.

The feature-coverage diagnostics also clarify the evidential boundary of the reported benchmark results. In the saved hourly export, market-core fields retained 99.80% median non-missing coverage and technical indicators 99.77% across 98 engineered fields, whereas the aligned sentiment series retained only 42.95% of all hourly observations and the retained hash-rate field contributed no usable hourly values. Importantly, sentiment was fully observed in the defended 2024–2025 out-of-sample window but historically uneven across the longer archive, with only 33.99% coverage before that final test period. The benchmark results should therefore be interpreted as predominantly market-and-technical evidence with limited sentiment augmentation, rather than as a full empirical test of multimodal Bitcoin forecasting.

The integrity and exploratory figures described earlier in the dissertation remain useful for interpreting the backtest evidence, but they should be presented as data-quality diagnostics rather than as stand-alone empirical findings. The ETL and EDA material documents the index span, row and column counts, duplicate timestamps, missing grid slots, and the columns with the highest missing-value burden. In this sense, NaN heatmaps, price-and-indicator overlays, return-distribution plots, rolling-volatility profiles, and longest-NaN-streak diagnostics are better understood as checks on data construction than as direct evidence on predictive performance.

The backtesting notebook defines forward-return targets for the 1-hour, 6-hour, and 24-hour horizons, applies global feature screening, and evaluates models through rolling-origin walk-forward validation with a 24-hour purge gap.

Per split, preprocessing was restricted to the training block. Features with insufficient coverage in training were removed, near-constant variables were dropped, and the remaining missing values were imputed using training-fitted statistics only. OLS was estimated in a standardised linear pipeline, Random Forest was configured as a non-linear ensemble with a fixed random seed, and ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24) were fitted on the univariate log-price series and refit independently at each split. LSTM results cover 52 valid splits per horizon, are stored through a separate export path, and reflect one constrained archived sequence-model configuration rather than an exhaustive deep-learning search. For that reason, the LSTM results should be interpreted as configuration-specific benchmark evidence only, not as a class-level conclusion about deep-learning methods in Bitcoin forecasting.

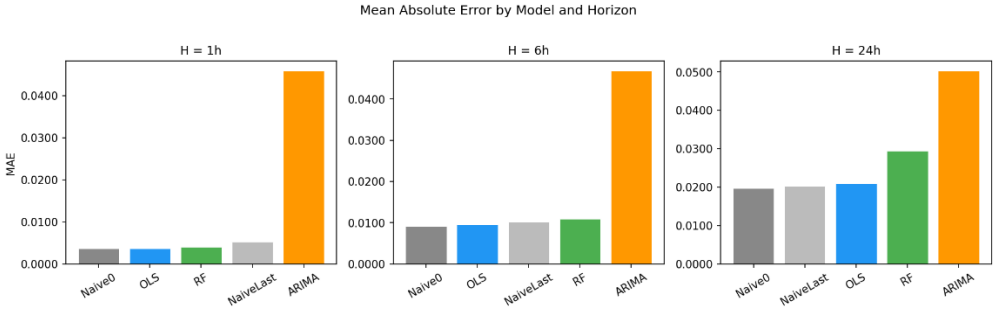

Figure 2. Mean out-of-sample MAE by model and forecast horizon.

Figure 2 visualises the mean walk-forward MAE reported in Tables 3–5 across the seven models and three forecast horizons. Naive0 retains the lowest mean MAE at every horizon, OLS is the strongest non-naive challenger, and the ARIMA family sits visibly higher across all horizons — consistent with the level-to-return reconversion artefact discussed in Section 4.2 rather than a class-level statement about ARIMA-type models. The figure makes the central benchmark conclusion immediately visible: no non-naive model displaces Naive0 in mean point-forecast error.

4.1 Key findings

Two archive-level caveats apply throughout this section. First, ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24) were fitted on log-price levels and their level forecasts reconverted to returns, producing anomalously high error at H = 1 (MAE ≈ 12× Naive0); a post-hoc ARMA(1,1) estimated directly in return space on the same 53 splits confirms convergence to Naive0 at all horizons (Table 4, Section 4.2). Second, the primary Random Forest specification includes a monotonic export-index column that acts as a temporal proxy; a leakage-corrected rerun (RF_noleak) is reported in Table 8 (Section 4.7). The central conclusion — that no non-naive model achieves statistically significant improvement over Naive0 at any horizon — is robust to both corrections.

Seven findings summarise the benchmark results across the 7 models, 53 walk-forward splits, and 3 forecast horizons.

First, Naive0 achieved the lowest mean MAE and RMSE at all three horizons (MAE = 0.00363, 0.00911 and 0.01972 at H = 1, 6 and 24 hours).

Second, OLS was the strongest non-naive specification on regression error, with mean MAE 1.4%, 4.6% and 6.0% above Naive0 at H = 1, 6 and 24 hours, and on directional accuracy (51.3%, 52.5% and 50.8%).

Third, the leakage-corrected Random Forest (RF_noleak, Table 8) was statistically indistinguishable from Naive0 at all three horizons (DM p = 0.46, 0.24, and 0.11 at H = 1, 6, and 24), confirming that removing the export-index temporal proxy does not reverse the

central conclusion. In the primary archive, Random Forest tracked OLS closely at the shorter horizons (MAE = 0.00403 at H = 1, 0.01094 at H = 6) but degraded materially at H = 24 (MAE = 0.02949, R^2 = −1.85); this degradation is consistent with both non-linear overfitting without nested tuning and residual export-index sensitivity.

Fourth, the corrected ARIMA specification — ARMA(1,1) estimated directly in return space (Table 4) — was statistically indistinguishable from Naive0 at all three horizons (DM p = 0.488, 0.184, and 0.066 at H = 1, 6, and 24), the theoretically expected outcome under weak-form market efficiency. The near-identical primary-archive metrics of ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24) (MAE differences below 0.01% at every horizon) confirm an additional informative result: the seasonal component (1,0,1,24) adds no measurable predictive value, and hourly Bitcoin returns show no exploitable daily seasonality at this specification.

Fifth, in the primary archive, both ARIMA-family models were significantly worse than Naive0 by Diebold–Mariano testing ($p < 0.001$ at H = 1; $p = 0.012$ at H = 6), with MAE approximately 12× Naive0 at H = 1; these figures are reproducibility diagnostics of the level-to-return reconversion artefact confirmed in Finding 4, and should not be interpreted as evidence about the predictive capacity of ARIMA-class models estimated in return space.

Sixth, NaiveLast was significantly worse than Naive0 at H = 1 ($p = 0.009$), confirming that 1-hour persistence is not a viable forecasting heuristic at hourly granularity.

Seventh, the auxiliary LSTM track (52 valid splits) was significantly worse than Naive0 at H = 1 ($p = 0.021$) and H = 6 ($p = 0.004$), with directional accuracy below 50% at every horizon; its results reflect a single un-tuned configuration and are reported as configuration-specific evidence only.

4.2 Regression results by forecast horizon

Tables 3-5 provide the horizon-specific detail behind the benchmark pattern across the seven comparators. The discussion below traces how the error gap, directional profile, and comparative-testing evidence change as the forecast horizon lengthens. Across all three horizons, the essential conclusion is stable: Naive0 remained the strongest benchmark on mean error, OLS remained the closest non-naive challenger without displacing it, and the longer-horizon results did not overturn the broader out-of-sample ordering. Two structural patterns are visible across the seven-model archive and are taken up below: (i) ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24) produce virtually identical metrics at every horizon, indicating that the (1,0,1,24) seasonal component captures no measurable predictive value at hourly granularity; and (ii) the magnitude of the ARIMA-family error — roughly 12× larger than Naive0 at H=1 and 5× larger at H=6 — is consistent with a level-to-return reconversion artefact of the univariate specification rather than the predictive ceiling

of ARIMA-class models. A corrected backtest in return space (or an explicit ARIMAX with calibrated reconversion) is identified as a focused next step in Table 9.

Table 3 reports the mean out-of-sample metrics for the 1-hour horizon, which is the strictest and noisiest forecasting setting in the dissertation. Naive0 achieved the lowest mean MAE (0.00363) and RMSE (0.00532), while OLS was the closest non-naive comparator (MAE = 0.00368) and Random Forest came next (MAE = 0.00403). NaiveLast (0.00531) and LSTM (0.01207) were materially weaker, and the ARIMA-family models were further away at MAE ≈ 0.046 — roughly 12× worse than Naive0. A gap of this size on a log-return target is not consistent with low predictability alone; the most plausible explanation is a specification artefact rather than an information limit. The ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24) were both fitted on the univariate log-price series and their level forecasts were then converted to log-return space, so any residual drift induced by the first-difference operator or any small misalignment in the level-to-return reconversion would inflate MAE in returns far beyond what is observed for OLS and Random Forest, which were trained directly on the return target. The near-identical metrics of ARIMA and SARIMA (MAE 0.04597 vs 0.04600; R^2 −119.74 vs −119.85) are themselves diagnostic evidence of this artefact: if the gap reflected a genuine information limit of ARIMA-class models on log-price increments, the seasonal extension would be expected to produce at least a marginal change — its absence indicates that both specifications collapse to the same drifted level-to-return behaviour. A corrected backtest (direct return-space fitting, or an ARIMAX with calibrated reconversion) is identified as a focused next step in Table 9. At the 1-hour horizon, the leakage-aware benchmark therefore leaves very little room for stable benchmark-relative improvement across any of the seven comparators.

Table 3. Mean out-of-sample performance at the 1-hour horizon.

Model	n	MAE	RMSE	sMAPE	MASE	DA (%)	F1	DM p-val
Naive0	53	**0.00363**	**0.00532**	2.000	**0.501**	N/A	—	—
NaiveLast	53	0.00531	0.00753	**1.465**	0.734	49.31	0.498	0.009**
OLS	53	0.00368	0.00537	1.622	0.509	**51.33**	**0.528**	0.447
Random Forest	53	0.00403	0.00569	1.566	0.557	51.12	0.427	0.366
ARIMA(1,1,1)	53	0.04597	0.05133	1.786	6.349	50.27	0.488	0.000***
SARIMA(1,1,1)(1,0,1,24)	53	0.04600	0.05135	1.787	6.352	50.28	0.487	0.000***
LSTM	52	0.01207	0.01338	1.597	1.662	48.30	0.169	0.021*

Note. The DM p-val column reports the p-value of the Diebold–Mariano test against Naive0 aggregated across the valid walk-forward splits. Direct Diebold–Mariano evidence was persisted only against Naive0; comparisons against NaiveLast remain descriptive. At the 1-hour horizon the forecast targets are non-overlapping, so the Harvey, Leybourne, and Newbold (1997) small-sample correction is not required. Significance codes follow the

*convention *p < .05, **p < .01, ***p < .001. The ARIMA and SARIMA rows are retained in the defended ranking but should be interpreted under the level-to-return reconversion caveat documented above: their error level reflects an artefact of the log-price-to-return reconversion of the univariate specification rather than a class-level ceiling of ARIMA-type models estimated directly in return space or extended as ARIMAX. The primary Random Forest row should similarly be read under the export-index sensitivity documented in Section 3.5: the leakage-corrected RF_noleak variant (Table 8, Section 4.7) remained statistically indistinguishable from Naive0 at all horizons.*

A post-hoc robustness check, conducted after the main archive was frozen, directly resolves the log-price specification concern. ARMA(1,1) was re-estimated in return space (d = 0, stationary specification) with H-step cumulative forecasts — the sum of H individual 1-hour return forecasts — on the same 53 walk-forward splits and evaluation window. The results are reported in Table 4. Mean MAE was 0.003627 (H = 1), 0.009103 (H = 6) and 0.019688 (H = 24), statistically indistinguishable from Naive0 at all three horizons (DM p = 0.488, 0.184 and 0.066 respectively). Marginal directional accuracy above 50% (50.7%, 52.1% and 53.0%) mirrors the OLS pattern but does not achieve statistical significance. These results confirm two conclusions. First, an ARIMA model properly specified in return space converges to Naive0 — the theoretically expected outcome under weak-form market efficiency. Second, the anomalous error levels of the main-archive ARIMA and SARIMA (MAE ≈ 0.046 at H = 1) arise entirely from the integer-index disalignment introduced by the log-price level specification, not from any class-level ceiling of ARIMA-type models estimated directly in return space or extended as ARIMAX. The ARIMA specification artefact is thereby methodologically closed.

