Benchmarking Cryptocurrency Forecasting Models in the Context of Data Properties and Market Factors

---

Author:
Stephan W.H. Akkerman
6397514

Supervisors:
Dr. Ioana R. Karnstedt-Hulpus
Prof. Cornelis W. Oosterlee
Dr. Abdulhakim A.A. Qahtan

A thesis submitted in fulfillment
of the requirements for the degree of
MSc. Artificial Intelligence
45 ECTS

Department of Information and Computing Sciences
Graduate School of Natural Sciences
Faculty of Science

Utrecht University
Utrecht, The Netherlands

October 2023

Author:
Stephan W.H. Akkerman

Supervisors:
Dr. Ioana R. Karnstedt-Hulpus
Prof. Cornelis W. Oosterlee

Institute:
Utrecht University, Utrecht, The Netherlands

CONTENTS

ABSTRACT

Cryptocurrency price prediction presents a significant challenge due to the inherent non-linearity of the market. In this thesis, we assess the performance of thirteen time series forecasting models in predicting the prices of twenty-one different cryptocurrencies across four specific time frames. Our analysis centers on how data characteristics and market conditions affect the precision of these models and explores the implications of both broadening the scope of training data and extending the forecast periods. Our findings indicate that TBATS, LightGBM, XGBoost, and ARIMA consistently deliver the most accurate results. We identify key factors influencing prediction accuracy, including market trends, heteroskedasticity, volatility, and market capitalization. Additionally, the choice of time frame markedly affects all models' predictive accuracy. Contrary to expectations, we observe that increasing the volume of training data does not necessarily enhance the performance of deep-learning and RNN-based models. Our thesis offers a comprehensive benchmark of forecasting models within the cryptocurrency context, underscoring the conditions crucial for improving prediction accuracy. We have made our code accessible on GitHub[1] to promote research transparency and facilitate further collaborative exploration in this field.

1 https://github.com/StephanAkkerman/crypto-forecasting-benchmark

1 INTRODUCTION

Cryptocurrencies, like Bitcoin (BTC) and Ethereum (ETH), have captured widespread attention in recent years, offering a new way to store and exchange value [1, 2]. As the cryptocurrency market has grown, so has the need for reliable methods to predict their prices. Getting these predictions right can help investors make smart decisions and develop effective trading strategies [3].

Predicting cryptocurrency prices is particularly complex due to their distinct and challenging characteristics. Not only do cryptocurrencies exhibit nonlinearity [4] and a lack of discernible seasonal patterns [5], but they also possess a high degree of noise [6] and demonstrate pronounced trends [7]. These properties render cryptocurrencies an intriguing subject for research. However, it's pivotal to acknowledge that these characteristics may vary significantly with different time frames, introducing complexities in analysis and prediction [8]. Furthermore, volatility, a measure of price variability, presents another layer of challenge. Cryptocurrencies, known for their high volatility, can experience vast price swings within short periods, complicating predictive modeling [9]. In addition to these, the market capitalization of cryptocurrencies influences study outcomes. Most research tends to concentrate on prominent cryptocurrencies, which are typically those with substantial trading volumes and market capitalizations. This narrow focus may skew data and findings, as higher market capitalizations generally correspond to reduced volatility, not just in cryptocurrencies [10] but also in traditional stocks [11], affecting the predictability and stability of the assets.

In this thesis, we examine the influence of data properties and market factors on cryptocurrency price predictions through diverse forecasting models. We analyze both state-of-the-art and traditional models, assessing their performance across various cryptocurrencies and connecting outcomes to specific data characteristics and market conditions. Moreover, we consider multiple time frames and the impact of data span to comprehensively understand their effects on the models' predictive accuracy.

1.1 TIME SERIES FORECASTING

Time series forecasting, the practice of predicting future data points by analyzing past values, plays a crucial role in various disciplines including economics and finance [12].

This technique takes into account values gathered at consistent intervals and seeks to forecast subsequent data points in the series.

Broadly, time series forecasting encompasses two primary approaches: univariate and multivariate methods [13]. The univariate approach, involving a single time series for predictions, is often appreciated for its simplicity and ease of implementation, albeit sometimes at the cost of neglecting potential complexities within the data. On the contrary, multivariate methods use multiple time series and, while potentially offering greater accuracy, can be more challenging both to implement and interpret due to their consideration of multiple variables. This paper specifically spotlights univariate methods, employing them to predict cryptocurrency asset prices.

Traditionally, the field has leaned on statistical models, such as the autoregressive integrated moving average (ARIMA) and exponential smoothing models, for time series forecasting. However, recent advancements in machine learning have unlocked new potentials, introducing models capable of deciphering complex patterns within data. This thesis navigates through the latest evolutions in time series forecasting models, offering insights into their applicability for predicting cryptocurrency asset prices, with a research lens focused on examining measured volatility across various time frames. Subsequent sections delve deeper, explaining the pivotal role of volatility in this context.

1.2 DATA PROPERTIES

Financial time series often exhibit specific data properties that, if not properly understood and accounted for, can impact the accuracy of forecasting models. This thesis examines key properties, including stationarity, autocorrelation, trends, seasonality, heteroskedasticity, and stochasticity, due to their significant influence on model predictions [14].

Recognizing these properties in our data allows us to apply appropriate transformations, potentially enhancing forecasting accuracy. These transformations, detailed in later sections, are crucial because they help us mitigate any distortions these properties might introduce into the forecasts. Additionally, understanding the presence of these properties, even after data transformation, provides valuable insights that can explain certain model outcomes.

In Section § 3.2, we delve deeper into the nature of each property, their importance, and the methods used for their investigation. Subsequently, Section § 5.2 explores the consequences of these data properties on forecasting performance, leveraging the insights gained from our analysis.

1.3 VOLATILITY

Volatility, within the realm of finance, relates to the measurement of price fluctuations of a security over a designated period [15, 16]. A concept that dynamically shifts in accordance

with prevailing market conditions, volatility is considered important for comprehending both the risks entwined with investments and the prospective returns thereof. High volatility often signals an elevated risk level, attributed to the quick and unpredictable shifts in prices, whereas low volatility typically conveys stability and diminished risk [17]. Nonetheless, exceptions are plausible, such as during a bear market's volatility [18], or instances where high volatility aligns with low risk amidst robust economic fundamentals.

The consequential impact of volatility extends to the predictive accuracy of time series forecasting models, given that variations in volatility can introduce fluctuations in data patterns, thereby influencing the precision of projections [19]. Models deployed for forecasting purposes hinge on historical data to formulate predictions regarding forthcoming trends. Encountering high volatility within historical data often complicates the model's capability to discern pertinent patterns and data relationships, potentially diminishing predictive accuracy [20]. Conversely, low volatility within historical data might enhance the model's capacity to identify significant patterns and relationships, potentially fostering more accurate predictions.

One widely adopted method for calculating a financial instrument's volatility involves determining the standard deviation of the instrument's returns over a specified duration. Typically, returns—calculated as the percentage alteration in the instrument's price across the duration—are daily. The computational representation is as follows:

$$\sigma_T = \sigma\sqrt{T} \tag{1.1}$$

σ_T= Volatility over a time horizon,

σ= Standard deviation of returns,

$\sqrt{T}$= Number of periods within a time horizon.

This thesis seeks to investigate the influence of volatility on the predictive accuracy of forecasting models, with a particular emphasis on cryptocurrencies. Specifically, it aims to discern whether periods of heightened volatility correlate with an augmented difficulty in accurately predicting cryptocurrency price movements.

1.4 MARKET CAPITALIZATION

We take a detailed look at the role of market capitalization and its impact on forecasting models within the financial realm, particularly focusing on its relationship with volatility. Market capitalization, in simple terms, refers to the total value of a financial instrument, like stocks or cryptocurrencies, which is calculated by multiplying its price by the total number of units available [21]. Financial instruments commonly undergo classification into distinct groups—large-cap, mid-cap, and small-cap—each determined by their respective market capitalization values, adhering to established boundaries. In alignment with prior

research, we adhere to previously defined thresholds: entities valued at ten billion are classified as large-cap; those between two and ten billion are deemed mid-cap; and those valued at under two billion are categorized as small-cap [22].

Generally, instruments with a larger market capitalization exhibit more stable and predictable price movements due to their widespread investor base and higher liquidity, and therefore, are often seen as less volatile [10]. This predictability is theorized to potentially enhance the accuracy of time series forecasting models when they are applied to larger market capitalization instruments. However, the exact relationship between volatility and market capitalization is not deeply rooted in academic research. A study of the Nairobian stock exchange revealed only a weak connection between these two factors [23], and very little research directly explores this relationship in the field of cryptocurrency. This study, therefore, ventures into this under-explored domain, probing the interactions between market capitalization and volatility specifically within cryptocurrency markets.

Despite any conclusions regarding the strength of the relationship between market capitalization and volatility, exploring the performance of time series forecasting models across cryptocurrencies of various market capitalizations remains critical. Researching smaller market cap instruments, in particular, should not be set aside for several reasons.

Firstly, instruments with smaller market capitalization might present greater potential for growth compared to their larger counterparts, as they have additional scope to increase in value [24, 25]. If forecasting models are determined to be accurate in predicting the performance of small capitalization instruments, investors could leverage this knowledge to pinpoint and take advantage of potential growth opportunities.

Secondly, small capitalization instruments could be more volatile and less predictable than large capitalization instruments, due to their restricted liquidity and smaller investor bases [26]. Consequently, forecasting models using data from small capitalization instruments may not predict their future values as accurately. Gaining insight into the predictability of forecasting models on small capitalization instruments could assist investors in better navigating the risks and potential returns of these instruments.

Thirdly, examining the predictability of forecasting models on small capitalization instruments might further illuminate the relationship between market capitalization and the accuracy of forecasting models, thereby enriching future research and enhancing the efficacy of forecasting models across various financial instruments.

Conclusively, this thesis aspires to thoroughly examine the cryptocurrency market, deploying analyses across a varied assortment of cryptocurrencies with differing market capitalizations, with the purpose of gaining more comprehensive insights into market behaviors and predictability.

1.5 TIME FRAMES

The interval at which data points are gathered—known as the time frame—plays a pivotal role in shaping the effectiveness of time series forecasting models [27]. Different time intervals may reveal distinct patterns and attributes in the data, which, in turn, influence the accuracy of the resulting forecasts [28]. Furthermore, time frames carry notable weight in the computation of volatility, an essential factor in our investigation.

A body of research indicates that data collected at higher time frames, such as on a daily or weekly basis, tends to exhibit less noise and greater predictability compared to data from lower time frames, like hourly or minute data [8]. This distinction is attributed to a 'smoothing effect' inherent in larger time frames. Consequently, time series forecasting models, when trained on data from higher time frames, may offer more precise predictions regarding the future values of financial instruments. This reality has led many researchers to focus primarily on higher time frames, anticipating a superior performance from their forecasting models.

However, the significance of lower time frame data warrants attention, capturing as it does more detailed and short-term fluctuations, which might be omitted in data from higher time frames. Thus, models trained on such detailed data may offer more accurate predictions regarding short-term financial movements.

Even though higher time frames may initially appear to be preferable, the growing popularity of lower time frames in contemporary trading practices compels consideration. Notably, the average holding period of shares has dramatically decreased from 8 years in the late 1950s to a mere 5.5 months in 2020, according to Reuters [29]. This shift has been partly driven by the advent of high-frequency trading, which leverages advanced computer algorithms to execute numerous orders within milliseconds [30]. This trend underscores the necessity of factoring in lower time frames when employing time series forecasting models for financial predictions.

Notably, recent research suggests that Bitcoin markets exhibit efficient behavior at one-minute time frames, while the same study shows inefficiencies when observed at five-minute intervals [28]. These findings might also apply to other cryptocurrency markets or to markets exhibiting similar behaviors to that of Bitcoin.

An additional consideration is that lower time frames may demonstrate a heightened sensitivity to market conditions and external factors, such as news events or economic data releases [31]. A thorough understanding of the predictability of time series forecasting models, particularly when applied to lower time frames, can equip investors and traders with the insight to adeptly navigate and respond to such events.

1.6 RESEARCH QUESTIONS

This thesis seeks to explore the comparative performance of state-of-the-art machine learning models and traditional forecasting models in the domain of cryptocurrency price prediction. Furthermore, it delves into the investigation of how data characteristics and market conditions impact the efficiency of these forecasting models. Another layer of this research probes into the models' performance variability when training and testing data are altered. Guided by the following research questions (*RQ*s), this investigation unfolds:

*RQ*1: *How do the performances of state-of-the-art forecasting models compare to traditional models in forecasting cryptocurrency prices?*

*RQ*2: *What effects do data properties and market factors have on cryptocurrency prediction models?*

> *RQ*2*a*: How do data properties, like stationarity, autocorrelation, trend, seasonality, heteroskedasticity, and stochasticity impact the performance of cryptocurrency forecasting models?
>
> *RQ*2*b*: How do market factors such as volatility and market capitalization affect predictions?
>
> *RQ*2*c*: Does the time frame alter model effectiveness?

*RQ*3: *How sensitive are state-of-the-art forecasting models to the quantity of training data and to what degree is prediction performance sustained over time?*

To navigate through the responses to these research questions, we perform a comparative analysis, benchmarking the forecasting models against cryptocurrencies characterized by varying degrees of volatility, market capitalization, and time frames. Additionally, to address RQ3, the same benchmarking procedure is conducted, using distinct sets of training and testing data, thereby examining the models' adaptability and predictive stability over differing future periods.

1.7 THESIS OUTLINE

Our thesis is organized into six chapters to clearly address the research questions. Chapter 2 outlines previous studies, explains the background of forecasting cryptocurrency prices, and highlights the contributions of this thesis. In Chapter 3, we explain how we gather and analyze our data. Chapter 4 discusses the specific methods we use in our research, including how we assess different models. Our findings are examined in Chapter 5, where we end with clear answers to our research questions. Finally, Chapter 6 shares the conclusions we draw from this thesis.

2

RELATED WORK

In this chapter, an overview of diverse approaches and techniques used in the forecasting of financial time series data, specifically focusing on cryptocurrency price prediction, is presented. Additionally, insights derived from prior comparative evaluations are included. The aim of this section is not to delve into a detailed exposition of the workings of models but to discuss the models we use in this study and assess their performance as evaluated by preceding research.

We start with an exploration of the most fundamental forecasting model, as detailed in Section § 2.1. Subsequent to this, advanced techniques are discussed in Section § 2.2, wherein Recurrent Neural Networks, along with Long Short-Term Memory and Gated Recurrent Unit architectures, are addressed. In Section § 2.3 we delve into deep learning-based models, covering convolutional, residual, and attention-based architectures.

Section § 2.4 covers the ensemble-based models, which apply strategies such as bagging and boosting. We provide insights into the chosen hybrid model in Section § 2.5. Following this, attention shifts to time series decomposition-based models in Section § 2.6, exploring both trigonometric and Bayesian methodologies. In Section § 2.7 we discuss the critical role of data, emphasizing the importance of the data employed in the forecasting models. Concluding the chapter, Section § 2.8 presents the limitations of prior research and our three main contributions.

2.1 STATISTICAL ANALYSIS MODEL

This section introduces the Autoregressive Integrated Moving Average model, commonly referred to as ARIMA [32]. Originating as an extension of the ARMA model—a hybrid of the traditional autoregressive (AR) [33] and Moving Average (MA) [34] time series forecasting models—ARIMA incorporates an integration component. This addition allows it to adeptly model non-stationary time series by accounting for trends in the data [35]. Notwithstanding, ARIMA encounters limitations in handling non-linear dependencies within the data [36].

Despite being the most historically rooted model explored in this thesis, ARIMA sustains its contemporary relevance and usage, attributed to its notable flexibility [37]. Consequently, this model will serve as a benchmark in this thesis, providing a foundational

comparative metric against which newer and potentially enhanced forecasting models will be evaluated.

2.2 RECURRENT NEURAL NETWORK-BASED MODELS

Machine learning approaches, renowned for their capacity to manage non-linear dependencies, present a compelling application in fitting financial data, such as stocks [38] and cryptocurrency prices [39]. One prominent machine-learning model used for time series forecasting is the Recurrent Neural Network (RNN) [40, 41]. The performance of RNNs within the financial sector yields mixed outcomes. Research indicates that RNNs outperform Artificial Neural Networks (ANN) in crafting investment portfolios across various stock markets [42]. Conversely, RNNs do not demonstrate significant superiority over simpler, more easily fitted Markov models in predicting options volatility [43].

This inconsistent performance may stem from a notable challenge associated with RNNs: the vanishing gradients problem, as shown by Hochreiter (1997) [44]. The study explicates that suboptimal weight selection or initialization can result in significantly diminished gradients as they traverse through multiple layers, thereby hampering the network's learning capability. This issue potentially complicates the training of RNNs on extensive data sequences, as ineffective gradient propagation through the network may occur. Consequently, this renders a critical consideration in employing RNNs within financial forecasting contexts.

2.2.1 *Long-Short Term Memory*

The Long-Short Term Memory (LSTM) model emerged as a solution to the vanishing gradients problem, which was identified as a significant hurdle in effective neural network training [45]. This model demonstrates superior efficacy in handling sequential data, particularly by adeptly capturing longer-term dependencies within the data, in comparison to its predecessor, the traditional Recurrent Neural Network. Nonetheless, this capability does not come without computational expense: the LSTM uses a variety of gates to manage information flow, which inherently demands additional calculations. Moreover, due to an increased array of weights and biases, the LSTM necessitates heightened computational resources and time during the training phase of the network.

Existing research presents an intriguing narrative with respect to the comparative performance of the LSTM and the RNN, particularly in the domain of stock price forecasting [46]. A slightly superior performance by the RNN is observed in certain contexts, such as in the prediction of stock, Ethereum, and Bitcoin prices [47]. Importantly, the margin of outperformance of the RNN over the LSTM is not strikingly substantial. Consequently, in the spirit of thorough analysis, this thesis includes both models to compare their performances in the context of a larger dataset.

2.2.2 *Gated Recurrent Unit*

The Gated Recurrent Unit (GRU), introduced as a variant of the LSTM, strives for a balance of simplicity and computational efficiency without compromising the ability to manage long-term data dependencies within time series data [48]. Notably, the GRU refines the architectural complexity of the LSTM by curating a reduced parameter and computational footprint during network training, aiming to preserve the robustness in capturing long-term dependencies intrinsic to time series data.

Prior research presents mixed results regarding the efficacy of the GRU in financial forecasting contexts. For instance, some studies illuminate the GRU's superior performance over other RNN-based models in predicting stock prices [49]. In contrast, the very authors of the GRU point towards circumstances where it does not invariably outperform its RNN and LSTM counterparts, particularly citing instances where the LSTM exhibited superior performance with longer data sequences.

This difference in research outcomes extends into cryptocurrency forecasting as well. Some studies endorse the GRU for its commendable predictive accuracy concerning the prices of Bitcoin, Ethereum, and Litecoin [50, 51] while contrasting findings suggest the LSTM to be a more proficient model in forecasting Bitcoin prices specifically [52]. Considering these varied outcomes, we include the GRU in our comparison of time series forecasting models.

2.3 DEEP LEARNING-BASED MODELS

Deep learning-based models, emerging as a subtype of artificial neural networks like their RNN-based counterparts, are crafted to manage substantial data volumes and discern hierarchical data representations through the deployment of multiple artificial neuron layers. This exploration intends to scrutinize three distinct deep learning architectures for time series forecasting, namely convolutional, residual, attention-based, and RNN architectures.

2.3.1 *Convolutional Architecture*

Despite recurrent architectures frequently serving as the initial framework for sequence modeling tasks, a growing body of research indicates that convolutional architectures might potentially outperform them in sequence processing undertakings [53–56]. The Temporal Convolutional Network (TCN) employs a convolutional architecture and has demonstrated its capacity to outperform RNN-based models across diverse datasets [57]. Notably, the authors did not benchmark using a financial dataset.

Subsequent research affirms the TCN's superior performance relative to RNN-based models, even within the context of stock price datasets [58]. Furthermore, the TCN emerged as the best-performing model in a study that focused on predicting stock prices using ultra-high frequency stock change data [59]. Collectively, these studies underscore the TCN as a

promising convolutional architecture, illustrating its ability to surpass RNN-based models in forecasting stock prices. Additional research highlights the TCN's top-tier performance, even amongst RNN-based models, in predicting Ethereum prices [60]. Consequently, this model is incorporated into this thesis, where it will be subject to evaluation against a more expansive cryptocurrency dataset.

2.3.2 *Residual Architecture*

N-BEATS uses backward and forward residual links and is faster to train, interpretable, and applicable to a wide array of target domains [61]. The authors show its efficacy by highlighting a superior performance relative to the 2019 victor of the M4 competition, an event featuring a dataset comprising 100,000 time series [62]. This underscores N-BEATS as a promising model in the realm of time series forecasting.

Nevertheless, the deployment of N-BEATS in financial predictions is limited, evidenced by two studies that compare its performance with LSTM and ARIMA in the context of Bitcoin price prediction [63, 64]. The findings from this study divulged that N-BEATS surpassed LSTM and ARIMA across diverse time frames. However, this favorable outcome did not extend to stock market predictions, wherein N-BEATS was relegated to the position of the least proficient model when compared to LSTM, GRU, TCN, and Temporal Fusion Transformer (TFT) [65]. These mixed results lead us to include this forecasting model in this thesis.

2.3.3 *Attention-based Architecture*

The Temporal Fusion Transformer (TFT), which employs an attention-based architectural framework, demonstrates noteworthy capabilities in forecasting, as demonstrated by its empirical performance across diverse datasets [66]. Notably, the authors illustrate that the model outperforms ARIMA in various contexts, including a dataset about stock volatility.

Prior research further substantiates the capabilities of TFT by revealing its superior performance relative to the LSTM in predicting the trajectory of the S&P 500 and Google's stock price [67]. In contrast, another study shows that TFT performs worse compared to LSTM, GRU, TCN, and ARIMA in predicting cryptocurrency prices [68]. This underwhelming performance was corroborated by another paper, wherein TFT did not manage to outperform any model in the context of Ethereum price prediction [60]. We include TFT in our analysis because of its mixed results in the domain of financial forecasting.

2.4 ENSEMBLE-BASED MODELS

Ensemble models hold a pivotal role in machine learning, strategically combining predictions from various basic models to create a more accurate and trustworthy forecast. Simply put, ensemble models blend the strengths of several models to counterbalance their indi-

vidual shortcomings, providing a more resilient and reliable prediction. We cover two subcategories of ensemble models, bagging and boosting, each offering a distinctive strategy for merging predictive information. We now discuss these subcategories in more detail.

2.4.1 *Bagging*

In this section, we delve into a specific type of bagging model, the Random Forest [69]. Random Forest, a collection of numerous decision trees, serves as a model that merges the predictions of individual trees, aiming to reduce variability and enhance the overall prediction quality.

Examining its real-world application, it becomes clear that the performance of the Random Forest model can vary in different contexts. For example, in predicting the price changes of Ethereum, it did not surpass the predictive accuracy of ARIMA and the RNN [70]. In contrast, it did outperform the LSTM when predicting Bitcoin prices [71]. It is worth noting that the Random Forest model is commonly used with multiple features, as was the case in the studies mentioned. However, this thesis focuses solely on price as an input for the models, aiming to explore its performance when restricted to a single feature. Thus, this model is included in the study to assess its performance in a one-feature scenario.

2.4.2 *Boosting*

We delve into another category of ensemble models, specifically, those that use boosting methodologies. Our focus is primarily on eXtreme Gradient Boosting (XGBoost) [72] and Light Gradient Boosting Machine (LightGBM) [73], both of which hinge on the Gradient Boosting technique. This technique, a specialized form of boosting, employs gradient descent optimization. Notably, while the individual, so-called "weak" models in random forests are trained independently of each other, gradient boosting takes a sequential approach. Each model in the sequence aims to correct the errors of its predecessors.

Prior research shows that XGBoost can outperform other machine learning models such as Random Forest and LightGBM, particularly in forecasting gold prices [74]. Another study indicates that XGBoost also performs well in predicting short-term returns in cryptocurrencies, outperforming traditional forecasting models [75]. Furthermore, in the realm of Bitcoin price prediction, XGBoost has demonstrated superior capabilities compared to LSTM [76]. It is pivotal to note that while previous research often employed XGBoost in a multivariate context, this thesis uses it within a univariate model framework. A compelling reason for using XGBoost here is the relatively scarce comparisons available with other models in predicting various cryptocurrencies.

LightGBM has also been noticed for its good predictions in some situations. For instance, it has been found to outperform XGBoost in stock price predictions in some cases [77]. Additionally, LightGBM has shown high accuracy and robustness in forecasting cryptocurrency price trends, surpassing the performance of random forest models [78]. Regrettably,

to our current understanding, there exists no research that compares the performance of LightGBM with other time series forecasting models in the context of cryptocurrency price prediction. This knowledge gap serves as a primary motivator for incorporating this model into our thesis.

2.5 HYBRID MODELS

Hybrid models in machine learning blend two or more distinct models, leveraging the strengths of each constituent model to heighten predictive accuracy. This approach, intertwining multiple models, aims to decrease the limitations of individual models, enhancing the overall predictive prowess. We focus on the N-HiTS hybrid model, a relatively novel addition to the academic landscape [79]. The N-HiTS model builds upon the foundational N-BEATS model discussed in Section § 2.3.2. The authors of the N-HiTS model illustrate that their model surpasses the N-BEATS and ARIMA models, especially in long-horizon forecasting scenarios, using multiple datasets, including an exchange-rate dataset. Nevertheless, there is a lack of research exploring the performance of N-HiTS concerning cryptocurrency or stock price prediction, guiding our decision to incorporate this model into our thesis.