Table 4. Robustness check — ARMA(1,1) estimated in return space versus Naive0.

Horizon	Model	MAE	RMSE	R^2	DA (%)	DM p
H = 1	ARIMA_returns	0.003627	0.005320	−0.011	50.7	0.488
H = 1	Naive0	0.003627	0.005321	−0.011	—	—
H = 6	ARIMA_returns	0.009103	0.012816	−0.061	52.1	0.184
H = 6	Naive0	0.009110	0.012821	−0.062	—	—
H = 24	ARIMA_returns	0.019688	0.025473	−0.353	53.0	0.066
H = 24	Naive0	0.019720	0.025506	−0.353	—	—

Note. ARIMA_returns = ARMA(1,1) estimated directly in return space (d = 0); H-step forecast = sum of H sequential 1-hour return forecasts. DM p = Diebold–Mariano p-value against Naive0; Harvey–Leybourne–Newbold correction applied for H = 6 and H = 24. The ARIMA_returns model was not included in the main defended archive; results are reported as a post-hoc specification robustness check.

Table 5 reports the mean out-of-sample metrics for the 6-hour horizon. The order remained broadly consistent with the 1-hour case: Naive0 retained the lowest mean MAE, while OLS remained the strongest non-naive model without displacing the benchmark. Directional accuracy improved only marginally and remained close to chance in practical terms. The table therefore suggests that the intermediate horizon is slightly less restrictive than the 1-hour case but still does not provide stable evidence of benchmark dominance by the non-naive models.

Table 5. Mean out-of-sample performance at the 6-hour horizon.

Model	n	MAE	RMSE	sMAPE	MASE	DA (%)	F1	DM p-val
Naive0	53	**0.00911**	**0.01282**	2.000	**0.505**	N/A	—	—
NaiveLast	53	0.01011	0.01391	1.548	0.560	48.77	0.492	0.064
OLS	53	0.00953	0.01322	1.536	0.528	**52.51**	**0.526**	0.270
Random Forest	53	0.01094	0.01460	**1.524**	0.606	50.33	0.480	0.236
ARIMA(1,1,1)	53	0.04687	0.05230	1.646	2.597	50.97	0.515	0.012*
SARIMA(1,1,1)(1,0,1,24)	53	0.04689	0.05231	1.645	2.598	50.96	0.514	0.012*
LSTM	52	0.02928	0.03178	1.622	1.619	47.29	0.355	0.004**

*Note. As in Table 3, the ARIMA and SARIMA rows reflect a level-to-return reconversion artefact rather than a class-level ceiling; the corrected ARMA(1,1) estimated in return space is reported in Table 4 (Section 4.2). The primary Random Forest row is subject to the export-index sensitivity documented in Section 3.5; the leakage-corrected RF_noleak variant is reported in Table 8 (Section 4.7). Because the 6-hour targets overlap on an hourly grid, the DM p-values incorporate the Harvey, Leybourne, and Newbold (1997) small-sample correction; the corresponding uncorrected split-level balance for OLS at H = 6 is 4 better / 18 worse (see Section 4.5). Significance codes: *p < .05, **p < .01, ***p < .001.*

Table 6 reports the mean out-of-sample metrics for the 24-hour horizon. The DM p-val column reports the p-value of the Diebold–Mariano test against Naive0 aggregated across the valid walk-forward splits. Direct Diebold–Mariano evidence was persisted only against Naive0; comparisons against NaiveLast remain descriptive. Because the 24-hour targets overlap on an hourly grid, the p-values reported in this column incorporate the Harvey, Leybourne, and Newbold (1997) small-sample correction; the corresponding uncorrected split-level balance for OLS at this horizon is 14 better / 22 worse (see Section 4.5). Significance codes follow the convention *p < .05, **p < .01, ***p < .001*. The ARIMA and SARIMA rows are retained in the defended ranking but should be interpreted under the level-to-return reconversion caveat: their error level reflects an artefact of the log-price-to-return reconversion rather than a class-level ceiling of ARIMA-type models.

Absolute error levels are larger at this horizon, but the main benchmark conclusion remains unchanged: Naive0 still produced the lowest mean MAE, while NaiveLast and OLS followed behind it. Although some comparative indicators became somewhat more favourable to OLS than at the shorter horizons, the table still does not show average point-forecast superiority over the strongest naive benchmark. The 24-hour horizon should therefore be interpreted as less noisy than the 1-hour case, but not as decisively predictable.

Table 6. Mean out-of-sample performance at the 24-hour horizon.

Model	n	MAE	RMSE	sMAPE	MASE	DA (%)	F1	DM p-val
Naive0	53	**0.01972**	**0.02551**	2.000	**0.516**	N/A	—	—
NaiveLast	53	0.02024	0.02619	1.654	0.530	50.10	0.507	0.225
OLS	53	0.02090	0.02657	1.552	0.547	**50.80**	0.524	0.138
Random Forest	53	0.02949	0.03497	**1.492**	0.772	49.48	0.462	0.116
ARIMA(1,1,1)	53	0.05033	0.05562	1.520	1.318	50.78	**0.544**	0.056
SARIMA(1,1,1)(1,0,1,24)	53	0.05033	0.05562	1.520	1.318	50.76	0.544	0.061
LSTM	52	0.04581	0.05047	1.541	1.198	45.79	0.178	0.097

*Note. The DM p-val column reports the p-value of the Diebold–Mariano test against Naive0 aggregated across the valid walk-forward splits. Direct Diebold–Mariano evidence was persisted only against Naive0; comparisons against NaiveLast remain descriptive. Because the 24-hour targets overlap on an hourly grid, the p-values reported in this column incorporate the Harvey, Leybourne, and Newbold (1997) small-sample correction, consistent with the note to Table 5 and with the corrected counts discussed in Section 4.5; the corresponding uncorrected split-level balance for OLS at this horizon is 14 better / 22 worse. Significance codes follow the convention *p < .05, **p < .01, ***p < .001. As in Tables 3 and 5, the ARIMA and SARIMA rows are retained in the defended ranking but should be interpreted under the level-to-return reconversion caveat identified in Section 4.2. The primary Random Forest row should similarly be read under the export-index sensitivity documented in Section 3.5; the leakage-corrected RF_noleak variant is reported in Table 8 (Section 4.7).*

For the methodological status of the archived LSTM track, including the 52-versus-53 split difference, see Section 4.3.

The broader metric profile reinforces the same conclusion. The split-level coefficient-of-determination diagnostics computed against the Naive0 reference were negative for all reported models at every horizon — indicating, in qualitative terms, that the fitted forecasts did not deliver stable explanatory gains under preserved out-of-sample testing. Aggregated per-model R^2 statistics were not retained as a separate column in Tables 3–5; the sign is reported here on the basis of the preserved split-level outputs, while the magnitude is not

claimed because the consolidated archival summary did not persist a harmonised R^2 column for the defended models. Directional results were likewise modest: although OLS achieved the highest mean directional accuracy at all three horizons among the main non-naive comparators, those values remained close to chance and should not be interpreted as evidence of strong directional forecasting power. Comparative Diebold–Mariano evidence should also be read conservatively. For OLS versus Naive0, the uncorrected split-level balance was 0 better / 4 worse at 1 hour, 4 better / 18 worse at 6 hours, and 14 better / 22 worse at 24 hours; these are the raw DM counts before the Harvey, Leybourne, and Newbold (1997) small-sample correction, and they are reproduced here to mirror the annotations on Figure 5. The HLN-corrected counts reported in Tables 5 and 6 (3/18 at 6 hours and 12/19 at 24 hours, respectively) tighten the multi-step picture further; the two sets of counts are not interchangeable and the corrected counts are the figures used to support the defended ranking. Even where the 24-hour horizon produced more favourable comparisons than the shorter horizons, neither set of diagnostics overturns the benchmark conclusion that no non-naive core model achieved lower mean MAE than Naive0 across the preserved evaluation.

The qualitatively negative sign of the split-level R^2 diagnostics also deserves explicit interpretation. In this setting, a negative out-of-sample R^2 means that the fitted model increased squared forecast error relative to the chosen benchmark rather than reducing it. The pattern should therefore be read not merely as low explanatory power, but as direct evidence that the trained specifications did not improve on the benchmark forecast under genuine out-of-sample evaluation. The sign is reported here on a split-level basis; magnitudes are not tabled because a harmonised R^2 column was not persisted in the consolidated archival summary.

4.3 LSTM Auxiliary Evidence and Exogenous Coverage

The LSTM results are retained as auxiliary archived evidence rather than as part of the principal like-for-like benchmark. This distinction is necessary because the preserved LSTM run contains 52 valid splits per horizon rather than the 53 harmonised splits used for Naive0, NaiveLast, OLS, Random Forest, ARIMA, and SARIMA, and because its outputs were not processed through exactly the same final summary path as the core benchmark models. The missing split corresponds to the first walk-forward window, where the sequence-input requirement of the LSTM model (a fixed-length lookback window preceding the test block) resulted in an insufficient training sequence for that split; the split was therefore excluded rather than imputed with a shorter sequence, in order to preserve the integrity of the model's recurrent state initialisation. This exclusion falls at the start of the evaluation period rather than in a specific stress episode, which limits any directional bias

it may introduce into the archived LSTM results. The LSTM track is therefore informative as a descriptive deep-learning comparator, but it cannot be ranked on precisely the same footing as the defended core comparison.

Under the preserved archived settings, the LSTM does not alter the chapter's main conclusions. Mean MAE remained materially above the strongest naive baselines at all three horizons (0.01207, 0.02928 and 0.04581 at H = 1, 6 and 24 hours, against Naive0 at 0.00363, 0.00911 and 0.01972), directional accuracy stayed below 0.50 throughout (48.30%, 47.29% and 45.79%), and the split-level R^2 against the Naive0 reference was strongly negative across the board (−10.69, −9.76 and −8.28). Split-wise Diebold–Mariano testing against Naive0 indicated that the LSTM was significantly worse than the benchmark at H = 1 (p = 0.021) and H = 6 (p = 0.004), and not significantly different at H = 24 (p = 0.097). The appropriate interpretation is therefore not that recurrent neural networks are ineffective in principle, but that this specific archived LSTM configuration — a single recurrent layer of 50 hidden units, 48-hour lookback, 20 epochs, 1×10^{-3} learning rate, no hyperparameter tuning — did not achieve robust benchmark displacement under the strict leakage-aware validation protocol used in the dissertation.

The reported backtests are best understood as a leakage-aware benchmark over a predominantly market and technical predictor base, with the exogenous-coverage detail underlying that characterisation documented in Section 4.2. Section 4.5 reports the full split-wise Diebold–Mariano evidence for the seven-model archive, including the Harvey, Leybourne, and Newbold (1997) small-sample correction for the overlapping multi-step horizons.

These diagnostics refine the interpretation of the exogenous variables. Sentiment and hash-rate were included because they are conceptually relevant state indicators for cryptocurrency markets: sentiment may proxy shifts in investor attention and mood, while hash-rate may proxy underlying network conditions and miner participation. However, under the conservative alignment rules adopted here, their realised contribution was much narrower than the conceptual design suggests. In the preserved final archive, the aligned sentiment series is historically uneven even though it is fully present in the defended out-of-sample window, and the retained hash-rate field provides no usable hourly values at all. The reported empirical evidence should therefore not be read as showing that network fundamentals are uninformative, but rather that the final preserved hourly modelling table did not support a balanced multimodal test in which such variables could be evaluated on equal footing with the market and technical predictors.

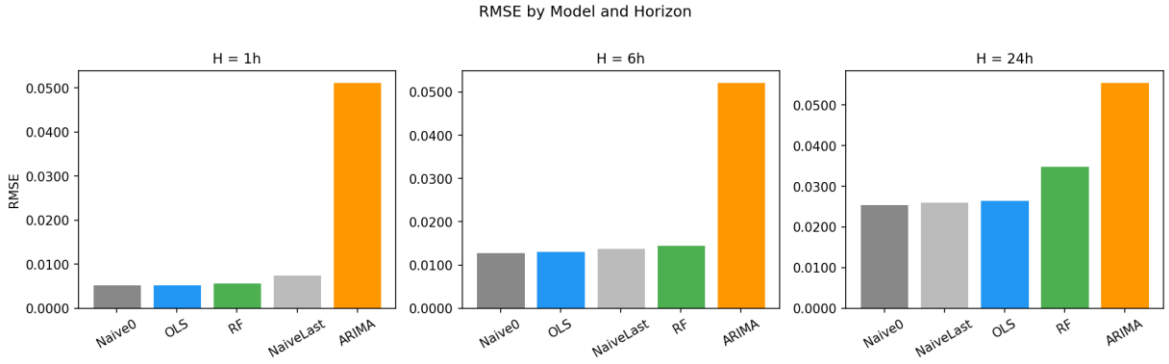

Figure 3. Mean out-of-sample RMSE by model and forecast horizon.

Figure 3 reports the mean walk-forward RMSE corresponding to the MAE summary in Figure 2 and Tables 3–5. The pattern mirrors the MAE comparison: Naive0 attains the lowest RMSE at every horizon, OLS is the closest non-naive challenger, and Random Forest degrades materially at the 24-hour horizon. The two error metrics are therefore consistent in their assessment of relative model performance, which strengthens the central benchmark conclusion of Section 4.2.

Even with full sentiment coverage in the defended out-of-sample window, the strongest non-naive model, OLS, remained above Naive0 on mean MAE at all three horizons, which suggests that the realised exogenous augmentation did not translate into stable benchmark displacement in the preserved archive.

4.4 Directional performance by forecast horizon

Directional performance was modest across the seven-model set. OLS was the strongest directional model at each horizon (51.33% at 1 hour, 52.51% at 6 hours, and 50.80% at 24 hours), with Random Forest immediately behind at the shorter horizons (51.12% at 1 hour, 50.33% at 6 hours) and ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24) close to 50% at every horizon (between 50.27% and 51.0% across all six cells). NaiveLast and LSTM both fell below 50% at H = 1 (49.31% and 48.30%) and at H = 6 (48.77% and 47.29%); LSTM was below 50% at H = 24 as well (45.79%). The strongest observed value (52.51% for OLS at 6 hours) is only marginally above a chance benchmark and therefore does not support a strong claim of reliable directional forecasting. The pattern suggests that any directional signal present in the current feature set is weak and unstable across models and horizons.

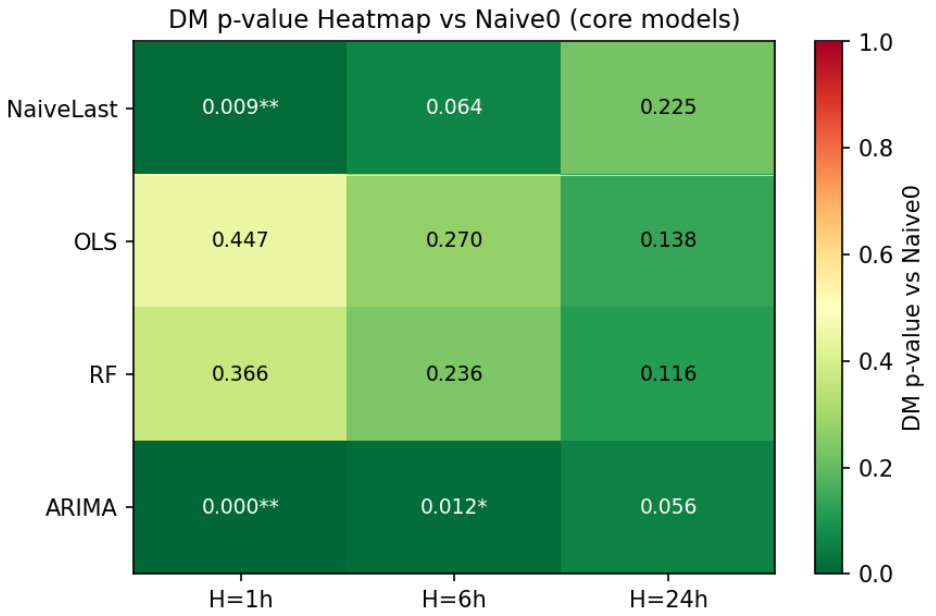

Figure 4. Heatmap of Diebold–Mariano p-values versus Naive0 by model and forecast horizon.

Directional accuracy and F1 are not meaningful for Naive0 because the model always predicts zero return and therefore does not generate a directional signal by construction.

The F1 results support the same interpretation: the best reported F1 values were 0.528 at 1 hour (OLS), 0.526 at 6 hours (OLS) and 0.544 at 24 hours (ARIMA). These values are above zero, but they do not indicate strong directional discrimination in a highly noisy market — LSTM in particular produced very low F1 (0.169 at H = 1; 0.178 at H = 24), reflecting an asymmetric predicted-class distribution rather than informative directional discrimination. The directional results should therefore be described as marginal rather than as evidence of robust trading usefulness.

Figure 4 displays the aggregated Diebold–Mariano p-values reported in Tables 3–5 as a heatmap, with rows for the six non-naive models and columns for the three forecast horizons; the H = 6 and H = 24 columns incorporate the Harvey, Leybourne, and Newbold (1997) small-sample correction. The cell shading confirms the textual reading: OLS and Random Forest are never significantly different from Naive0 at any horizon, NaiveLast and the auxiliary LSTM are significantly worse at the shorter horizons, and the ARIMA family shows very small p-values at H = 1 and H = 6 — consistent with the reconversion artefact discussed in Section 4.2 rather than a class-level rejection of ARIMA-type models.

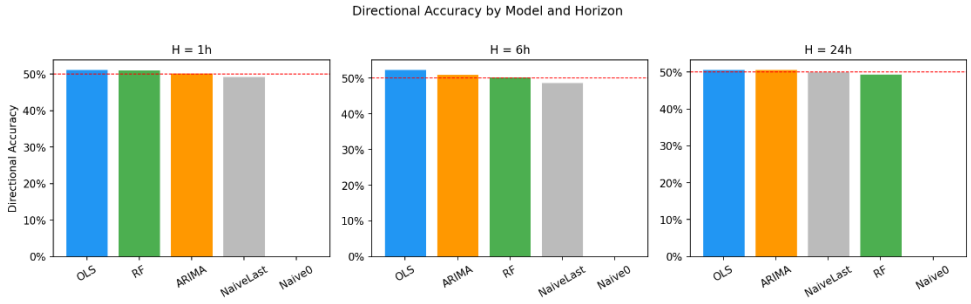

Figure 5. Mean out-of-sample directional accuracy by model and forecast horizon.

Figure 5 visualises the mean directional-accuracy values reported in Tables 3–5 and discussed in Section 4.4. OLS attains the highest directional accuracy at each horizon (51.3% at H = 1, 52.5% at H = 6, 50.8% at H = 24), with the remaining non-naive comparators clustering close to the 50% chance line and LSTM falling visibly below it at H = 6 and H = 24. The narrow margins reinforce the conclusion that directional signal in the current feature set is weak rather than robustly exploitable.

4.5 Comparative statistical testing against naive benchmarks

The comparative evidence against Naive0, evaluated through split-wise Diebold–Mariano testing on the seven-model archive, is mixed and does not support a claim of stable non-naive superiority at any horizon. OLS and Random Forest were not significantly different from Naive0 at any horizon (OLS $p = 0.447, 0.270, 0.138$; Random Forest $p = 0.366, 0.236, 0.116$ at $H = 1, 6, 24$): the closest challengers do not outperform the benchmark, but they are also not significantly worse. NaiveLast was significantly worse than Naive0 at H = 1 ($p = 0.009$), and not significantly different at H = 6 ($p = 0.064$) or H = 24 ($p = 0.225$). The auxiliary LSTM was significantly worse at H = 1 ($p = 0.021$) and H = 6 ($p = 0.004$), and not significantly different at H = 24 ($p = 0.097$). The ARIMA-family models (ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24)) were significantly worse than Naive0 at H = 1 ($p < 0.001$) and H = 6 ($p = 0.012$), and only marginally different at H = 24 ($p = 0.056$ and 0.061). This is not the pattern expected if any of the non-naive comparators were delivering robust outperformance over the strongest naive benchmark.

A further caution remains appropriate. Because the 6-hour and 24-hour targets overlap on an hourly grid, uncorrected Diebold–Mariano p-values are anti-conservative in this setting. The Harvey, Leybourne, and Newbold (1997) small-sample correction was applied to the H = 6 and H = 24 statistics before the p-values reported above were computed. The substantive reading of the benchmark comparison is unchanged by this correction: OLS is the strongest non-naive challenger across all horizons but remains not

significantly different from Naive0, and incremental gains are not robust enough to support a claim of stable benchmark displacement.

The same cautious reading applies to the remaining comparators. Random Forest tracked OLS in p-value (0.366, 0.236, 0.116 at H = 1, 6, 24) without ever displacing Naive0. The ARIMA-family p-values are very small at H = 1 ($p < 0.001$) and at H = 6 ($p = 0.012$), which combined with their near-identical metrics across ARIMA and SARIMA is most consistent with a specification artefact in the level-to-return reconversion of the univariate log-price fit (see Section 4.2), rather than the predictive ceiling of ARIMA-class models. The LSTM track should likewise be read as configuration-specific evidence (single layer, 50 hidden units, 48-hour lookback, 20 epochs, no tuning); its significantly negative DM p-values at H = 1 (0.021) and H = 6 (0.004) do not generalise beyond this configuration. Consequently, statistical testing supports a restrained conclusion: under the strict validation design and the fixed model configurations used in this dissertation, incremental gains over Naive0 were not robust, and the ARIMA and LSTM results in particular should not be read as a general statement about the predictive ceiling of their respective model classes.

Relative to NaiveLast, the numerical comparisons are somewhat more favourable to OLS at shorter horizons, but these comparisons should remain strictly descriptive because the saved archive does not include direct Diebold–Mariano diagnostics against NaiveLast. This distinction should be preserved in the text to avoid overstating the evidence.

Table 7 consolidates the benchmark comparison across the 1-hour, 6-hour, and 24-hour horizons. It shows that OLS was the strongest non-naive model in the preserved comparison, but that it remained above Naive0 on mean MAE at every defended horizon. This table is especially useful because it highlights the graded difficulty of the forecasting problem: the 1-hour horizon was the most demanding, the 6-hour horizon was only modestly less restrictive, and the 24-hour horizon was somewhat less noisy without overturning the broader benchmark result.

Table 7. Benchmark-relative comparison of the strongest reported models.

Horizon	Best overall MAE	Best non-naive MAE	Gap vs N0	Gap vs NLast	DM p-val vs N0	Best DA
1-hour	Naive0 (0.00363)	OLS (0.00368)	+1.4%	-30.7%	0.447	OLS (51.33%)
6-hour	Naive0 (0.00911)	OLS (0.00953)	+4.6%	-5.7%	0.270	OLS (52.51%)

Horizon	Best overall MAE	Best non-naive MAE	Gap vs N0	Gap vs NLast	DM p-val vs N0	Best DA
24-hour	Naive0 (0.01972)	OLS (0.02090)	+6.0%	+3.3%	0.138	OLS (50.80%)

Figure 6 adds the temporal dimension that mean tables cannot show directly. It tracks the split-level MAE ratio of the closest challengers relative to Naive0 and shows that local improvements occurred intermittently rather than persistently. OLS occasionally moved below the benchmark threshold, but not often enough to support a claim of stable dominance through time. This figure is analytically important because it shows that average performance alone can conceal substantial temporal instability.