2.6 TIME SERIES DECOMPOSITION-BASED MODELS

Models based on time series decomposition employ statistical methods to break down time series data into fundamental components: trend, seasonality, and residuals. The goal of these models is to find underlying patterns and relationships in the data by identifying and isolating the trend and seasonality components. The residuals are then modeled using conventional approaches, like those outlined in Section § 2.1, and combined with the trend and seasonality components to create a complete forecast. This discussion will navigate through two distinctive frameworks: trigonometric and Bayesian.

2.6.1 *Trigonemtric*

The TBATS model uses a trigonometric framework and adeptly captures intricate patterns in the data, thereby generating accurate forecasts [80]. Although the application of TBATS for predicting financial instruments has not been expansively researched, one study focusing on cryptocurrency prices demonstrates its capability to outperform other models, including ARIMA [81]. Given the limited comparative analyses of TBATS with various time series forecasting models within the cryptocurrency realm, it finds a place in our thesis.

2.6.2 *Bayesian*

Prophet, employing a Bayesian framework, models both linear and non-linear trends within time series data [82]. The academic narrative reveals that Prophet might not consistently outperform other models such as LSTM, RNN, and ARIMA in the context of predicting stock prices [83, 84]. Yet, it demonstrated superior precision compared to ARIMA in Bitcoin price predictions in one instance [85]. Contrarily, other research presented mixed results, where ARIMA outperformed Prophet, although Prophet did manage to surpass XGBoost [86]. Another study indicated that, while Prophet lagged behind LSTM in predicting Bitcoin prices, it did perform better than ARIMA [87]. Given the conflicting results regarding Prophet's efficacy in forecasting cryptocurrencies and its relatively limited comparison with other time series forecasting models, it has been selected for exploration in this thesis.

2.7 DATA COMPARISON

This section aims to analyze the data types and variables leveraged in prior works, especially those highlighted in preceding sections, exploring specifics related to cryptocurrencies, time frames, and features. A summary table providing an overview of the data used across related papers can be found in Table 2.1.

A recurrent observation from the table is the predominant reliance on a daily time frame and a singular time interval in the majority of the research. This methodological choice is discussed in depth in Section § 1.5. Furthermore, a substantial bias towards BTC as a primary subject of cryptocurrency research is evident. In instances where multiple cryptocurrencies are examined, the selection often gravitates towards those with substantial market capitalization, a limitation explored in Section § 1.4. This is the reason we have opted for a new dataset, one that was not used by any of the mentioned papers. The previous papers did not have the desired collection of variated cryptocurrencies. Moreover, none of the papers had collected a dataset considering multiple time frames. The details of our dataset are discussed in Section § 3.1.

When examining the features used in these studies, a tendency for employing multivariate models becomes apparent, given that numerous studies use more than one feature. In contrast, this thesis opts for a univariate forecasting approach, confining the feature set to closing price data across all models. We have chosen this approach to be able to compare multiple forecasting models as not all forecasting models in this thesis are able to handle multivariate data.

Remarkably absent among the features explored in previous works is the variable of volatility. Despite its ascendant prominence in recent financial forecasting research, volatility has infrequently been combined with price forecasting studies. This gap is notable, especially as research shows that strategies including volatility can be more valuable than those that ignore it [88]. So, this thesis aims to examine how volatility affects prediction

accuracy in cryptocurrency price forecasting To do so we will not use volatility as a feature but use statistical tests to test whether volatility has a significant impact on the forecasting performance of our models.

Ref.	Forecasting Models	Cryptocurrencies	Time Frame	Timesteps	Features
[50]	GRU and LSTM	BTC, ETH, and LTC	Daily	1,250	OHLC
[51]	GRU and LSTM	BTC	Fifteen-minute	221,000	OHLCV
[52]	GRU and LSTM	BTC, ETH, and LTC	Daily	1,825	OHLC and adjusted close
[63]	ARIMA, LSTM, and N-BEATS	BTC	Daily	2,920	O
[64]	ARIMA, LSTM, and N-BEATS	ADA, BNB, BTC, ETH, LTC, and XRP	Hourly	25,560	OHCLV and 14 indicators
[87]	ARIMA, LSTM, and Prophet	BTC, ETH, and LTC	Daily	2,920	OHLCV and weighted price
[86]	ARIMA, Prophet, and XGBoost	BTC	Daily	3,379	OHLCV
[78]	LightGBM and Random Forest	BTC	Daily	180	C and multiple stock indices
[71]	LSTM and Random Forest	BTC	Daily	2,559	C and 47 explanatory variables
[70]	ARIMA, Random Forest, and RNN	ETH	Hourly	19,757	C
[47]	LSTM and RNN	BTC and ETH	Daily	2,991 and 2,160	OHLCV and market cap
[81]	ARIMA and TBATS	BTC, ETH, LTC, XMR, and XRP	Daily	2,008	C
[60]	GRU, LSTM, N-BEATS, TCN, and TFT	ETH	Hourly	18,451	C
[68]	ARIMA, GRU, LSTM, Random Forest, TCN, and TFT	BTC, ETH, LTC, XMR, and XRP	Daily	1,826	C
[76]	LSTM and XGBoost	BTC	Daily	2,991	OHLCV and market cap

Table 2.1. Summary of data used in previous studies, excluding forecasting models not relevant to the current research focus. The letters 'OHLCV' represents the standard financial market data points: Opening price, Highest price, Lowest price, Closing price, and trading Volume within a specific timeframe.

2.8 CONTRIBUTIONS

In spite of notable progress within the domain of time series forecasting, a discernible gap persists in the existing literature regarding the evaluative study of model effectiveness within the financial sector. Numerous models, such as the N-HiTS model, have been proposed in recent years; however, research into their performance, particularly in the forecasting of financial instruments, remains conspicuously sparse. Moreover, existing studies focused on the predictability of time series forecasting models in finance often limit their scope to a single asset or a single time frame, therefore not realizing a comprehensive analysis that spans across various assets and time frames.

The contributions of this thesis are threefold.

Firstly, we offer a comparative evaluation of both state-of-the-art and traditional forecasting models in the context of cryptocurrency price prediction. A comprehensive analysis of these forecasting models is undertaken, furnishing insights into their capacities for forecasting cryptocurrency prices accurately. The results in Section § 5.1 show how the selected forecasting models perform.

Secondly, we quantify the impact of data properties and market factors on the predictive performance of the chosen forecasting models. This is accomplished by calculating the models' performance and examining their significance in relation to them within corresponding periods. The outcomes shown in Section § 5.2 to Section § 5.4 cast light upon the relationship between the characteristics of our dataset and forecasting performance.

Finally, we examine how the data time span impacts predictive accuracy. In Section § 5.5.1, we assess whether expanding training data enhances the performance of our deep-learning and RNN-based models. Additionally, in Section § 5.5.2, we evaluate the effectiveness of these models in making long-term forecasts.

3

DATA

In this chapter, we outline the methods we used for collecting the data and analysis of it. Our objective is to provide a comprehensive understanding of the underlying properties and behavior of the financial time series data. We use this information to transform the data and make it fit the assumptions of our forecasting models. This chapter is organized into several subsections, each focusing on a distinct aspect of time series analysis.

First, we discuss the data collection process in Section § 3.1, detailing the sources of our datasets and the steps undertaken to ensure data consistency and reliability. Next, we examine the stationarity of the time series in Section § 3.2.1, an essential prerequisite for many forecasting models, and discuss the tests employed to determine if the series is stationary or requires further transformation.

We then delve into the autocorrelation structure of the data in Section § 3.2.2, exploring how the values within the time series are related to their past observations. Subsequently, we investigate the presence of trend and seasonality in the data in Section § 3.2.3 and § 3.2.4 respectively, as understanding these patterns can significantly improve the accuracy of forecasting models.

In Section § 3.2.5 we evaluate whether the variance of the time series remains constant over time or exhibits changes, as this has implications for the choice of appropriate models. Next, in Section § 3.2.6 we explore the stochasticity of the data, examining whether the time series data follows a specific stochastic process, such as geometric Brownian motion. Finally, Section § 3.2.7 focuses on volatility the of the data, analyzing how the price fluctuations in the financial time series data evolve over time. Understanding the influence that volatility dynamics have on forecasting models is one of the main components of this thesis.

Finally, Section § 3.3 introduces an alternative method for data transformation. We propose that this technique enhances the performance of deep-learning and RNN-based forecasting models, particularly when predicting data over shorter time frames.

3.1 DATA COLLECTION

Before beginning the data collection process, several decisions must be made. These decisions include selecting a time frame, cryptocurrencies, and cryptocurrency exchange

from which to obtain the data. Due to the vast amount of available data, it is not feasible to utilize all time frames for all cryptocurrencies in this thesis. Thus, we have selected distinct time frames of one-minute, fifteen-minute, four-hour, and daily for our analysis. Each of these time frames is significantly different from one another and can be used for different trading strategies, such as scalping, day trading, swing trading, and position trading, respectively [89].

The next decision that we need to make is the exchange since this is where we are getting the data from. We have chosen Binance[1] as an exchange as this is the most popular cryptocurrency exchange, has the highest average liquidity, and handles the most trading volume [90].

To ensure diversity in our cryptocurrency selection, we considered multiple factors such as market capitalization category, end-use, and listing date on Binance. We avoided selecting cryptocurrencies with similar market capitalization categories and purposes to prevent correlation. We also took into account the availability of data for these cryptocurrencies, considering their listing date on Binance. Ultimately, we chose twenty-one cryptocurrencies, with seven in each market capitalization category. The large-cap coins we selected are Bitcoin (BTC), Ethereum (ETH), Binance Coin (BNB), Ripple (XRP), Cardone (ADA), Dogecoin (DOGE), and Polygon (MATIC). For medium-cap, we picked Chainlink (LINK), Ethereum Classic (ETC), Stellar (XLM), Litecoin (LTC), TRON (TRX), Cosmos (ATOM), and Monero (XMR). Our small-cap selection includes VeChain (VET), Algorand (ALGO), EOS, Chilliz (CHZ), IOTA, NEO, and Tezos (XTZ).

We used the spot market for gathering the historical price data and used Tether (USDT) as the quote asset. We chose USDT as the quote asset because it is the most widely used stablecoin currently [91].

To ensure that the data is reliable, we gathered the data from the Binance API[2], using our own Python code. We collected 1,000 data points for each combination of cryptocurrency and time frame, resulting in a total of 84 datasets. Since we required at least 1,000 days of trading data for each cryptocurrency, the coins we selected tended to be "older". We also used the most recent data for this gathering process, meaning that the first data point is on the 15th of April and goes back 1,000 timesteps for each dataset.

3.2 DESCRIPTION OF DATA PROPERTIES

In this section, we analyze the inherent properties of the time-series data central to our study, aligning with the elements specified in *RQ*2*a*: stationarity, autocorrelation, trend, seasonality, heteroskedasticity, and stochasticity. We approach each property systematically: initially establishing its definition, then clarifying its importance in time-series forecasting, with an emphasis on cryptocurrency contexts. Subsequently, we detail the techniques used to assess these properties, sharing insights derived from our data alongside

1 https://www.binance.com
2 https://www.binance.com/en/binance-api

comparisons with existing research findings. Moreover, we explain the targeted data transformations implemented to counteract the effects of any unfavorable properties, aiming to improve the reliability and precision of our forecasts.

3.2.1 *Stationarity*

A stationary time series refers to one where the mean and variance remain constant over time, indicating that the properties of the series are independent of the time of observation [92]. When these properties remain stable, it becomes easier for models to capture patterns and make accurate forecasts. It is crucial to ensure that time series data is stationary before applying models to it because most time series models, such as ARIMA and related models rely on it. Non-stationary data can lead to spurious results, unreliable parameter estimates, and poor forecast performance [14, 35].

We test for stationarity using the widely recognized Augmented Dickey-Fuller (ADF) test [93]. This test is a common choice for examining stationarity in time series data, especially in financial data, due to its capability to accommodate higher-order autoregressive processes. Although the ADF test is frequently used for determining stationarity, it has limited power in detecting unit roots in small samples or near-unit root series. As a result, we also employ the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [92] to complement the ADF test, as they have opposing null hypotheses, with the KPSS test's null hypothesis being stationarity. By utilizing both tests, we can identify various types of non-stationarity and obtain a more comprehensive evaluation of whether the series is stationary.

We applied both tests to all 84 datasets in our study. The ADF test results show that only four datasets have a p-value below 0.05, suggesting stationarity. However, three out of these four datasets have a p-value close to 0.05. Similarly, the KPSS test results indicate that only one dataset is stationary, with a small p-value of 0.1. Based on these findings, we conclude that the data is predominantly non-stationary.