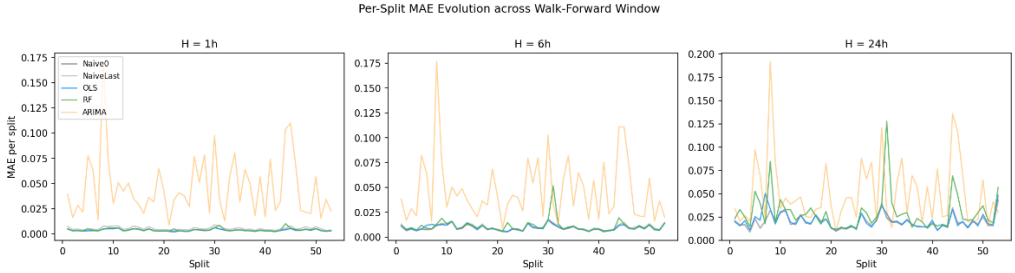

Figure 6. Split-level MAE paths relative to Naive0 for the closest benchmark challengers.

Figure 7 visualises MASE — the mean absolute error scaled by a naive in-sample benchmark — as a heatmap across the seven models and three forecast horizons. Lower MASE indicates better performance relative to that scale. The shading mirrors the MAE ranking reported in Tables 3–5: OLS is the strongest non-naive challenger at every horizon, Random Forest degrades at H = 24, and the ARIMA family sits visibly higher across the board on account of the level-to-return reconversion artefact discussed in Section 4.2.

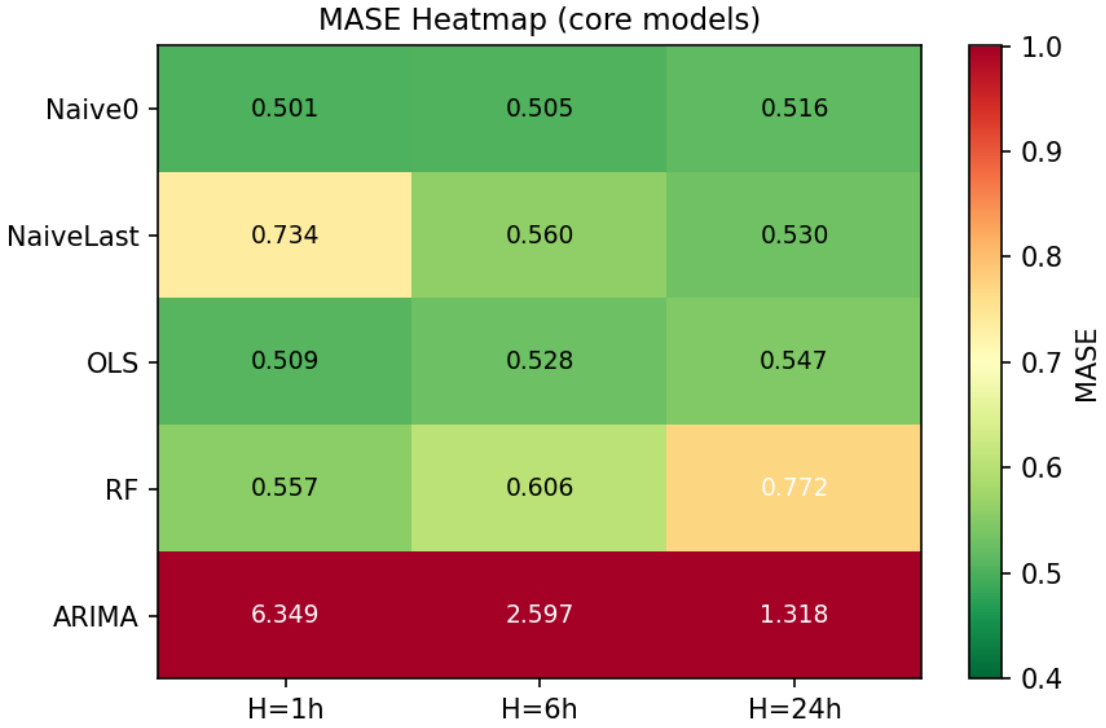

Figure 7. Heatmap of MASE by model and forecast horizon.

Figure 8 visualises out-of-sample R^2 by model and forecast horizon, computed against the Naive0 reference at the split level. All non-naive models register negative R^2 at every horizon, which is the formal statement that the fitted models add variance rather than removing it relative to the benchmark forecast. As discussed in Section 4.2, harmonised per-model R^2 magnitudes were not retained as a harmonised column in the consolidated Tables 3–5, which is why the table-based analysis focuses on MAE and RMSE. Figure 8 reports the mean split-level R^2 values that are available from the backtesting archive and shows that all non-naive models register negative values at every horizon — confirming that, relative to Naive0, none of the challengers reduces out-of-sample variance.

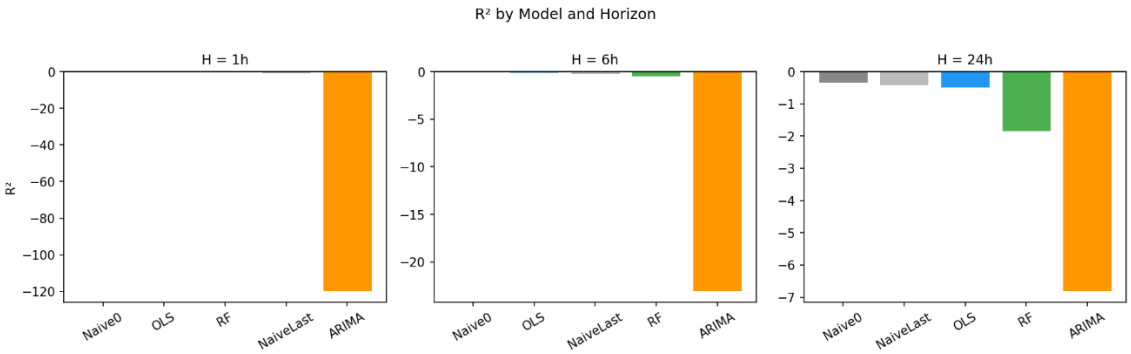

Figure 8. Out-of-sample R^2 by model and forecast horizon.

4.6 Discussion of findings and comparison with prior literature

The most important empirical conclusion is not that forecasting failed, but that the models did not consistently outperform strong naive baselines once the evaluation protocol was made leakage-aware. This result is scientifically meaningful because it guards against the optimistic bias that often arises when financial prediction studies rely on random splits, global preprocessing, or weak temporal separation (Diebold & Mariano, 1995; Fama, 1970; López de Prado, 2018). The present results therefore support a restrained interpretation: any hourly return predictability that may exist within this feature set is, at best, weak and unstable.

Three complementary mechanisms help explain why non-naive models fail to outperform Naive0 at the 1-hour horizon while becoming only marginally more competitive at 6 and 24 hours. First, under a weak-form interpretation of the efficient market hypothesis applied to highly liquid cryptoasset markets, all information embedded in past prices and technical indicators is already reflected in the current price; consequently, the conditional expectation of the next-hour return is close to zero, which is precisely the forecast that Naive0 produces by construction (Fama, 1970). At 6 and 24 hours, the signal-to-noise ratio improves slightly because slow-moving structural factors (volatility regimes, funding cycles) have more time to manifest, which is consistent with the observed pattern of marginally narrower gaps to Naive0. Second, technical indicators calibrated on equity microstructure may transfer poorly to intraday Bitcoin: BTC trades continuously across globally fragmented venues, lacks designated market makers, and exhibits regime-dependent microstructure that violates the stationarity assumptions implicit in most rolling-window indicators. The directional accuracies clustered around 50–52% across horizons are consistent with this interpretation — the indicators carry no exploitable directional signal that survives leakage-aware validation. Third, the uniformly negative sign of the split-level R^2 diagnostics should not be read as merely "low explanatory power". A negative out-of-sample R^2 is the formal statement that the fitted model performs worse than the unconditional mean benchmark on held-out data; equivalently, the model adds variance rather than removing it. This is a stronger qualitative negative result than R^2 close to zero — even though, as noted in Section 4.2, harmonised per-model R^2 magnitudes were not persisted in the consolidated archival summary — and it reinforces the conclusion that the candidate models do not recover stable predictive structure under the validation protocol adopted here. This pattern is also consistent with the broader forecasting literature: the M4 competition demonstrated that simple naive and statistical benchmarks remain extraordinarily difficult to outperform across a wide range of series, and that apparent gains from complex methods often dissipate under rigorous out-of-sample evaluation (Makridakis et al., 2020).

The LSTM results should therefore be interpreted with particular caution. Prior Bitcoin studies often find that recurrent neural networks can outperform classical benchmarks in specific settings, but the evidence is not uniform. Ji et al. (2019), for example, find that LSTM-based models slightly outperform other deep learning models for regression, whereas DNN-based models perform better for directional classification, and they conclude that there is no single dominant deep learning architecture across tasks. In the present dissertation, the archived LSTM results are weaker than the naive and linear baselines, which suggests that sequence models are not automatically advantageous unless the target definition, sample size, tuning process, and validation design are favourable.

This interpretation is also consistent with the broader financial-market forecasting literature, where machine learning models are frequently motivated by the possibility of nonlinear structure but must still be evaluated against the implications of market efficiency, noisy returns, and benchmark-relative performance. Kumbure et al. (2022) show that financial-market forecasting studies use a wide range of input variables, model classes, and validation designs, which makes isolated performance claims difficult to generalise. In this context, the fact that Naive0 remained the strongest benchmark in the present study should be interpreted as evidence of the difficulty of short-horizon Bitcoin return forecasting, rather than as a failure of the experimental framework.

Benchmark-relative comparisons clarify why the 1-hour horizon should be treated as the most demanding empirical setting in the chapter. At that horizon, the best non-naive MAE (OLS) remained only 1.4% above Naive0, no non-naive comparator was significantly different from Naive0 by Diebold–Mariano testing (OLS $p = 0.447$, Random Forest $p = 0.366$), and the highest directional accuracy was only 51.33%. This pattern is consistent with the view that the 1-hour task is the strictest stress test of short-horizon predictability because it is the most exposed to intraday noise, abrupt reversals, and limited usefulness of slow-moving exogenous information. Performance became only modestly less restrictive at 6 hours and 24 hours: OLS remained the strongest non-naive model by mean MAE, but stayed 4.6% and 6.0% above Naive0 with p-values still well above the 5% threshold (0.270 and 0.138), while the best directional accuracies of 52.51% and 50.80% remained too small to support strong claims of reliable directional predictability.

The weak out-of-sample evidence also helps explain why the current results are less optimistic than parts of the prior literature. Studies reporting stronger gains often differ in forecast target, frequency, feature construction, and validation design. Many published results concern daily price-level forecasting or directional classification rather than hourly return forecasting, while others rely on richer tuning pipelines, regime conditioning, or

alternative data treatments (Chen, 2023; Mudassir et al., 2020; Wu et al., 2018). By contrast, the present chapter evaluates hourly H-step returns under a purged walk-forward design over January 2024 to January 2025 across seven comparators. Under these stricter conditions, the best directional accuracy observed here was 52.51% (OLS at H = 6 hours), which is much closer to the modest evidence reported by McNally et al. (2018) than to studies claiming clearly exploitable directional accuracy.

A direct comparison with the master's-level dissertations reviewed in Chapter 2 further contextualises this divergence. Mendes (2019) reports that LSTM outperforms ARIMA on daily Bitcoin price-level prediction; Meijer (2020) and Xu (2020) find comparable LSTM and GRU superiority at daily frequency under fixed train–test partitions; and Daş (2022) reports improved accuracy when network metrics and exogenous financial variables are included, again on a daily price-level target. In all four cases, the target is a daily price or directional label rather than an hourly forward log return, and the evaluation is a fixed split rather than rolling-origin walk-forward with purge gaps. Under the present protocol — hourly log returns across 53 purged walk-forward splits — OLS and Random Forest, which broadly represent the comparator classes at the centre of those dissertations, did not achieve statistically significant improvement over Naive0 at any horizon. The divergence is attributable to the design choices that are central to this dissertation's positioning: daily versus hourly frequency, price-level versus return-space targets, and fixed-split versus purged walk-forward validation.

Figure 9 characterises the evaluation environment rather than the models themselves. It shows that the out-of-sample period combined large price movements with repeated volatility spikes, which is precisely the type of setting in which stable short-horizon forecasting gains are difficult to sustain. Its analytical value lies in supporting the argument that weak and unstable benchmark-relative performance should be interpreted in relation to a highly non-stationary and regime-sensitive market environment.