To transform non-stationary time series into stationary ones, various methods can be employed, with differencing being a prevalent approach [14]. Differencing calculates the difference between consecutive observations, stabilizing the time series mean and mitigating or eliminating trends and seasonality. In the context of financial instruments, differencing results in returns, which represent the price difference per time step. When we apply the ADF test to the returns, all combinations yield a p-value below 0.05. Similarly, the KPSS test results on the returns confirm stationarity. These findings suggest that utilizing cryptocurrency price returns yields a stationary time series, which is more suitable for our time series forecasting analysis.

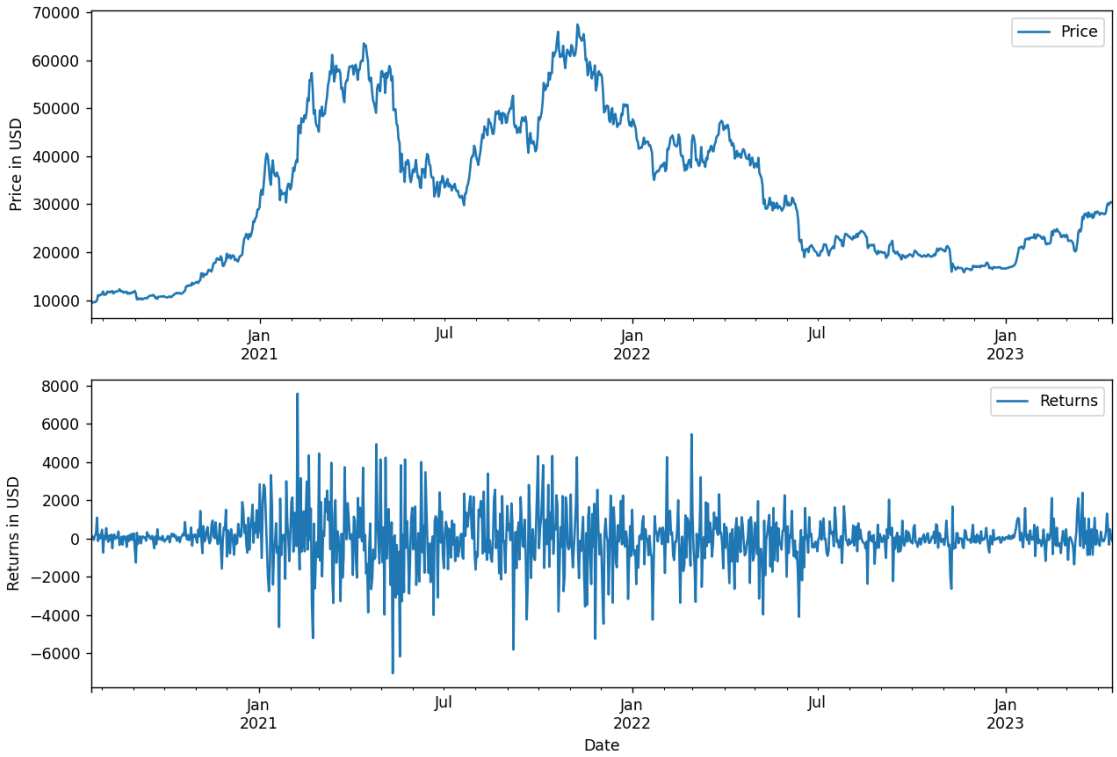

Figure 3.1. Visualization of Bitcoin price fluctuations over the last 1000 days, with a comparison between original price data and the data after differencing.

The graphs in Fig. 3.1 show what happens when we calculate the returns of the daily price data of Bitcoin. The graphs show the original price data of Bitcoin from July 2020 until March 2023 and the results after differencing the price data. We can see that it has decreased the range of values on the y-axis and is now zero-centered.

In conclusion, ensuring that the time series data is stationary is an important procedure for accurate modeling and forecasting. We employed both the ADF and KPSS tests and found that the majority of our datasets are non-stationary. After differencing, we were able to transform the non-stationary price series into stationary returns. This transformation creates a more useful dataset for time series forecasting, which we use in the following sections.

3.2.2 *Autocorrelation*

Autocorrelation occurs when the values of a time series are correlated with their own past values at specific time lags, implying that a variable's current value is influenced by its previous value at a certain timestep [94]. Consequently, a predictable pattern may emerge in the data, with similar values often appearing close together in time. Similarly to stationarity, numerous forecasting models rely on the assumption that the data and residuals exhibit no autocorrelation [14]. Moreover, identifying and addressing autocorrelation was shown to contribute to enhancing the efficiency and accuracy of time series forecasting

models [95]. Prior research has shown that if the autocorrelation is addressed by making machine-learning models aware of it, it can improve efficiency and performance [95, 96]. Research has also shown that there is an inverse relationship between autocorrelation and volatility for European stocks [97]. Therefore, we investigate if this relationship also exists for the selected cryptocurrencies in Section § 5.2.1. Thus, dealing with autocorrelation improves efficiency and accuracy in forecasting, this is particularly relevant for cryptocurrencies, as previous research demonstrated that a significant number of cryptocurrencies display signs of autocorrelation, especially the groups with medium to low liquidity [98].

To assess autocorrelation in our data, we used the Durbin-Watson [99], Ljung-Box [100], and Breusch-Godfrey [101] tests. These tests have distinct properties: the Durbin-Watson test is designed to detect first-order autocorrelation in a regression model, while the Ljung-Box and Breusch-Godfrey tests are capable of identifying higher-order autocorrelation in the residuals of a time series model. The methods for identifying higher-order autocorrelation differ, so by using both tests, we increase the reliability of our autocorrelation detection.

We performed all three tests on each dataset, observing autocorrelation in every instance. For the Ljung-Box and Breusch-Godfrey tests, we examined multiple lags ranging from 1 to 100 and consistently found significant p-values. Similarly, the Durbin-Watson test revealed statistically significant autocorrelation across all datasets. Based on these results, we conclude that every combination of time frame and cryptocurrency in our dataset demonstrates substantial autocorrelation. However, when we test on our datasets after differencing we still see some significant results, meaning that not all autocorrelation has been removed. Therefore we apply another method to remove more autocorrelation from the data. We do this by the log returns, which can be calculated by taking the difference of the natural logarithm of consecutive prices. Log returns are preferred in finance as they are more symmetric [102] and additive over time [103]. Therefore, this is a common preprocessing approach for forecasting financial time series [104]. The results shown in Table 3.1 show the test results and the corresponding dataset.

Autocorrelation Test	Prices	Returns	Logarithmic Returns
Durbin-Watson	84	1	0
Breusch-Godfrey	84	51	41
Ljung-Box	84	58	54

Table 3.1. Results of autocorrelation tests (Durbin-Watson, Breusch-Godfrey, and Ljung-Box) applied to different datasets: Prices, Returns, and Logarithmic Returns. Values in the table represent the test statistics, indicating the presence of autocorrelation in the time series data.

Utilizing logarithmic returns yields the best results across all tests. Employing price data for autocorrelation tests would lead to all datasets exhibiting statistically significant autocorrelation. However, when using returns, the number of datasets with autocorrelation is reduced by almost half for both the Breusch-Godfrey and Ljung-Box tests, and the Durbin-Watson test identifies only one dataset. Further transforming the data leads to even better outcomes. The Durbin-Watson test reports no datasets with autocorrelation, while the Breusch-Godfrey and Ljung-Box test results are slightly better when using the logarithmic returns. These results represent averages for all lags tested.

In this section, we investigated the presence of autocorrelation in our cryptocurrency dataset, as it plays a crucial role in the efficiency and accuracy of time series forecasting models. By employing various tests, including the Durbin-Watson, Ljung-Box, and Breusch-Godfrey tests, we observed autocorrelation in all datasets across different time frames.

Our analysis revealed that transforming the data by calculating returns and logarithmic returns successfully reduced autocorrelation. While some higher-order autocorrelation remained, the Durbin-Watson test showed no signs of first-order autocorrelation after these transformations. The Ljung-Box and Breusch-Godfrey tests also indicated a decrease in autocorrelated datasets. The persistence of higher-order autocorrelation suggests that more complex models might be better suited for capturing these higher-order relationships.

Our findings align to some extent with previous research, but we identified fewer autocorrelated datasets. This discrepancy could be attributed to our use of logarithmic returns instead of solely returns. However, residual autocorrelation persists in our dataset, the implications of which we will examine in Section § 5.2.1.

3.2.3 *Trend*

In the context of time series, trends represent long-term movements or patterns that are not the cause of random fluctuations, seasonality, or cyclical behavior [14]. These trends can be caused by different factors, such as economic growth or decline, or government regulations. It is generally believed that strong trends upwards are marked by decreases in volatility, whereas strong trends downwards generally show an increase in volatility [105]. A trend in the data can have significant implications for time series forecasting and analysis. For instance, if a trend is not accounted for, the forecasts generated by a model may be systematically biased and inaccurate. Prior research has shown that this is especially true for machine-learning-based forecasting models [106].

As highlighted before, cryptocurrencies exhibit unique price trends, characterized by their volatile and often unpredictable nature [7]. Consequently, we anticipate observing a considerable presence of trends in our datasets prior to detrending.

In evaluating the presence of a significant trend in our data, we employ the modified Mann-Kendall test as developed by Hamed and Rao (1997)[107], recognizing its suitability for autocorrelated data. Further, we explore three additional variations of the Mann-Kendall

test, as proposed by Yue and Wang (2002, 2004)[108, 109]. The ensuing plot elucidates the results derived from the application of all the trend tests.

Normally the Mann-Kendall test is used for proving if there is a significant trend in the data, as this is a common non-parametric trend test. However, it is proven not to work well for autocorrelated data as shown by Hamed and Rao (1997) [107]. The same authors showed that their modified version is more suitable for autocorrelated data. For that reason, we used their modified variant to test if there is a significant trend in our data. We also applied three other variations of the Mann-Kendall test proposed by Yue and Wang (2002, 2004) [108, 109]. The first test variation they proposed uses pre-whitening, which eliminates or reduces short-term stochastic persistence, named Pre-Whitening [110]. The second test removes the trend and then applies pre-whitening, denoted as Trend Free. The third test variation uses an effective sample size (ESS) to eliminate the effect of serial correlation on the Mann-Kendall test, denoted as ESS. Table 3.2 shows the results of all the trend tests.

Dataset	No Trend	Decreasing Trend	Increasing Trend	Trend Test
Price Data	56	19	9	Hamed Rao
	49	27	8	Pre-Whitening
	45	36	3	Trend Free
	40	27	17	ESS
Logarithmic Returns	72	7	5	Hamed Rao
	81	2	1	Pre-Whitening
	81	2	1	Trend Free
	48	20	16	ESS

Table 3.2. Comparison of detected trends in price data and logarithmic returns using different trend tests.

It is clear that the data is mostly split between no trend and a decreasing trend for the price data. Only a few datasets show an increasing trend. This shows that applying the logarithmic return to the data also helps with removing most trends. Only the test using ESS shows almost no change. The other tests display a clear increase in no trend.

Time Frame	No Trend	Decreasing Trend	Increasing Trend
One-Minute	81	3	0
Fifteen-Minute	77	1	6
Four-Hour	61	0	23
Daily	63	21	0

Table 3.3. Trend analysis of logarithmic returns across various time frames. The results in this table represent a compilation of outcomes from the four trend tests applied.

From Table 3.3 it is evident that the one-minute time frame exhibits the fewest trends. This observation aligns with the understanding that lower time frames tend to be noisier and demonstrate fewer patterns [8]. Our results further support this notion, as higher time frames, such as the four-hour and daily, display more trends compared to the one-minute and fifteen-minute time frames.

To sum up, our initial analysis of the price data indicated that the majority of datasets presented no trend or a decreasing trend, with only a few demonstrating an increasing trend. These findings align with our expectations, as prior research has suggested that cryptocurrencies often exhibit strong trends.

Despite this, we noticed a decrease in trends after applying logarithmic returns transformation to the data, as evidenced by the higher number of datasets classified as "no trend" in the test results. This transformation not only helps reduce the impact of trends but also improves the statistical properties of the data, making it more appropriate for time series forecasting. The Modified Yue Wang test was the only exception, showing minimal change. Additionally, our results revealed that higher time frames exhibited more signs of trends compared to lower time frames, which aligns with the findings of previous research. Finally, by detecting and addressing trends in the data, we can reduce potential biases in our predictions and ultimately improve the overall accuracy and reliability of our time series analysis.

3.2.4 *Seasonality*

Seasonality refers to regular and predictable fluctuations or patterns that recur over a fixed period within time series data [14]. These patterns are typically caused by external factors or systematic influences, such as weather, holidays, or other calendar-related events. For example, retail sales often exhibit seasonality, with an increase in sales during holiday seasons and a decrease in sales during other times of the year. Testing for seasonality in financial time series data is vital, as it helps identify and account for regular patterns or cycles that recur over specific time intervals. Recognizing and addressing seasonality enhances the performance of time series forecasting models by aligning with model assumptions and more accurately capturing underlying data patterns [111]. Consequently, assessing seasonality is a crucial step in the analysis and modeling of financial time series data to ensure reliable and precise forecasts. Research has found a correlation between seasonality and volatility, especially during the month of Ramadan on the Arabian stock market [112]. It could be the case that the cryptocurrency market also shows this effect during international holiday periods, for instance during New Year's Eve or on weekends. Just like trends, machine-learning-based forecasting models perform worse when the time series data exhibit seasonality [106]. Therefore, finding a way to remove seasonality from our data could increase performance.