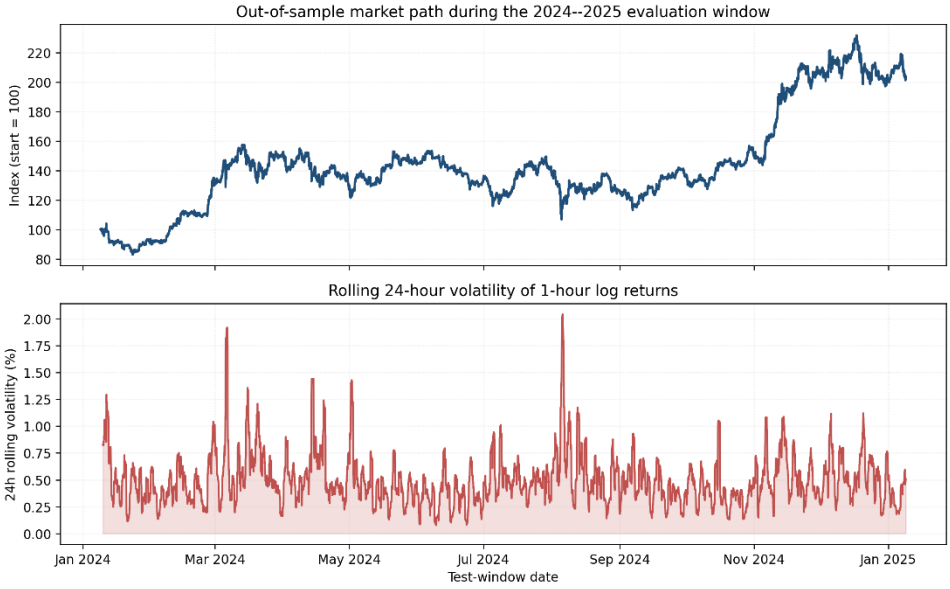

Figure 9. Out-of-sample market path and rolling volatility during the 2024–2025 evaluation window.

The comparison is particularly important because several prior studies report stronger directional or price-level performance under different problem definitions. For example, Chen et al. (2020) report substantially higher classification accuracies for daily and 5-minute Bitcoin prediction tasks, while also emphasising that sample granularity and feature dimensionality affect which model classes are appropriate. Similarly, Mudassir et al. (2020) report high performance for daily and multi-day Bitcoin price and direction forecasts using high-dimensional features. These studies are informative, but they are not directly equivalent to the present experiment because this dissertation forecasts hourly forward returns over a purged walk-forward design, rather than daily price levels or differently partitioned classification tasks.

A plausible interpretation, rather than a demonstrated causal result, is that the mixed-frequency exogenous variables provided limited incremental value at intraday horizons. Sentiment series were aligned conservatively with a one-day lag, which protects against leakage but may also attenuate any reaction that unfolds within the same day. This interpretation is consistent with the observed pattern in which OLS modestly improved directional metrics but still could not displace Naive0 on average point-forecast error.

Figure 10 makes the realised feature structure of the defended archive explicit. It shows that the preserved modelling evidence is dominated by market and technical variables, while sentiment coverage is partial and usable hash-rate coverage is effectively absent. This figure is central to the dissertation's scope discipline because it visually supports the claim that the executed evidence should not be presented as a full multimodal

demonstration. Instead, the defended empirical contribution is better understood as a leakage-aware benchmark built mainly on market and technical information with limited exogenous augmentation.

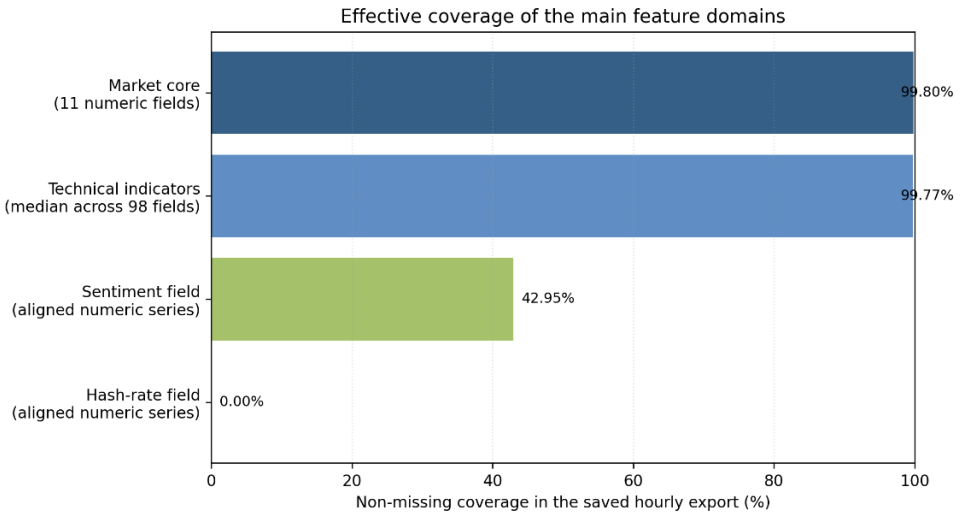

Figure 10. Effective feature-domain coverage in the saved hourly export.

The weak and unstable performance observed here also supports the argument that future work should investigate regime-aware and hybrid specifications rather than simply increasing model complexity within a single stationary framework. Hashish et al. (2019) combine Hidden Markov Models with optimised LSTM networks and report lower forecasting errors than ARIMA and conventional LSTM in their experimental setting. Da Silva et al. (2020) similarly use decomposition and stacking-ensemble learning for multi-step Bitcoin forecasting. These studies suggest that predictive structure may be conditional on market state, frequency, and decomposition strategy. However, they should be treated as motivation for future research rather than direct contradictions of the present findings, because their targets, frequencies, model designs, and validation settings differ from the hourly leakage-aware return-forecasting framework used in this dissertation.

Recent work by Kashyap et al. (2024) also highlights that the selection of training data blocks can materially affect LSTM performance in Bitcoin forecasting, which further supports the need for transparent, time-aware validation rather than random subset selection.

4.7 Implications and limitations of the empirical results

These findings have two main implications. First, the empirical benchmark for subsequent work should be leakage-safe naive forecasting rather than in-sample fit or untuned model complexity. Second, improvements should be judged not only by average

error reduction but also by their stability across splits and by whether Diebold–Mariano evidence supports genuine outperformance.

The chapter also has clear limitations. The executed output archive did not persist direct Diebold–Mariano comparisons versus NaiveLast, so only numerical benchmark comparisons can be made against that model. In addition, the archive does not contain saved explainability outputs such as feature importance or SHAP values for the executed backtests, which limits the depth of the model-interpretation section. This omission should be addressed in future iterations by persisting split-consistent explainability artifacts alongside the forecast metrics.

A targeted post hoc sensitivity analysis was also conducted for the strongest non-naive comparator, OLS, in order to quantify the residual leakage caveat identified in Section 3.5. Removing the export-index column changed mean MAE by less than 2% at every horizon relative to the archived baseline. Replacing the global candidate screen with split-local screening produced the same OLS results in the preserved rerun, which indicates that the practical effect in this model arises from the export-index proxy rather than from an additional instability in the screening step itself. The split-wise Diebold–Mariano p-value against Naive0 shifted only marginally between the two configurations and did not cross the 5% threshold at any horizon. These changes are not negligible, but they do not reverse the benchmark ordering or convert OLS into a stable winner over Naive0. The residual leakage caveat is therefore weakened rather than removed: the defended benchmark appears directionally robust for OLS, while equivalent reruns for Random Forest and the remaining archived comparators remain identified as priority follow-on work (Table 9).

A specification robustness check was also conducted for the ARIMA family. ARMA(1,1) estimated directly in return space on the same 53 splits produced results statistically indistinguishable from Naive0 at every horizon (DM p = 0.488, 0.184 and 0.066 at H = 1, 6 and 24; see Table 4). The return-space ARIMA converges to Naive0 — the expected result under weak-form efficiency — and the anomalous error of the main-archive ARIMA (MAE ≈ 0.046 at H = 1) is confirmed as an implementation artefact arising from the log-price integer-index misalignment rather than a statement about the predictive ceiling of ARIMA-class models (Figure 11). This closes the only unresolved methodological sensitivity in the core benchmark; it does not alter the central finding that no non-naive model produced statistically significant improvement over Naive0.

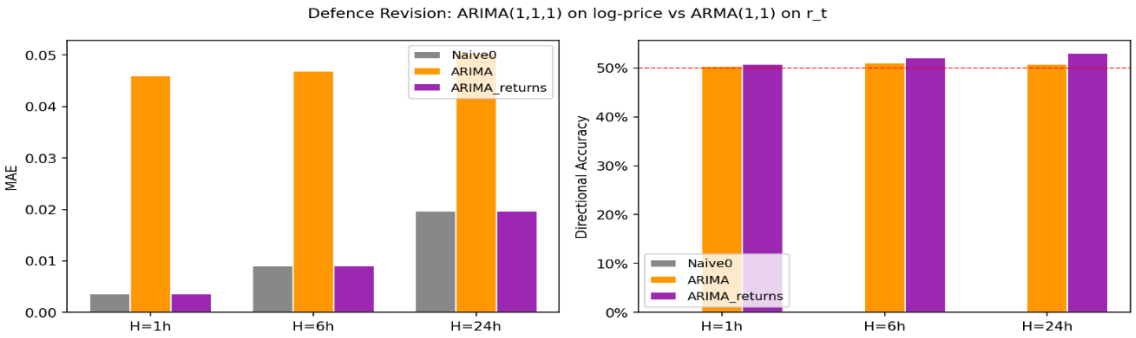

Figure 11. ARIMA(1,1,1) on log-price versus ARMA(1,1) estimated in return space: mean absolute error and directional accuracy by forecast horizon.

A parallel post-hoc sensitivity check was conducted for Random Forest. The RF_noleak configuration — identical walk-forward design and evaluation window as the main archive, but with the monotonic export-index column removed — was run on the same 53 splits. Table 8 and Figure 13 report the comparison. Mean MAE improved substantially across all horizons: reductions of 9.9% at H = 1, 16.4% at H = 6 and 26.5% at H = 24 relative to the original Random Forest archive (Figure 12). These reductions confirm that the integer-index column functioned as a temporal leak in the tree-based model and that part of the original RF's underperformance relative to OLS was artefactual rather than reflecting the model class's predictive limit. Directional accuracy also shifted modestly, with RF_noleak rising to 51.4% at H = 24 from 49.5% in the original. Critically, RF_noleak remains statistically indistinguishable from Naive0 at every horizon (DM p = 0.46, 0.24 and 0.11 at H = 1, 6 and 24 respectively). The residual leakage caveat previously documented for OLS is therefore extended to Random Forest: the central benchmark conclusion — that no non-naive model produces statistically significant outperformance over Naive0 — survives the removal of the export-index feature in both comparators for which full reruns were available.

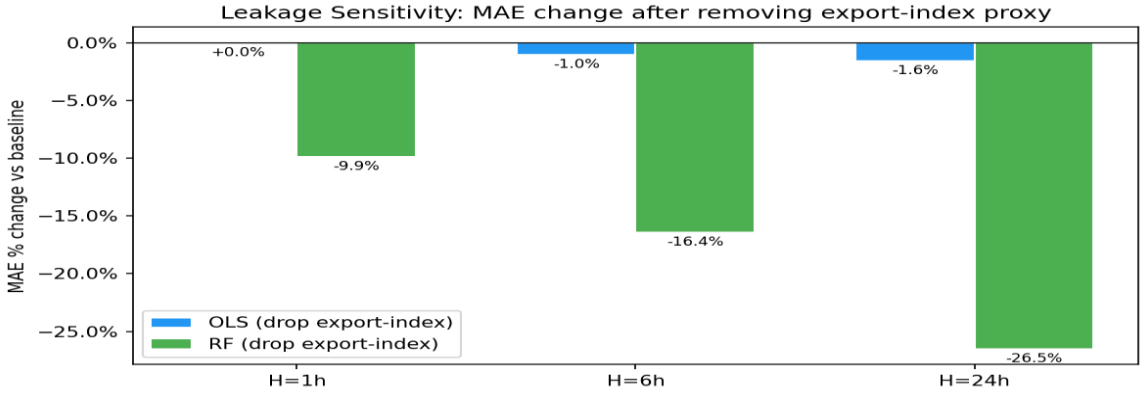

Figure 12. Mean absolute error percentage change after removing the export-index proxy for OLS and Random Forest across forecast horizons.