There are a few methods for testing seasonality in the data, one common approach is to use seasonal decomposition of the time series (STL) [113]. Normally this would produce mul-

tiple plots, showing the trend, seasonality, and residuals. Unfortunately, it is not feasible to inspect the seasonality plot of each dataset, therefore we use strength calculation suggested by Hyndman and Athanasopoulos (2018) [14]. Their formula quantifies the seasonality that is produced by the STL, so we can see which datasets have a strong seasonality at a specific lag. The seasonal strength ranges from zero to one, with the value of one meaning the strongest seasonality. We tested each dataset on seasonality, depending on the time frame we tested for hourly, daily, weekly, monthly, quarterly, and yearly seasonality.

Time Frame	Pattern	Price Data	Logarithmic Returns
One-Minute	1 hour	0.382	0.359
	2 hours	0.439	0.392
Fifteen-Minute	1 hour	0.292	0.382
	2 hours	0.274	0.399
Four-Hour	1 day	0.256	0.306
	7 days	0.339	0.368
	30 days	0.493	0.560
Daily	7 days	0.285	0.342
	30 days	0.391	0.357
	90 days	0.337	0.357
	1 year	0.786	0.810

Table 3.4. Seasonal strength in price data and logarithmic returns across various time frames and patterns.

In Table 3.4 we see that yearly seasonality possesses the highest strength on the price data. Upon examining the test results, we observed that all seasonality strength values exceeding 0.75 were associated with yearly seasonality in the daily time frame. The logarithmic returns yield results that are comparable to those obtained from the price data, with the yearly pattern on the daily time frame remaining the most pronounced seasonal trend. Some minor differences emerge, such as the average seasonality strength remaining nearly the same for all patterns except for the 30-day pattern on the four-hour time frame. For values greater than 0.5, other time frames also emerge, with monthly seasonality becoming more prevalent in the four-hour time frame. Table 3.5 shows the occurrence of various time frames and seasonality for strengths exceeding 0.5.

Pattern and Time Frame	Price Data	Logarithmic Returns
Two-hourly (One-Minute)	4	0
Monthly (Four-Hour)	11	20
Yearly (Daily)	21	21

Table 3.5. The number of datasets that exhibit a seasonal pattern of strength greater than 0.5.

The yearly pattern clearly stands out as the strongest among the tested seasonality patterns. However, out of the 231 results obtained from testing 11 different patterns on our dataset, only 36 exhibit a seasonality score greater than 0.5, accounting for approximately five percent. Thus, we can conclude that while some seasonality is present in the original data, it is not substantial. This correlates with the findings of previously mentioned research [5, 114]. Upon examining the logarithmic returns, we find that the seasonal patterns with a strength greater than 0.5 differ from the original data. The monthly and yearly patterns are almost evenly distributed, while the two-hourly pattern has vanished. The number of results with a strength exceeding 0.5 has also increased to 41, constituting approximately 18 percent of the dataset. Given that our research includes 21 cryptocurrencies, these results indicate that nearly every coin in our dataset exhibits some degree of monthly and yearly seasonality. Consequently, the stationarity of the data does impact our approach to seasonality testing.

Conducting seasonality tests is essential to ensure the accuracy and reliability of forecasts generated by time series models. Although previous research suggested that cryptocurrencies typically do not exhibit strong seasonality, we performed tests to confirm this characteristic within our datasets.

Our analysis, which included both price data and logarithmic returns, revealed that the most discernible pattern was yearly seasonality on the daily time frame, although not overwhelmingly strong. The stationarity of the data impacted our assessment of seasonality, as nearly every coin in our dataset displayed some degree of monthly and yearly seasonality. These findings differ from prior research, which reported no seasonality. This discrepancy could be attributed to our use of logarithmic returns, as opposed to solely examining price data. If we consider only the price data before transforming it, we can concur that the seasonality identified is not substantial.

In further analysis in Section § 5.2.3, we use the insights gained from this section to find out if there is any correlation between volatility and model performance. We expect that certain forecasting models would perform better on the daily time frame due to the more noticeable seasonality present in this specific time frame. However, for forecasting models that do not account for seasonality, such as ARIMA, this time frame could be more difficult to predict and lead to lower performance.

3.2.5 *Heteroskedasticity*

We also investigate the presence of heteroskedasticity, a phenomenon characterized by non-constant variance in the error terms of a regression model. Heteroskedasticity typically manifests in two forms: conditional and unconditional. Conditional heteroskedasticity refers to non-constant volatility that is related to the volatility of prior periods (e.g., daily), whereas unconditional heteroskedasticity relates to general structural changes in volatility, independent of prior period volatility [115]. Unconditional heteroskedasticity is relevant when it is possible to identify future periods of high and low volatility.

A homoskedastic model, characterized by constant error term variance, is generally preferred. In contrast, a heteroskedastic model demonstrates varying error term variance depending on the levels of the independent variables [116]. Traditional time series forecasting models, such as ARIMA, assume constant variance over time (homoskedasticity) [32]. When confronted with heteroskedastic data, these models may yield suboptimal forecasts. However, modern models, such as RNN-based models, can address heteroskedastic data. In this section, we investigate the presence of heteroskedasticity in our data, both before and after applying suitable transformations.

Testing for heteroskedasticity is particularly relevant in the context of cryptocurrency data for several reasons. First, cryptocurrency prices are known for their high volatility, which can lead to non-constant variance in the error terms [117]. Second, changing market conditions have been shown to introduce heteroskedasticity in the data [118]. This is also applicable to the cryptocurrency market, where market sentiment can shift rapidly.

To detect general (unconditional) heteroskedasticity, we applied both the Breusch-Pagan test [119] and the Goldfeld-Quandt test [120], as they employ distinct methods for identifying heteroskedasticity. The Breusch-Pagan test is based on the relationship between the squared residuals and a set of specified explanatory variables. In contrast, the Goldfeld-Quandt test involves dividing the dataset into two or more subsamples and comparing the ratio of the residual variances between these subsamples.

The Breusch-Pagan test showed that there are 81 datasets that have a p-value below 0.05, meaning that the null hypothesis of homoskedasticity is rejected and that there is evidence of heteroskedasticity. However, for the Goldfeld-Quandt test, there are only 22 datasets that showed evidence of heteroskedasticity. After transforming the data in the logarithmic returns, there are 61 datasets that showed evidence of heteroskedasticity according to the Breusch-Pagan test and 20 for the Goldfeld-Quand test. This is a small improvement compared to the price data. Table 3.6 shows the improvement of using the logarithmic returns when testing for heteroskedasticity.

Heteroskedasticity Test	Prices	Logarithmic Returns
Goldfeld-Quandt	22	20
Breusch-Pagan	81	61

Table 3.6. Number of heteroskedastic datasets according to the Goldfeld-Quandt and Breusch-Pagan test results.

Using the unconditional heteroskedasticity test results, we made an analysis of the time frames where heteroskedasticity occurred the most. The results are shown in the Table 3.7.

Time Frame	Breusch-Pagan	Goldfeld-Quandt
One-Minute	21	0
Fifteen-Minute	12	14
Four-Hour	9	6
Daily	19	0

Table 3.7. Number of heteroskedastic datasets in logarithmic returns, grouped by time frame.

It is intriguing to observe the significant differences in test results across various time frames. The Breusch-Pagan test results indicate that every cryptocurrency on the one-minute time frame exhibits unconditional heteroskedasticity, while the Goldfeld-Quandt test identifies none. Similarly, the daily time frame displays comparable discrepancies in test outcomes. However, the test results are more consistent for the fifteen-minute and four-hour time frames, suggesting a higher likelihood of cryptocurrencies within these time frames being unconditionally heteroskedastic. This information is valuable when evaluating the forecasting results for specific cryptocurrencies and time frames. The identified cryptocurrencies and time frames with heteroskedasticity are expected to be more challenging to forecast accurately, as heteroskedasticity complicates the modeling process. We analyze this expectation further in Section § 5.2.4.

Next, we are going to measure conditional heteroskedasticity in all of our datasets. Prior research has already shown that Bitcoin shows signs of conditional heteroskedasticity [121]. Unfortunately, there was no other research that clearly measured conditional heteroskedasticity in cryptocurrency. However, financial instruments are known to show conditional heteroskedasticity. Therefore, we expect to see it in most of our datasets as well.

We use Engle's Autoregressive Conditional Heteroskedasticity (ARCH) Test [122] to determine if there is conditional heteroskedasticity in the datasets. The ARCH test examines the presence of autoregressive conditional heteroskedasticity in the residuals. We only test

on the logarithmic returns, as using the price data would lead to spurious results, because of non-stationarity and trends. The results of the conducted Engle's ARCH test are as follows, 75 datasets show heteroskedasticity, and 9 show homoskedasticity. The presence of conditional heteroskedasticity remains substantial in the data, with only about 11 percent of the total data exhibiting no conditional heteroskedasticity. This is considerably less compared to the test results for unconditional heteroskedasticity. Additionally, we analyzed the results across different time frames and observed that the fifteen-minute time frame exhibited the least conditional heteroskedasticity, with five datasets classified as homoskedastic. In contrast, the other time frames had either one or two datasets that demonstrated no conditional heteroskedasticity. The results were expected, as it is very common for financial data to show conditional heteroskedasticity. We also analyze the results per time frame, these are shown in Table 3.8.

Time Frame	Heteroskedasticity	Homoskedasticity
One-Minute	20	1
Fifteen-Minute	16	5
Four-Hour	19	2
Daily	20	1

Table 3.8. Conditional heteroskedasticity in logarithmic returns, grouped by time frame.

The results show intriguing similarities to those presented in Table 3.7. Both the one-minute and daily time frames exhibit the highest levels of heteroskedasticity, consistent with the Breusch-Pagan test findings. Interestingly, the fifteen-minute time frame displays the lowest heteroskedasticity, contrasting with our earlier observations from the unconditional heteroskedasticity tests, where the fifteen-minute time frame had the highest overlap between test results.

The observed results seem to contradict our findings from the unconditional heteroskedasticity tests. In this instance, the fifteen-minute time frame should be easier to predict due to its lower heteroskedasticity. In Section § 5.2.4 we examine whether this is indeed the case and if our models perform better or worse on this particular time frame.

In conclusion, our investigation into heteroskedasticity within cryptocurrency datasets reveals the presence of both unconditional and conditional heteroskedasticity. The use of logarithmic returns reduced the presence of unconditional heteroskedasticity, but conditional heteroskedasticity remains a prevalent characteristic, especially in financial time series data like cryptocurrencies, where volatility clustering is common.

These findings have important implications for time series forecasting of cryptocurrency prices. Significant differences in test results across various time frames and the presence of heteroskedasticity complicate the modeling process and may impact forecasting accuracy. Traditional time series forecasting models, which assume constant variance over time,

may not be optimal for predicting cryptocurrency prices due to the presence of heteroskedasticity. As a result, more advanced models, such as RNN-based models, should be considered, as they are capable of handling heteroskedastic data.

3.2.6 *Stochasticity*

The Hurst exponent (H) is a measure of the long-term memory of a time series, and it can be employed to distinguish between various types of stochastic processes [123]. The value of H lies between 0 and 1, and it provides insights into the behavior of the time series. Moreover, the Hurst exponent provides a measure for predictability [124]. Research has shown that machine-learning forecasting models predict financial time series better when the Hurst exponent is large, compared to an H-value close to 0.5. To classify the H-value we use the same categories as previous research that employed the Hurst exponent for forecasting financial time series [125].

When H is close to 0.5 (between 0.45 and 0.55), the time series resembles a random walk or Brownian motion, characterized by a lack of long-term memory. In this scenario, future price changes are independent of past price changes.

If H is less than 0.45, the time series exhibits mean-reverting behavior, implying that it tends to return to its long-term average value over time. Consequently, positive price changes are more likely to be followed by negative price changes and vice versa.

If H is greater than 0.55, the time series displays persistence or long-range dependence, signifying that positive price changes are more likely to be followed by positive price changes and negative price changes by negative price changes.

We used the oldest and best-known approach for estimating H, called the rescaled range (R/S) analysis [126]. The figure below illustrates the interpretation of the Hurst exponent, categorized by each time frame. When H falls between 0.45 and 0.55, it is classified as "Brownian motion". If H exceeds 0.55, it is grouped as "Positively correlated".

Hurst Exponent Interpretation	Daily	Four-Hour	Fifteen-Minute	One-Minute
Brownian Motion	3	6	10	2
Positively Correlated	18	15	11	19

Table 3.9. Interpretation of Hurst exponent values for logarithmic returns using R/S analysis results.

Table 3.9 indicates that elements of Brownian motion are present in our datasets, particularly in the fifteen-minute time frame. In this time frame, nearly half of the dataset is classified as Brownian motion. Out of the 21 datasets exhibiting Brownian motion characteristics, almost half belong to the fifteen-minute time frame. This finding suggests that

forecasting models may face greater difficulty in predicting cryptocurrency data on the fifteen-minute time frame compared to other time frames due to the inherent randomness of Brownian motion. Unfortunately, we did not see any values below 0.45 in all of our datasets. Therefore, the interpretation of mean reverting behavior is not shown in the table.

Our findings align with prior research that showed that each cryptocurrency in their dataset had a Hurst exponent close to 0.55 or slightly higher [127]. In contrast, research by David et al. (2021) yielded different outcomes, with their Hurst indices averaging around 0.48 [128]. Such values would be indicative of Brownian motion. The discrepancy between these results may be attributed to the age of the data used in the study by David et al., which could lead to variations in the calculated Hurst indices.

In this section, we analyzed the Hurst exponent to study the long-term memory of time series data. The R/S analysis results, categorized by each time frame, show that our datasets contain elements of Brownian motion, particularly in the fifteen-minute time frame. Almost half of the datasets in this time frame exhibit Brownian motion, accounting for nearly half of the 21 datasets with Brownian motion characteristics. This finding implies that time series forecasting models may encounter increased difficulty in predicting cryptocurrency data on the fifteen-minute time frame compared to other time frames due to the inherent randomness associated with Brownian motion.

3.2.7 *Volatility*

In Section § 1.3, our methodology for computing volatility, utilizing the conventional formula for historical volatility (as delineated in Eqn. 1.1), was expounded. A crucial parameter to predetermine in this computation is the standard deviation's window size, for which several alternatives exist. For daily data, window sizes can be broadly categorized into short (7-14 days), medium (20-30 days), and long (60-90 days), each embodying its own set of merits and demerits briefly elucidated in this section.

Figure 3.2. Influence of varied window sizes on volatility computations.

As illustrated in Fig. 3.2, disparate window sizes distinctly impact volatility calculations. Short windows, while adept at rapidly mirroring recent market variations and being advantageous for ephemeral trading strategies, are also prone to noise, causing a spike in volatility estimate variability.

Conversely, a medium window provides an equilibrium between rapid responsiveness and stability, adeptly capturing prevailing market trends while mitigating noise susceptibility. This accords with the prevalent usage of the 30-day window amongst financial analysts, providing a facile monthly trading data interpretation and comparison [129–131].

A long window, on the other hand, affords a more stabilized volatility estimate by diluting short-term fluctuations and noise, which proves advantageous for long-term investment strategies and portfolio management, albeit with reduced responsiveness to recent market shifts.

For our volatility calculations, we have selected the medium window, owing to its widespread acceptance and its position as a compromise between the short and long windows. Subsequently, our objective is to classify volatility into three categories: low, normal, and high. To do so, we use the 25th and 75th percentiles to classify a cryptocurrency's volatility as low or high, respectively. The concept of using percentiles to classify volatility in finance has been explored previously in research investigating the relationships between implied volatility indices and stock index returns[132]. However, we will only use three categories instead of the twenty categories that previous research utilized. Fig. 3.3 depicts

the results of this approach using daily time frames, with the light gray lines representing the volatility of each cryptocurrency.

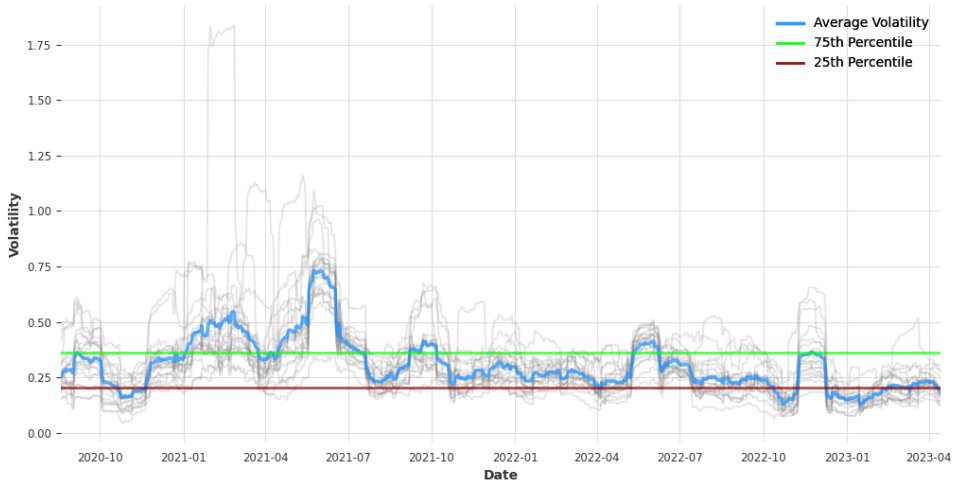

Figure 3.3. The 25th and 75th percentiles of overall volatility on the daily time frame.

The percentile values facilitate the establishment of thresholds for classifying a cryptocurrency's daily volatility as high or low. Considering that volatility generally diminishes on shorter time scales, we must perform this calculation for each time frame under investigation. The values presented in Table 3.10 reveal the outcomes of the 25th and 75th percentile for each time frame. These values will guide our investigation into how the categorization of volatility in our datasets influences model performance. Using these values, we can classify our training and testing sets and investigate how certain combinations of training and testing volatility can influence performance. As previously mentioned, this topic has not been researched extensively. Thus, most of our methods and predictions cannot be grounded in prior research. Nonetheless, we hypothesize that a mismatch between training and testing volatility negatively influences the forecasting performance of the models.

Time frame	25th percentile	75th percentile
One-Minute	0.003	0.006
Fifteen-Minute	0.009	0.017
Four-Hour	0.053	0.095
Daily	0.206	0.363

Table 3.10. The 25th and 75th percentiles for each time frame.

In summary, the exploration into the impact of volatility on cryptocurrency datasets underscores the intricate nature of selecting appropriate window sizes for volatility calculations

and classifying volatility levels. The medium window size, which balances responsiveness and stability, was employed in our calculations, while two approaches were considered for categorizing volatility into distinct classifications. Although the juxtaposition of cryptocurrency volatility against the TOTAL market index and percentile-based categorization each presented their own merits and challenges, they facilitated a deeper understanding and categorization of the inherent volatility in the examined datasets. This exploration sets the stage for an in-depth investigation into the influence of various volatility levels on forecasting model performance, a topic that will be thoroughly examined in Section § 5.3.1.

3.3 EXPERIMENT DATASET

To enhance the robustness and predictive accuracy of our time series forecasting models, we introduced a third dataset featuring scaled logarithmic returns, in addition to the raw price data and unscaled logarithmic returns. This dataset employs a MinMax scaler, specifically configured with a feature range of [-1,1], to normalize the logarithmic returns [133].

Our preliminary experiments revealed that models trained on unscaled logarithmic returns at a one-minute time frame frequently underperformed. These models exhibited difficulties in discerning the intricate patterns within the high-frequency data, leading to a substantial number of inaccurate forecasts. We hypothesize that the small numerical values of the logarithmic returns at this granular time frame may be contributing to numerical instability and scale sensitivity, thereby undermining the models' predictive capabilities.

To mitigate these challenges, we applied a MinMax scaler to the logarithmic returns. This transformation standardizes each feature to lie within a predefined range, in this case, [-1,1]. This range was selected for its symmetry around zero, which is crucial for financial data that can exhibit both gains and losses. Such scaling ensures equitable treatment of positive and negative returns in terms of magnitude while maintaining their directional attributes. This choice is also supported by existing literature in the field of financial time series forecasting [134]. To be able to compare the models' performance on this dataset with the logarithmic returns we scaled the output of the forecasting models to a logarithmic scale.

We conduct the same tests as those mentioned in the previous section on the scaled dataset to maintain consistency in our analysis. Initially, our stationarity tests yield results that are identical to those from the logarithmic returns dataset, indicating that the process of scaling the data appears not to affect the outcomes of these stationarity tests. However, a significant observation is the marked reduction in autocorrelation in comparison to the logarithmic returns. Specifically, there is a decrease in 34 datasets exhibiting autocorrelation according to the Ljung-Box test results, and 36 fewer as per the Breusch-Godfrey test outcomes. This suggests that these specific tests are sensitive to the manner in which our data is transformed. Interestingly, this reduction in autocorrelation could potentially

enhance the effectiveness of models such as ARIMA [32], which rely on the assumption of non-autocorrelation.

Further analysis of the trend test results, presented in Table 3.11, reveals that contrary to the original dataset, the majority of trend tests indicate an absence of a discernible trend in this scaled dataset. This underscores the effect that data scaling has on the trend tests employed in our study.

Dataset	No Trend	Decreasing Trend	Increasing Trend	Trend Test
Logarithmic Returns	72	7	5	Hamed Rao
	81	2	1	Pre-Whitening
	81	2	1	Trend Free
	48	20	16	ESS
Scaled	84	0	0	Hamed Rao
	84	0	0	Pre-Whitening
	84	0	0	Trend Free
	69	9	6	ESS

Table 3.11. Results of trend tests performed on logarithmic returns and scaled data.

The outcomes of the seasonal strength test, displayed in Table 3.12, reveal a general rise in seasonal strength across nearly all observed patterns. This finding stands out because it is the sole instance where there is an uptick in the property being measured. One possible explanation for this could be that scaling tends to amplify specific values in the time series, making them more prevalent than they might naturally occur. An increase in seasonal strength potentially aids certain forecasting models that thrive on identifying seasonality and adjusting their predictions accordingly.

Time Frame	Pattern	Logarithmic Returns	Scaled
One-Minute	1 hour	0.359	0.402
	2 hours	0.392	0.583
Fifteen-Minute	1 hour	0.382	0.368
	2 hours	0.399	0.477
Four-Hour	1 day	0.306	0.330
	7 days	0.368	0.411
	30 days	0.560	0.773
Daily	7 days	0.342	0.340
	30 days	0.357	0.374
	90 days	0.357	0.484
	1 year	0.810	1.000

Table 3.12. Comparison of average seasonality patterns between logarithmic returns and scaled data.

We also observe a reduction in datasets exhibiting unconditional heteroskedasticity, as shown in Table 3.13. A similar trend is evident in conditional heteroskedasticity, which diminishes from 75 to 24 datasets when comparing the scaled data to logarithmic returns. This decline in heteroskedasticity may favor forecasting models that assume a constant variance over time like ARIMA [32].

Heteroskedasticity Test	Logarithmic Returns	Scaled
Goldfeld-Quandt	20	0
Breusch-Pagan	61	6

Table 3.13. Counts of heteroskedastic datasets identified by each heteroskedasticity test.

The outcomes of our stochasticity tests remain unchanged, indicating that this transformation of the data does not affect the computation of the Hurst exponent. Through the use of this scaled logarithmic returns dataset, our objective is to mitigate concerns associated with numerical instability and the small scale of the logarithmic returns. Furthermore, this dataset may enhance the outcomes of various forecasting models, as the test results suggest it could be a better dataset relative to the logarithmic returns. Nonetheless, the primary intention behind employing this dataset is to explore whether scaling the data is more advantageous on shorter time frames. We investigate this further in Section § 5.1.

4

METHODS AND MODEL EVALUATION

This chapter details the methodology of this thesis. Section § 4.1 introduces the forecasting models used, explains data partitioning, and our approach for one-step-ahead forecasting with the sliding window method. Section § 4.2 delves into the vital process of hyperparameter fine-tuning for each model, with specific optimization search spaces detailed in subsections § 4.2.1 to § 4.2.11. Section § 4.3 evaluates the models, starting with the criteria, setting comparative standards, and exploring the impact of various factors on predictive accuracy. Finally, Section § 4.4 outlines the technical resources employed.

4.1 MODELS AND DATA PARTITIONING

In our experiment, we use a majority of the models discussed in Chapter 2, including ARIMA, RNN, LSTM, GRU, TCN, N-BEATS, TFT, Random Forest, XGBoost, LightGBM, N-HiTS, TBATS, and Prophet. We employ the Darts library in Python to implement these forecasting models [135].

In this study, we employ the dataset compiled and detailed in Section § 3.1 to train our models. Adhering to the methodology outlined by Borovykh et al. [136], we partition the data into distinct, non-overlapping periods, each comprising 75 percent training data and 25 percent testing data. This division of the time series into multiple periods facilitates a more nuanced understanding of volatility during specific intervals. To gauge each model's performance, we implement one-step-ahead forecasting, a critical aspect of time series forecasting, which assesses the model's capability to predict the immediate future data point based on current and historical data.