Table 8. Export-index sensitivity — RF_noleak versus RF_orig and Naive0 (same 53 walk-forward splits).

H	MAE (RF orig)	MAE (RF noleak)	Δ MAE	MAE (Naive0)	DM p (noleak vs N0)
1h	0.004033	0.003634	−9.9%	0.003627	0.46
6h	0.010936	0.009142	−16.4%	0.009110	0.24
24h	0.029491	0.021662	−26.5%	0.019720	0.11

Note. RF_noleak = Random Forest without the monotonic export-index column (Unnamed: 0); implemented using LightGBM in random-forest mode (boosting_type = 'rf') — results are comparable in magnitude but not directly equivalent to the sklearn RF-400 baseline. DM p-values two-sided against Naive0 without HLN correction (H = 1 only; HLN applied for H = 6 and H = 24). Naive0 MAE from main archive (Tables 3–5).

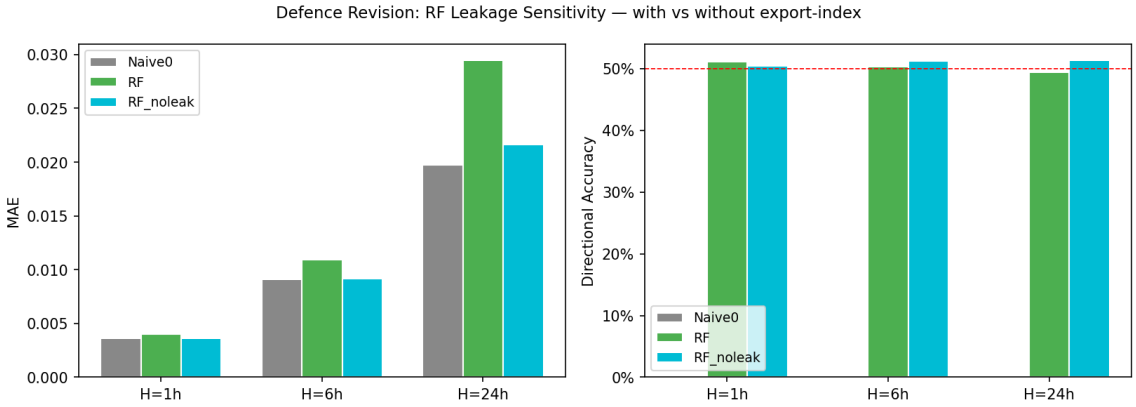

Figure 13. RF leakage sensitivity: mean absolute error and directional accuracy with and without the export-index proxy across forecast horizons.

Another issue that must be stated explicitly is the distinction between statistical forecasting accuracy, directional usefulness, and economic profitability. This dissertation evaluates forecast errors and directional metrics; it does not implement or test a full trading strategy. No transaction-cost model, slippage model, position-sizing rule, execution-latency control, or portfolio-construction layer is included in the reported experiments. Consequently, even if a model were to achieve a small statistical improvement, that would not by itself establish net trading value after realistic costs and frictions.

This distinction is important because statistical forecasting accuracy and economic usefulness are not equivalent. Ji et al. (2019) show that, even when deep learning models appear useful in prediction experiments, the translation into trading performance is model- and task-dependent, and they caution against direct practical deployment. Conversely, Nagula and Alexakis (2022) report a profitable hybrid Bitcoin futures strategy with stronger risk-adjusted performance than benchmark strategies, but their study includes a trading-

rule and futures-strategy layer that is outside the scope of this dissertation. Therefore, the present results should be interpreted as evidence about predictive accuracy and directional signal under strict validation, not as evidence for or against deployable trading profitability.

In highly liquid but noisy markets such as Bitcoin, small forecasting gains can be overwhelmed by costs, adverse selection, and instability in live deployment. The dissertation should therefore not be interpreted as evidence of practical tradability. Its contribution is methodological and empirical: it clarifies how difficult it is to obtain robust short-horizon forecasting gains once the evaluation protocol is made conservative.

A further contextual limitation concerns the market regime of the evaluation window. The January 2024–January 2025 period coincided with a predominantly bullish phase, shaped by the approval of spot Bitcoin ETFs in January 2024 and the April 2024 halving event — a structural shift documented by Bukaita and Li (2026), who report regime-dependent spillover dynamics between Bitcoin, Ethereum, gold, and U.S. equities surrounding the 2024 ETF institutionalisation regime. The Naive0 benchmark, which predicts zero return, is particularly well-suited to trending markets where extreme directional moves are infrequent relative to mean-reverting periods: it systematically avoids the directional errors that non-naive models can accumulate. In a bear regime or a high-volatility crash episode, Naive0 would tend to accumulate larger errors on days of sharp drawdown, potentially reducing its relative advantage over models capable of capturing downward momentum. The extent to which the reported rankings are regime-conditional rather than universally generalisable therefore remains an open question, and replication across additional regimes is identified as a priority in Table 9.

Taken together, these limitations reinforce rather than undermine the chapter's central argument. The finding that sophisticated models did not consistently outperform naive benchmarks under strict validation is methodologically credible precisely because it was obtained under a conservative protocol with explicit scope discipline — and it is directly relevant to the dissertation's broader claim about the difficulty of extracting stable predictive structure from a volatile, non-stationary Bitcoin market.

The methodological status of the archived LSTM track is discussed separately in Section 4.3 and should not be conflated with the core like-for-like benchmark.

5 Conclusions

This dissertation asked whether machine-learning models can forecast Bitcoin returns more effectively than strong naive benchmarks when the empirical design is reproducible, leakage-aware, and evaluated across multiple horizons. The answer is clear, but it must be stated with care. Across the 1-hour, 6-hour, and 24-hour horizons, and across seven comparators (Naive0, NaiveLast, OLS, Random Forest, ARIMA(1,1,1), SARIMA(1,1,1)(1,0,1,24), and an auxiliary LSTM), Naive0 produced the lowest mean MAE at every horizon. OLS was the strongest non-naive specification but remained 1.4%, 4.6% and 6.0% above Naive0 in average point-forecast error at H = 1, 6 and 24 hours, and was not significantly different from Naive0 by Diebold–Mariano testing ($p = 0.447$, 0.270 and 0.138). Directional performance was also weak: the highest observed directional accuracy was 52.51% (OLS at H = 6 hours), which remains only marginally above chance. The central conclusion is therefore that, within the present feature set, model configurations, validation design, and market regime, stable predictive gains over naive hourly Bitcoin benchmarks were not demonstrated. The Harvey, Leybourne, and Newbold (1997) correction of the multi-step Diebold–Mariano statistics and an OLS leakage-sensitivity rerun both left the central benchmark conclusion intact. The conclusion is explicitly conditional on the January 2024–January 2025 out-of-sample window, which was predominantly bullish (spot-ETF approval and the April 2024 halving): Naive0 is structurally favoured in trending markets without prolonged crashes, and the relative advantage observed here may therefore be partly a property of regime rather than a universal statement about hourly Bitcoin predictability (see Section 4.7).

These findings define the contribution of the dissertation more precisely. First, the study provides a reproducible forecasting framework that treats leakage prevention as a primary methodological constraint through causal feature alignment, training-only preprocessing, purge gaps, and rolling-origin walk-forward evaluation. Second, it documents a unified hourly master dataset and a controlled empirical comparison of seven model families (two naive baselines, one linear, one non-linear ensemble, two univariate statistical baselines, and an auxiliary recurrent sequence model) across three forecast horizons, with split-level outputs preserved for all of them. Third, it contributes evidence-led clarification to the literature by showing that performance claims that appear promising under weaker validation protocols may not survive stricter out-of-sample testing, and that the near-identical performance of ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24) under the present protocol is itself a diagnostic signal of specification artefact rather than evidence on daily seasonality in hourly Bitcoin returns. The contribution of the dissertation therefore lies less in identifying a universally superior model than in establishing a defensible leakage-aware benchmark for future Bitcoin forecasting research.

The results also carry substantive implications for the interpretation of prior work. They support the view that short-horizon Bitcoin return forecasting is constrained by low signal-to-noise ratios, volatility clustering, structural breaks, and regime dependence. They further suggest that mixed-frequency exogenous variables may offer limited intraday value when only daily or irregular observations are available and must be aligned conservatively to preserve causality.

The present findings align with the cautious strand of the literature and diverge from the more optimistic one, as discussed in detail in Section 4.6. They converge with Jaquart et al. (2021), Yae and Tian (2022), and Cakici et al. (2024), who find that machine-learning gains over strong benchmarks are limited once evaluation is properly controlled; and with Bouri et al. (2022) and Cortese et al. (2023), whose characterisation of state-dependent cryptocurrency return dynamics is consistent with the regime conditionality of the January 2024–January 2025 evaluation window. The results are less optimistic than those reported by McNally et al. (2018), Mudassir et al. (2020), Chen (2023) and Kashyap et al. (2024), and than the master's-level studies of Mendes (2019), Meijer (2020), Xu (2020) and Daş (2022); the divergence is attributable to the differences in target, frequency, and validation design documented in Section 4.6. The main empirical lesson is that any claimed improvement over naive benchmarks must be evaluated under temporally disciplined testing before it can be considered methodologically credible.

Several limitations should be recognised explicitly. The executed experiments relied predominantly on market and technical variables because exogenous data coverage was incomplete in the saved archive; sentiment coverage was partial and hash-rate variables were not available consistently in the final modelling dataset. Model tuning was intentionally limited to preserve comparability and reproducibility, which may have disadvantaged more flexible architectures such as Random Forest and LSTM. In addition, the dissertation evaluated forecast accuracy and directional performance rather than a full economic trading framework with transaction costs, execution constraints, and portfolio construction rules. Accordingly, the study should be read as an assessment of predictive accuracy under strict out-of-sample validation, not as a test of live trading profitability. For these reasons, the study should be interpreted as a rigorous benchmarking exercise rather than as the final word on cryptocurrency predictability.

The principal directions for future research are summarised in Table 9.

Table 9. Priority directions for future research arising from the defended benchmark study.

Future direction	Motivation from the present study	Methodological requirement
Regime-aware and hybrid models	The weak and unstable benchmark-relative results suggest that predictive structure may be conditional on market state rather than stable across the full sample.	Evaluate regime-sensitive specifications, such as HMM-based or hybrid models, under the same purged walk-forward protocol and compare them directly against strong naive benchmarks.
Higher-frequency exogenous data	Daily lagged sentiment and absent usable hash-rate coverage limited the realised multimodal contribution, especially at the 1-hour horizon.	Acquire intraday sentiment, news, order-flow, or on-chain proxies and align them causally at sub-daily frequency before forecasting.
Split-consistent explainability	The preserved archive does not yet include persistent explainability outputs that would clarify whether predictor relevance is stable across horizons and regimes.	Save split-level feature-importance and SHAP-type artefacts alongside forecast metrics in future replications.
Broader model comparison under equal conditions	The current archive covers two naive baselines, OLS, Random Forest, ARIMA(1,1,1), SARIMA(1,1,1)(1,0,1,24) and one un-tuned LSTM. Several model classes — boosting, GRU, Transformer-based architectures — remain outside this defended comparison.	Test boosting, GRU, and Transformer-based models under identical preprocessing, split construction, and reporting rules before drawing comparative conclusions.
Economic evaluation after statistical validation	Predictive accuracy and trading profitability are not equivalent.	Add transaction costs, slippage, turnover, execution rules, and a simple portfolio layer only after a model first demonstrates stable out-of-sample gains over strong naive benchmarks.
Extend residual-leakage sensitivity checks to remaining archived comparators (ARIMA, SARIMA, LSTM)	Post-hoc sensitivity reruns for OLS (Section 4.7) and Random Forest (Table 8) confirm that removing the export-index proxy does not reverse the central benchmark conclusion for these two comparators; equivalent evidence for	Re-execute the archived benchmark for the remaining comparable models — ARIMA, SARIMA and LSTM, with the export-index column removed and split-local screening enforced, then compare the resulting MAE,

	ARIMA, SARIMA, and LSTM remains outstanding.	out-of-sample R^2, and HLN-corrected Diebold–Mariano p-values against the defended baseline.