The determination of the number of periods necessitates careful consideration of several factors. Primarily, each period should encapsulate a unique phase of volatility, aligning with our objective to investigate the influence of volatility on model performance. Additionally, the volume of data within a single period must be taken into account; opting for a larger number of periods would result in each period representing a smaller partition of the dataset. Upon examination of the historical volatility plots of our datasets, we concluded that an optimal division would consist of five periods.

Given the high volatility characteristic of cryptocurrency markets, we adopt a sliding window (or rolling window) approach. This method, more prevalent in this type of research

than the expanding window approach [137, 138], effectively captures the rapidly changing dynamics of the market and provides a more accurate account of recent price fluctuations. As we utilize one-step-ahead forecasting, the slide (or lag) is set to 1. The window size, determined by the five periods and the training-test split within each period, is set to 372, yielding 124 test samples per period. For validation, we allocate 10 percent of the training set.

4.2 HYPERPARAMETER OPTIMIZATION

Our methodology involves conducting hyperparameter optimization using the Ray Tune framework [139]. We optimize each dataset and model, with the optimization process focused on enhancing the model's performance on the validation set for a single period. For every dataset, we execute 20 trials of hyperparameter tuning, leveraging the Asynchronous Successive Halving (ASHA) hyperparameter optimization algorithm [140]. This algorithm has been demonstrated to surpass existing state-of-the-art hyperparameter optimization methods, allowing us to spend less time and resources on the hyperparameter tuning process. Additionally, we employ the cutting-edge search algorithm Heteroscedastic Evolutionary Bayesian Optimisation (HEBO) [141] to identify the most effective combination of hyperparameters. This algorithm has proven to outperform other existing black-box optimizers in machine-learning hyperparameter tuning tasks.

Table 4.1 presents the hyperparameters that require optimization for each machine-learning model examined in our research. The selection of the hyperparameter search space is informed by the findings of the study conducted by Oreshkin et al. (2021) [142].

Hyperparameter	Values
Input Chunk Length (lookback period)	1, 3, 6, 9, 12, 24
Epochs	25, 50, 75, 100
Batch Size	16, 32, 64, 128, 256
Dropout	(0.01, 0.5)

Table 4.1. The hyperparameter search space for all machine-learning models.

The tables below show the search space for model-specific hyperparameters. The first five models all utilize regression, so they do not use the hyperparameters specified in table Table 4.1.

4.2.1 *ARIMA*

In our quest to identify the optimal hyperparameters for the ARIMA model, we employed the Auto ARIMA model as implemented in the StatsForecast library [143]. This approach

mirrors the methodology utilized by Yenido et al. (2018) [85], thereby ensuring a robust and proven foundation for our analysis. Table 4.2 shows our search space for the ARIMA hyperparameters.

Hyperparameter	Minimum	Maximum
p	0	5
q	0	5
d	0	5
P	0	5
Q	0	5

Table 4.2. The hyperparameter search space for ARIMA.

4.2.2 *Random Forest*

In formulating the optimal search space for the Random Forest model, we leveraged the insights from the study conducted by Ghosh et al. (2022) [144]. Table 4.3 shows the values for each hyperparameter that we use for the random forest.

Hyperparameter	Values
Lags	1, 7, 14, 30
Estimators	10, 100, 250, 500, 1000
Maximum Depth	2, 4, 8, 10, 12, None

Table 4.3. The hyperparameter search space for random forest.

4.2.3 *Extreme Gradient Boosting (XGBoost)*

The selection of hyperparameters for the XGBoost model is guided by the research conducted by Wang and Ni (2019) [145]. In Table 4.4 we show the hyperparameter search space for the XGBoost model. Values denoted with parentheses mean that a range of values are being used.

Hyperparameter	Values
Lags	1, 7, 14, 30
Subsample	(0.8, 1)
Gamma	(0, 0.02)
Column Sample By Tree	(0.8, 1)
Minimum Child Weight	0, 2, 5, 7, 10
Maximum Leaves	0, 10, 25, 50, 100, 200
Maximum Depth	None, 5, 10, 20, 30

Table 4.4. Hyperparameter search space for the XGBoost model.

4.2.4 *LightGBM*

To determine a useful search space for our LightGBM model we used prior financial research that used this model [146, 147]. Table 4.5 shows the possible hyperparameters for tuning the LightGBM model.

Hyperparameter	Values
Lags	1, 7, 14, 30
Maximum Leaves	31, 100, 200, 400, 600
Estimators	50, 80, 100, 150
Maximum Depth	-1, 0, 40, 80
Minimum Child Samples	0, 2, 5, 7, 10
L1 Regularization	(0, 10)
L2 Regularization	(0, 0.1)

Table 4.5. The hyperparameter search space for LightGBM.

4.2.5 *Prophet*

The hyperparameter space for Prophet, shown in Table 4.6, is based on the documentation provided by Facebook [148] and prior research [149, 150].

Hyperparameter	Values
Changepoints	10, 25, 50, 100
Changepoints Range	0.4, 0.6, 0.8, 0.9
Changepoint Prior Scale	0.001, 0.01, 0.1, 0.5, 1
Seasonality Prior Scale	0.01, 0.1, 1.0, 10.0
Seasonality Mode	Additive, Multiplicative

Table 4.6. The hyperparameter search space for Prophet.

4.2.6 *TBATS*

TBATS is a relatively new model in the field, and consequently, there is a scarcity of prior research dedicated to its hyperparameter optimization. However, the TBATS model's design inherently involves fewer configurable hyperparameters, which simplifies the optimization process despite the lack of existing research as a reference. The implementation of TBATS that was used automatically determines the best hyperparameters during evaluation, so we do not need to run hyperparameter tuning trials, similar to AutoARIMA. The model automatically finds the suitable hyperparameters shown in Table 4.7.

Hyperparameter	Values
Use Box-Cox	True, False
Use Trend	True, False
Use Damped Trend	True, False
Use ARMA errors	True, False

Table 4.7. The hyperparameter search space for TBATS.

4.2.7 *N-BEATS*

The hyperparameter search space for the N-BEATS model shown in Table 4.8 is grounded in the research conducted by Oreshkin et al. (2021) [142].

Hyperparameter	Values
Layers	2, 3, 4
Blocks	1, 2, 3, 5, 10
Width	256, 512, 1024

Table 4.8. The hyperparameter search space for N-BEATS.

4.2.8 *RNN-based models*

Drawing upon the insights from previous studies on the hyperparameter optimization of RNN-based models for financial forecasting, we have delineated the following search space for the RNN, GRU, and LSTM models as shown in Table 4.9 [151, 152].

Hyperparameter	Values
Size	16, 32, 64, 128
Layers	1, 2, 3, 4
Training Length	225, 50, 75, 100

Table 4.9. The hyperparameter search space for RNN-based models.

4.2.9 *TCN*

The selection of the search space for our TCN model is shown in Table 4.10. This selection is informed by the default parameters provided in the Darts library, complemented by insights from existing research [153, 154].

Hyperparameter	Values
Kernel Size	2, 3, 5, 7, 9
Filters	3, 8, 11, 16, 24, 32
Dilation	2, 4, 8, 16, 32
Layers	0, 2, 4, 6

Table 4.10. The hyperparameter search space for TCN.

4.2.10 *TFT*

The determination of the search space for the TFT model as shown in Table 4.11 is informed by insights gleaned from previous studies [155, 156].

Hyperparameter	Values
Hidden Size	2, 4, 10, 25, 50
LSTM Layers	1, 2, 3, 4
Attention Heads	1, 2, 3, 4
Hidden Continous Size	2, 4, 8, 10, 12

Table 4.11. The hyperparameter search space for TFT.

4.2.11 *N-HiTS*

Given the recent introduction of the N-HiTS model in the literature, there is a lack of prior research specifically focused on its hyperparameter optimization. Consequently, we have adopted the model's default values as our starting point and established the following search space. As mentioned in Section § 2.3.2 the model is built upon the N-BEATS model, thus, providing a familiar framework for our optimization efforts as shown in Table 4.12.

Hyperparameter	Values
Stacks	2, 3, 4
Blocks	1, 2, 3, 5, 10
Layers	1, 2, 3, 4
Layer Width	256, 512, 1024

Table 4.12. The hyperparameter search space for N-HiTS.

4.3 MODEL PERFORMANCE COMPARISON

To compare the performance of forecasting models, it's essential to establish uniform evaluation criteria. We aim to assess models both within the same time frame and across diverse periods. Additionally, determining the impact of data properties and market factors requires careful consideration. The subsequent sections detail our approach to these evaluations.

4.3.1 *Performance Metrics*

To evaluate the performance of the forecasting models, we employ the Root Mean Square Error (RMSE) metric. RMSE is a widely accepted and utilized metric in the realm of time series forecasting [136, 157, 158]. By calculating the RMSE for each model, we aim to be able to compare the predictive performance of our selection of forecasting models. We also use it to investigate the influence of data properties and market factors on this metric, which we discuss more in-depth in Section § 4.3.3. Furthermore, the RMSE serves as a determinant for identifying the most optimal hyperparameters for each model.

However, the application of RMSE to our data presents a limitation. Lower time frames tend to yield smaller RMSE values due to the smaller changes they exhibit compared to higher time frames, where price fluctuations within a single time step can be substantial. Consequently, RMSE can only be used to compare performance within the same time frame.

4.3.2 *Baseline Model*

The ARIMA model serves as our baseline for evaluating the performance of other models. As discussed in Section § 2.1, ARIMA is a widely recognized baseline model for time series forecasting. Additionally, we compare our results with those from recent studies that employed similar models for predicting analogous cryptocurrencies. It is crucial to note that comparisons are only viable with recent research, given that older data may significantly deviate from our current dataset.

By using ARIMA as a baseline, we can compare a model's performance across time frames by calculating the performance difference relative to ARIMA, expressed as a percentage. We perform this analysis in Section § 5.4.

4.3.3 *Influence of Data Properties, Market Factors, and Data time Span*

To evaluate how data properties and market factors affect model predictiveness, we execute several tests. Our initial analysis focuses on whether a property significantly influences volatility, discussed in Sections § 5.2 and § 5.3.2.

Subsequently, we conduct statistical tests to determine any notable effects on the RMSE across all sections in Chapter 5. For analyses involving discrete variables, such as autocorrelation where datasets are categorized as autocorrelated or non-autocorrelated, we apply the Mann-Whitney U test. Conversely, for continuous variables, a regression test is employed. When both variable types are present, we utilize both tests, as is the case for volatility and market capitalization.

Concerning volatility, we also examine whether inconsistencies between training and testing volatility classes result in diminished model performance, with findings presented in Section § 5.3.1.

Lastly, we assess the influence of data duration in Section § 5.5, exploring how expanding training data impacts our deep-learning and RNN-based models. The efficacy of these models for long-term forecasting is analyzed in Section § 5.5.2 by prolonging the testing phase and studying the subsequent effects on model performance while considering the volatility experienced during these testing intervals.

4.4 SOFTWARE AND HARDWARE

All code for this project is developed using Python 3.9 and executed on a computer cluster with the following specifications:

- Memory: 1024 GB RAM
- Graphics Processing Unit: NVIDIA Ampere A100 80Gb
- CPU: 2 x AMD 7313 3.0Ghz, 16C/32T, 128M

5

ANALYSIS

In this chapter, we provide an evaluation of forecasting models across multiple time frames and present a comparative analysis in Section § 5.1. We then investigate the impact of various data properties, such as autocorrelation, trend, seasonality, heteroskedasticity, and stochasticity, on forecast performance in Section § 5.2. In Section § 5.3, we shift our attention to how market factors, notably volatility and market capitalization, shape predictive accuracy. We further investigate the effect of different time frames on forecasting accuracy in Section § 5.4, and in Section § 5.5, we examine the implications of broadening the data time span, focusing on the augmentation of training data and our models' performance on long-term forecasting. Closing the chapter, § 5.6 acknowledges this thesis' limitations and points to potential directions for future research, emphasizing the need to strengthen data robustness, enhance model performance, and understand the limits of hyperparameter tuning. We conclude with a discussion of our findings in Section § 5.7, where we summarize the answers to our research questions.

5.1 MODEL PERFORMANCE

In this section, we examine the performance metrics of various time-series forecasting models used in our study, aiming to assess and compare their accuracy in predicting future data points, to provide an answer to $RQ1$. Our evaluation covers multiple time frames, from daily to minute-level data observations, allowing us to evaluate the models' versatility and strength in different real-world conditions. We follow the categorization presented in Chapter 2, which provides us with six categories that guide our evaluation of the forecasting models' performance dynamics. Table 5.1 lists these categories and the models they include.

Category	Forecasting Model
Statistical Analysis (Baseline)	ARIMA
RNN-Based	RNN, LSTM, GRU
Deep Learning-Based	TCN, N-BEATS, TFT
Ensemble-Based	XGBoost (XGB), LightGBM, Random Forest
Hybrid	N-HiTS
Time Series Decomposition-Based	TBATS, Prophet

Table 5.1. Overview of forecasting model categories.