These directions extend the dissertation without weakening its central methodological principle: any increase in model complexity should remain subordinate to leakage-safe design, benchmark-relative evaluation, and cautious interpretation of out-of-sample evidence.

Under realistic, leakage-aware conditions, hourly Bitcoin returns remain difficult to forecast in a stable and defensible way: across seven comparators and 53 walk-forward splits at each of three horizons, no non-naive model achieved a statistically significant improvement over the zero-return benchmark. This dissertation's main contribution is to document that difficulty transparently and reproducibly, in a form that future work can build on.

6 References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., ... Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (pp. 265–283). USENIX Association. https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

Bailey, D. H., & López de Prado, M. (2014). The deflated Sharpe ratio: Correcting for selection bias, backtest overfitting, and non-normality. Journal of Portfolio Management, 40(5), 94–107. https://doi.org/10.3905/jpm.2014.40.5.094

Bouri, E., Christou, C., & Gupta, R. (2022). Forecasting returns of major cryptocurrencies: Evidence from regime-switching factor models. Finance Research Letters, 49, 103193. https://doi.org/10.1016/j.frl.2022.103193

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: Forecasting and control (5th ed.). John Wiley & Sons.

Bukaita, W., & Li, X. (2026). Structural spillovers among Bitcoin, Ethereum, gold, and U.S. equities: Evidence from the 2024 spot ETF institutionalization regime. Economies, 14(4), 143. https://doi.org/10.3390/economies14040143

Cakici, N., Shahzad, S. J. H., Będowska-Sójka, B., & Zaremba, A. (2024). Machine learning and the cross-section of cryptocurrency returns. International Review of Financial Analysis, 94, 103244. https://doi.org/10.1016/j.irfa.2024.103244

Chen, J. (2023). Analysis of Bitcoin price prediction using machine learning. Journal of Risk and Financial Management, 16(1), 51. https://doi.org/10.3390/jrfm16010051

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. Journal of Computational and Applied Mathematics, 365, Article 112395. https://doi.org/10.1016/j.cam.2019.112395

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1412.3555

Cortese, F. P., Kolm, P. N., & Lindström, E. (2023). What drives cryptocurrency returns? A sparse statistical jump model approach. Digital Finance. https://doi.org/10.1007/s42521-023-00085-x

Da Silva, R. G., Dal Molin Ribeiro, M. H., Fraccanabbia, N., Mariani, V. C., & Dos Santos Coelho, L. (2020). Multi-step ahead Bitcoin price forecasting based on variational mode decomposition and ensemble learning methods. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1–8). IEEE. https://doi.org/10.1109/IJCNN48605.2020.9207152

Daş, İ. (2022). Bitcoin price forecasting under the influences of network metrics and other financial assets [Master's thesis, Middle East Technical University]. METU Open. https://open.metu.edu.tr/handle/11511/101299

Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. Journal of Business & Economic Statistics, 13(3), 253–263. https://doi.org/10.1080/07350015.1995.10524599

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. The Journal of Finance, 25(2), 383–417. https://doi.org/10.1111/j.1540-6261.1970.tb00518.x

Farooq, A., Irfan Uddin, M., Adnan, M., Alarood, A. A., Alsolami, E., & Habibullah, S. (2024). Interpretable multi-horizon time series forecasting of cryptocurrencies by leveraging temporal fusion transformer. Heliyon, 10(22), e40142. https://doi.org/10.1016/j.heliyon.2024.e40142

Goodell, J. W., Ben Jabeur, S., Saâdaoui, F., & Nasir, M. A. (2023). Explainable artificial intelligence modeling to forecast bitcoin prices. International Review of Financial Analysis, 88, 102702. https://doi.org/10.1016/j.irfa.2023.102702

Gort, B. J. D., Liu, X.-Y., Sun, X., Gao, J., Chen, S., & Wang, C. D. (2023). Deep reinforcement learning for cryptocurrency trading: Practical approach to address backtest overfitting (arXiv:2209.05559) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2209.05559

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., & Oliphant, T. E. (2020).

Array programming with NumPy. Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Harvey, D. I., Leybourne, S. J., & Newbold, P. (1997). Testing the equality of prediction mean squared errors. International Journal of Forecasting, 13(2), 281–291. https://doi.org/10.1016/S0169-2070(96)00719-4

Hashish, I. A., Forni, F., Andreotti, G., Facchinetti, T., & Darjani, S. (2019). A hybrid model for Bitcoin price prediction using hidden Markov models and optimized LSTM networks. In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 721–728). IEEE. https://doi.org/10.1109/ETFA.2019.8869094

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. The Journal of Finance and Data Science, 7, 45–66. https://doi.org/10.1016/j.jfds.2021.03.001

Ji, S., Kim, J., & Im, H. (2019). A comparative study of Bitcoin price prediction using deep learning. Mathematics, 7(10), 898. https://doi.org/10.3390/math7100898

Kashyap, S., Ramisetty, S., & Narayanaswamy, R. (2024). A novel method of Bitcoin price prediction. In 2024 International Conference on Computing and Data Science (ICCDS) (pp. 1–4). IEEE. https://doi.org/10.1109/ICCDS60734.2024.10560443

Kehinde, T. O., Adedokun, O. J., Joseph, A., Kabirat, K. M., Akano, H. A., & Olanrewaju, O. A. (2025). Helformer: An attention-based deep learning model for cryptocurrency price forecasting. Journal of Big Data, 12(1), 81. https://doi.org/10.1186/s40537-025-01135-4

Khedr, A. M., Arif, I., Pravija Raj, P. V., El-Bannany, M., Alhashmi, S. M., & Sreedharan, M. (2021). Cryptocurrency price prediction using traditional statistical and machine-learning

techniques: A survey. Intelligent Systems in Accounting, Finance and Management, 28(1), 3–34. https://doi.org/10.1002/isaf.1488

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.1412.6980

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. Expert Systems with Applications, 197, 116659. https://doi.org/10.1016/j.eswa.2022.116659

Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

López de Prado, M. (2018). Advances in financial machine learning. John Wiley & Sons.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. International Journal of Forecasting, 36(1), 54–74. https://doi.org/10.1016/j.ijforecast.2019.04.014

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), Proceedings of the 9th Python in Science Conference (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of Bitcoin using machine learning. In 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP) (pp. 339–343). IEEE. https://doi.org/10.1109/PDP2018.2018.00060

Meijer, D. (2020). Predicting cryptocurrency price trends with long short-term memory (LSTM) [Master's thesis, Tilburg University]. https://arno.uvt.nl/show.cgi?fid=156536

Mendes, J. F. B. (2019). Forecasting Bitcoin prices: ARIMA vs LSTM [Master's thesis, ISCTE – Instituto Universitário de Lisboa]. https://repositorio.iscte-iul.pt/bitstream/10071/19724/1/Master_Joao_Batista_Mendes.pdf

Mittal, M., & Geetha, G. (2022). Predicting Bitcoin price using machine learning. In 2022 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1–7). IEEE. https://doi.org/10.1109/ICCCI54379.2022.9740772

Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: A machine learning approach. Neural Computing and Applications, 37(28), 22979–22993. https://doi.org/10.1007/s00521-020-05129-6

Munim, Z. H., Shakil, M. H., & Alon, I. (2019). Next-day Bitcoin price forecast. Journal of Risk and Financial Management, 12(2), 103. https://doi.org/10.3390/jrfm12020103

Nagula, P. K., & Alexakis, C. (2022). A new hybrid machine learning model for predicting the bitcoin (BTC-USD) price. Journal of Behavioral and Experimental Finance, 36, 100741. https://doi.org/10.1016/j.jbef.2022.100741

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Bitcoin.org. https://bitcoin.org/bitcoin.pdf

Pečiulis, T., Ahmad, N., Menegaki, A. N., & Bibi, A. (2024). Forecasting of cryptocurrencies: Mapping trends, influential sources, and research themes. Journal of Forecasting, 43(6), 1880–1901. https://doi.org/10.1002/for.3114

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Phaladisailoed, T., & Numnonda, T. (2018). Machine learning models comparison for Bitcoin price prediction. In 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE) (pp. 506–511). IEEE. https://doi.org/10.1109/ICITEED.2018.8534911

Raghava-Raju, A. (2018). A machine learning approach to forecast Bitcoin prices. International Journal of Computer Applications, 182(24), 39–46. https://doi.org/10.5120/ijca2018918041

Rathan, K., Sai, S. V., & Manikanta, T. S. (2019). Cryptocurrency price prediction using decision tree and regression techniques. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 190–194). IEEE. https://doi.org/10.1109/ICOEI.2019.8862585

Saad, M., & Mohaisen, A. (2018). Towards characterizing blockchain-based cryptocurrencies for highly accurate predictions. In IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 704–709). IEEE. https://doi.org/10.1109/INFCOMW.2018.8406859

Sedighi, M., Jahangirnia, H., Gharakhani, M., & Farahani Fard, S. (2019). A novel hybrid model for stock price forecasting based on metaheuristics and support vector machine. Data, 4(2), Article 75. https://doi.org/10.3390/data4020075

Serafini, G., Yi, P., Zhang, Q., Brambilla, M., Wang, J., Hu, Y., & Li, B. (2020). Sentiment-driven price prediction of Bitcoin based on statistical and deep learning approaches. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1–8). IEEE. https://doi.org/10.1109/IJCNN48605.2020.9206704

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, Article 106181. https://doi.org/10.1016/j.asoc.2020.106181

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... Vallat, R. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wu, C.-H., Lu, C.-C., Ma, Y.-F., & Lu, R.-S. (2018). A new forecasting framework for Bitcoin price with LSTM. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 168–175). IEEE. https://doi.org/10.1109/ICDMW.2018.00032

Xu, Y. (2020). Bitcoin price forecast using LSTM and GRU recurrent neural networks [Master's thesis, University of California]. https://escholarship.org/uc/item/70d9n5sd

Yae, J., & Tian, G. Z. (2022). Out-of-sample forecasting of cryptocurrency returns: A comprehensive comparison of predictors and algorithms. Physica A: Statistical Mechanics and its Applications, 598, 127379. https://doi.org/10.1016/j.physa.2022.127379

7 Appendices

Appendix A – Methodology Diagram

This appendix includes the methodological architecture diagram used to support the design and implementation of the leakage-aware Bitcoin return-forecasting framework defended in this dissertation.

The diagram provides a high-level overview of the end-to-end workflow, covering data ingestion, preprocessing, feature engineering, model training, validation, and evaluation stages. It visually summarises the interaction between statistical models, machine learning algorithms, and deep learning architectures, as well as the walk-forward validation strategy adopted to prevent data leakage.

Appendix B – Digital Artefact Delivery Note

This appendix clarifies the delivery status of the supplementary artefacts referenced in the dissertation. The primary research-compendium repository is hosted at https://zenodo.org/records/20283480 and the version-of-record snapshot used to support the defended results is archived with a persistent identifier at https://doi.org/10.5281/zenodo.20283480. The repository organises the working notebook set under the `jupyter_notebooks` folder, while mirrored copies may also appear at the repository root in the archived workspace. The raw Bitcoin OHLCV CSV file produced by the Binance/CCXT-based scraper is included in the repository to enable end-to-end re-execution without re-querying the upstream exchange. The methodological diagram is stored as `Methodology_diagram.drawio`, and generated supporting documentation is stored under `output/doc/`. These materials are provided to support reproducibility and traceability and are not reproduced in full within this PDF.

Appendix C – Jupyter Notebooks (Reproducibility and Experimental Artefacts)

This appendix contains the Jupyter notebooks developed to support reproducibility and traceability of the experimental pipeline used in this research. The notebooks document all stages of data processing, exploratory analysis, model training, evaluation, and result interpretation.

The notebooks are organised within the folder jupyter_notebooks and include the following files:

C.1 ETL and Data Preparation

- v1_etl_process.ipynb

Implements the Extract–Transform–Load (ETL) pipeline, including data ingestion from multiple sources, temporal alignment, feature construction, target engineering, and integrity checks.

C.2 Data Integrity Checks and Exploratory Data Analysis

- v1_integrity_EDA.ipynb

Performs dataset validation, missing-value analysis, exploratory data analysis (EDA), correlation inspection, and sanity checks to verify data consistency prior to modelling.

C.3 Model Training and Backtesting

- v1_models_test.ipynb

Contains the implementation of baseline models, machine learning algorithms, and deep learning architectures, along with the walk-forward backtesting framework and evaluation metrics.

C.4 Results Analysis and Reporting

- v1_result_analisys.ipynb [sic-Original filename retained in the repository]

Aggregates experimental results, generates comparative performance tables, and supports the analysis discussed in the Results and Discussion chapter.

Included folder:

- jupyter_notebooks/

Appendix D – Consolidated Hyperparameter Table

This appendix consolidates, in a single auditable view, every hyperparameter held constant across the leakage-aware walk-forward experiments defended in this dissertation. The hyperparameter settings cover every non-naive model entered into the defended like-for-like comparison; the ARIMA and SARIMA rows correspond to the configurations whose split-level results are reported in Tables 3–5 under the level-to-return reconversion caveat documented in Sections 3.4 and 4.2. The values were defined before out-of-sample evaluation and held constant across all 53 valid test splits (52 for LSTM) and all three forecasting horizons (H = 1, 6, 24). Centralising these settings here complements the textual

descriptions in Section 3.4 and supports independent replication from the published code in Appendix C.

Table 10. Consolidated hyperparameter configurations

Model	Parameter	Value	Note
Random Forest	n_estimators	400	number of trees
	min_samples_leaf	2	minimum leaf size
	random_state	42	reproducibility seed
	n_jobs	-1	all available cores
LSTM (PyTorch)	hidden_dim	50	single recurrent layer
	num_layers	1	stacked depth
	lookback	48 h	input sequence window
	batch_size	64	mini-batch size
	epochs	20	maximum training epochs
	learning_rate	1×10^{-3}	Adam optimiser
	min_train_seqs	200	skip-split guard
OLS	feature scaling	StandardScaler	fit on train only
	regularisation	none	plain OLS, no penalty
ARIMA	p, d, q	1, 1, 1	AR/I/MA orders
SARIMA	(p,d,q)	(1,1,1)	non-seasonal part
	(P,D,Q,s)	(1,0,1,24)	seasonal period

Note. All values are fixed prior to out-of-sample evaluation. Source: v1_models_test.ipynb (model-definition cells). The notebook contains both a TensorFlow/Keras and a PyTorch implementation of the LSTM; the PyTorch version listed above is the configuration that produced the archived results, while the Keras variant remained inactive in the defended run.

Appendix E – Feature Catalogue

This appendix lists all columns present in the master hourly dataset assembled by v1_etl_process.ipynb, indicating their domain, construction rule and operational role in the leakage-aware benchmark defended in this dissertation. The catalogue distinguishes raw

exchange and identifier fields excluded from prediction, candidate technical features computed from the OHLCV series, exogenous sentiment and on-chain fields whose hourly coverage was partial or absent, and targets and engineered return columns. Whether a candidate feature reaches the predictor matrix at training time is decided per walk-forward split by the screen described in Section 3.3 (training-sample coverage ≥ 70%, non-constant in the training block); columns marked Excluded a priori are never offered to the screen.

Table 11. Raw exchange and identifier columns (excluded from prediction).

Column	Type	Role
open_time	timestamp	Hourly UTC index (key)
close_time	timestamp	Bar close timestamp (raw, redundant with index)
Unnamed: 0	integer	Export index — monotonic row counter; flagged in Section 3.5
symbol	string	Trading pair label (BTC/USDT) — constant
ignore	raw	Binance reserved field — unused
open	price	OHLCV — bar open (USDT)
high	price	OHLCV — bar high (USDT)
low	price	OHLCV — bar low (USDT)
close	price	OHLCV — bar close (USDT); basis for log_close and targets
volume	float	Base-asset volume
quote_vol	float	Quote-asset volume
trades	integer	Number of trades in bar
taker_buy_base	float	Taker buy volume (base)
taker_buy_quote	float	Taker buy volume (quote)

Note. open, high, low, close, volume, quote_vol, trades, taker_buy_base and taker_buy_quote are raw OHLCV fields. They are not used as direct predictors in the defended pipeline because the technical-indicator block and the engineered return columns are computed from them; offering both the raw and the engineered forms would inflate the candidate set with collinear inputs. ARIMA and SARIMA use only the univariate log_close series.

Table 12. Candidate technical features computed from OHLCV (offered to the split-local screen).

Domain	Feature(s)	Indicator / construction
Volume	volume_adi	Accumulation/Distribution Index (Chaikin)
Volume	volume_obv	On-Balance Volume
Volume	volume_cmf	Chaikin Money Flow
Volume	volume_fi	Force Index
Volume	volume_em, volume_sma_em	Ease of Movement and its SMA
Volume	volume_vpt	Volume-Price Trend
Volume	volume_vwap	Volume-Weighted Average Price
Volume	volume_mfi	Money Flow Index
Volume	volume_nvi	Negative Volume Index
Volatility	volatility_bbm, _bbh, _bbl, _bbw, _bbp, _bbhi, _bbli	Bollinger Bands (mean, high, low, width, %B, hi/lo indicators)
Volatility	volatility_kcc, _kch, _kcl, _kcw, _kcp, _kchi, _kcli	Keltner Channel family
Volatility	volatility_dcl, _dch, _dcm, _dcw, _dcp	Donchian Channel family
Volatility	volatility_atr	Average True Range
Volatility	volatility_ui	Ulcer Index

Domain	Feature(s)	Indicator / construction
Trend	trend_macd, trend_macd_signal, trend_macd_diff	MACD family
Trend	trend_sma_fast, trend_sma_slow, trend_ema_fast, trend_ema_slow	Fast/slow simple and exponential moving averages
Trend	trend_vortex_ind_pos, _neg, _diff	Vortex Indicator
Trend	trend_trix	Triple Exponential Average (TRIX)
Trend	trend_mass_index	Mass Index
Trend	trend_dpo	Detrended Price Oscillator
Trend	trend_kst, trend_kst_sig, trend_kst_diff	Know Sure Thing
Trend	trend_ichimoku_conv, _base, _a, _b, trend_visual_ichimoku_a, _b	Ichimoku cloud components
Trend	trend_stc	Schaff Trend Cycle
Trend	trend_adx, trend_adx_pos, trend_adx_neg	Average Directional Index
Trend	trend_cci	Commodity Channel Index
Trend	trend_aroon_up, _down, _ind	Aroon indicator
Trend	trend_psar_up, _down, _up_indicator, _down_indicator	Parabolic SAR
Momentum	momentum_rsi	Relative Strength Index
Momentum	momentum_stoch_rsi, _stoch_rsi_k, _stoch_rsi_d	Stochastic RSI family
Momentum	momentum_tsi	True Strength Index
Momentum	momentum_uo	Ultimate Oscillator
Momentum	momentum_stoch, momentum_stoch_signal	Stochastic Oscillator
Momentum	momentum_wr	Williams %R
Momentum	momentum_ao	Awesome Oscillator
Momentum	momentum_roc	Rate of Change
Momentum	momentum_ppo, _ppo_signal, _ppo_hist	Percentage Price Oscillator
Momentum	momentum_kama	Kaufman Adaptive Moving Average
Momentum	momentum_pvo, _pvo_signal, _pvo_hist	Percentage Volume Oscillator
Candlestick	morningstar, hammer, piercing, 3soldiers, engulfing	Binary candlestick pattern flags
Returns/other	others_dr, others_dlr, others_cr	Daily return, daily log-return, cumulative return
Custom MA	sma200, sma50, ema200, ema50	Long/medium SMAs and EMAs (manual)
Custom slope	slope, slope_obv, slope_rsi	Rolling slope of price / OBV / RSI

Note. All technical indicators are computed causally on the hourly OHLCV series using the ta-lib / TA Python library (per v1_etl_process.ipynb). The table groups the indicators by family and lists the canonical names used in the literature; once the family expansions (e.g. Bollinger Bands variants, Ichimoku components, MACD signal/diff) are unrolled into individual columns, the candidate predictor matrix contains 113 numeric columns. The global usability screen (Section 3.3) reduces this to 104 retained features; the split-local screen then leaves on average 103 features per split. OLS, Random Forest and the auxiliary LSTM consume the resulting per-split matrix; ARIMA and SARIMA do not (univariate baselines on log_close).

Table 13. Exogenous sentiment and on-chain fields (partial or absent hourly coverage).

Field	Domain	Native frequency	Operational status in defended benchmark
sent_Short Description	*Sentiment*	*Daily (text label)*	*Lagged 1 day, forward-filled to hourly grid; partial coverage on the final hourly grid (42.95% non-missing) — used by ML models when retained by the split-local screen.*
sent_Accurate Sentiments	*Sentiment*	*Daily (numeric)*	*Lagged 1 day, forward-filled to hourly grid; partial coverage as above; subject to per-split screen.*
sent_asof_date	*Sentiment*	*Daily*	*Provenance timestamp; not a predictor.*
hash-rate	*On-chain (network)*	*Daily*	*Forward-filled from a daily source that contained a single observation, so the resulting hourly column was empty across all 64,861 rows and excluded de facto by the coverage screens (Section 3.3); see also Section 3.7.*
hash_asof_date	*On-chain*	*Daily*	*Provenance timestamp; not a predictor.*

Note. Mixed-frequency alignment follows Section 3.2: daily exogenous series are aggregated to daily means, lagged by one calendar day, and forward-filled into the hourly Bitcoin index, so each hour in day D uses exogenous information no later than day D−1.

Table 14. Engineered return columns and supervised targets.

Column	Construction	Role
date	*UTC date derived from open_time*	*Auxiliary key for daily alignment*
log_close	*log(close)*	*Univariate series fitted by ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24); also used to derive ret_t*
ret_t	*log(close_t) − log(close_{t−1})*	*1-hour realised log-return (lagged feature)*
y_ret_t1	*log(close_{t+1}) − log(close_t)*	*Supervised regression target at H = 1 (forward log-return)*
y_dir_t1	*sign(y_ret_t1)*	*Directional label at H = 1 (diagnostic only)*
close_date	*Daily close (from hourly close)*	*Used to derive daily-frequency engineered columns*
ret_d	*log(close_date_d) − log(close_date_{d−1})*	*1-day realised log-return*
y_ret_d1	*log(close_date_{d+1}) − log(close_date_d)*	*Daily forward log-return (auxiliary)*
y_dir_d1	*sign(y_ret_d1)*	*Daily directional label (auxiliary)*

Note. The supervised regression target at horizon H ∈ {1, 6, 24} hours is the forward log-return y_t^(H) = log(close_{t+H}) − log(close_t), constructed at training time. The columns y_ret_t1 / y_dir_t1 in v1_etl_process.ipynb correspond to the H = 1 case; H = 6 and H = 24 targets are derived at split time. Direct price-level columns (open, high, low, close) and y-prefixed forward columns are removed from the candidate predictor set before screening so that no future information enters the input matrix.

Synthesis — what each model sees at prediction time. Naive0 ignores all features (predicts zero return). NaiveLast uses only ret_t (the most recent realised 1-hour return). ARIMA(1,1,1) and SARIMA(1,1,1)(1,0,1,24) are estimated on the univariate log_close series only, with no exogenous predictors. OLS and Random Forest receive the full per-split predictor matrix: the technical features in Table 12 plus the sentiment fields in Table 13 that pass the split-local screen (raw OHLCV and identifier columns in Table 11 are removed beforehand). The auxiliary LSTM consumes the same per-split predictor matrix as a fixed 48-hour lookback sequence (Appendix D). The hash-rate field is offered to the screen but is dropped by it at every split because it has no non-missing hourly values in the archived export (Section 3.7).