We examine the results of the forecasting models by time frame, starting with the daily time frame and then moving to shorter periods. Each subsection includes an analysis of all models as a group, followed by a detailed review of the top five models in each category. We also investigate any unusual performance in specific datasets.

5.1.1 *Analysis of the Daily Time Frame*

To ensure a fair comparison of our forecasting models, we generate a boxplot for each, utilizing the aggregated RMSE results across all cryptocurrencies. These plots draw from forecasts converted into logarithmic returns, providing a consistent basis for comparison across our various datasets.

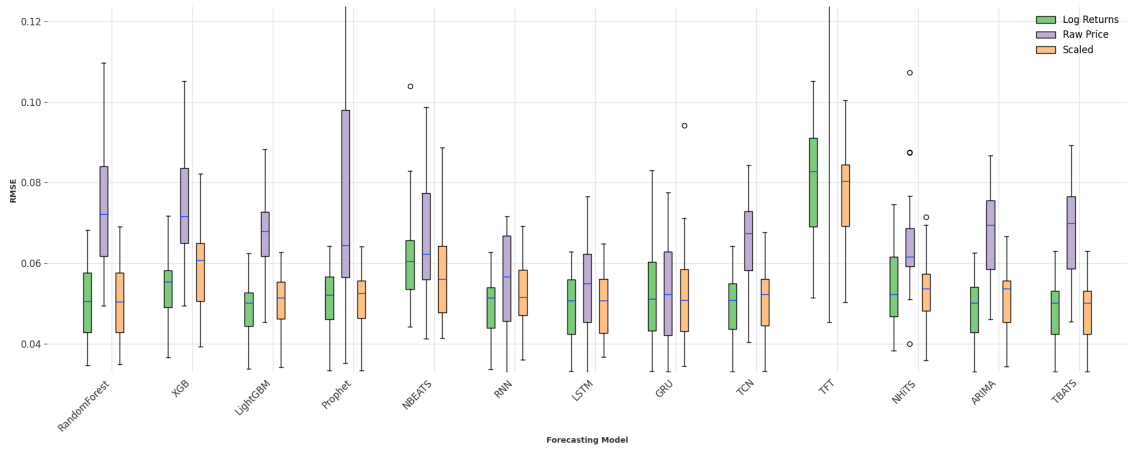

Figure 5.1. Comparative analysis of RMSE values among models using logarithmic returns on the daily time frame.

In Fig. 5.1, notable patterns emerge. We see that the logarithmic returns and scaled boxplots are very similar. However, the scaled dataset seems to be slightly worse compared to the logarithmic returns. This outcome is somewhat expected, as we assumed in Section § 3.3 that the scaled dataset would perform better on lower time frames and not necessarily

have better performance on the higher time frames. The raw price dataset had the worst performance, which is partly due to the process of converting raw price predictions back to logarithmic returns, which can greatly magnify even small errors in raw price predictions, increasing the RMSE values.

Prophet, N-BEATS, and TFT show underwhelming performance. Conversely, there's no clear best performer. Several models, including TBATS, ARIMA, LightGBM, LSTM, and TCN, perform well with scaled and logarithmic returns, indicating that most model categories, except hybrid, have at least one strong contender. The modest results of RNN-based and deep-learning models might be due to the scarce training data for each time frame, a point for further investigation in Section § 5.5.

To investigate the performance per cryptocurrency, we take a closer look at the forecasting performance of the top five models across various categories. We use boxplots once again, but this time each boxplot presents the RMSE of a forecasting model on a cryptocurrency, as shown in Fig. 5.2

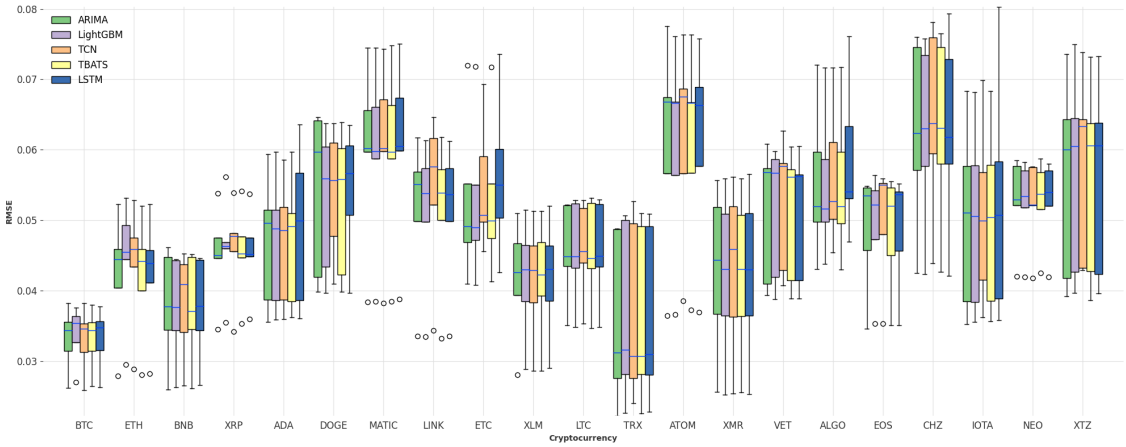

Figure 5.2. Comparative assessment of the RMSE of leading forecasting models using logarithmic returns in a daily time frame.

The forecasting models demonstrate a fairly consistent performance across most cryptocurrencies. BTC, known for its largest market capitalization and lowest volatility, consistently yields the most precise forecasts across all models. However, there are some unexpected results, especially with ETH, the second-largest cryptocurrency. Here, LightGBM and TCN perform significantly worse than other models, as we can see from the difference in their mean values. We have a closer look at what could be the possible reasons for these results in Fig. 5.3. This figure shows the mean and standard deviations in the forecasts of the top five forecasting models. We use this information to compare the forecasts to the actual time series data.

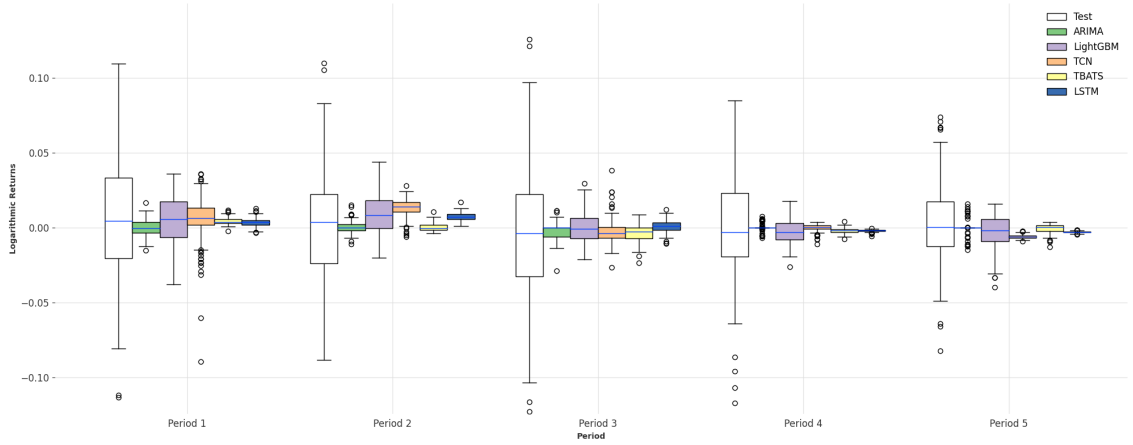

Figure 5.3. Predictive performance of top forecasting models on ETH on the daily time frame.

In Fig. 5.3 we notice that ARIMA's forecasts mostly group around zero. This leads to good performance in RMSE, as the logarithmic returns of the test data also share this mean. However, it is not able to fully capture the range of actual logarithmic returns, as shown by how different its predictions are from the test data. The LightGBM model shows a wider variation, closer to the test values, but its predictions often go in the wrong direction when they are high, resulting in worse RMSE results. The TCN model's weak performance in the second period is due to its average being quite different from the real data. This could be due to the training data being too different for this model to generalize on the test set.

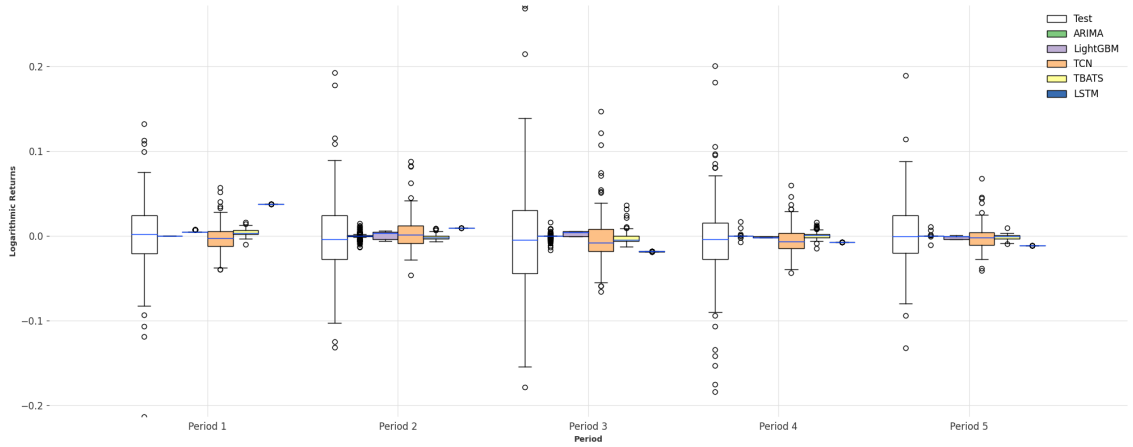

Figure 5.4. Predictive performance of top forecasting models on ETC on the daily time frame.

The results on ETC, shown in Fig. 5.4, were different for the top forecasting models, especially TCN and LSTM, which did not do as well in predictions for this cryptocurrency. The reasons for this weaker performance are similar to what happened with ETH, with the models' average often being quite far from the real average. Both LSTM and TCN con-

sistently showed this difference, leading to their lower performance as measured by RMSE. Additionally, the variation shown by the LSTM model is notably limited, a point discussed earlier in the results on ETH, and this trend continues in other datasets.

This analysis evaluates various forecasting models' accuracy on a daily time frame, using RMSE values for comparison. Results show consistent performance across the top-performing models, with none distinctly outperforming others. However, certain models struggle with specific cryptocurrencies like ETH and ETC, mainly due to variances in mean predictions versus actual values.

5.1.2 *Analysis of the Four-Hour Time Frame*

The observations in the four-hour time frame largely support the trends identified in the daily time frame as we can see in Fig. 5.5. Interestingly, the Prophet model adjusts its performance, aligning more with other top-performing models. A thorough review of the relevant datasets confirms that using logarithmic returns still produces the best predictive results across different forecasting models. This phenomenon is likely because the four-hour time frame is still considered a longer time frame, leading to our scaled dataset not contributing any improvement in performance.

While most models maintain performance levels similar to those seen in the daily time frame, some notable exceptions exist. Specifically, N-BEATS, TCN, TFT, and N-HiTS show a clear increase in RMSE. TCN, despite its stronger results in the daily time frame, experiences a significant downturn in this shorter period.

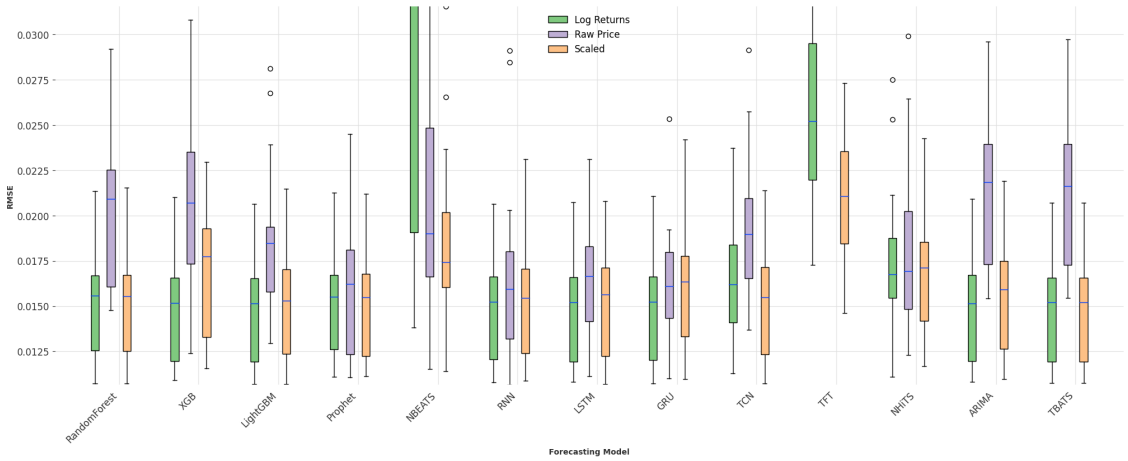

Figure 5.5. Comparative assessment of RMSE metrics across forecasting models using logarithmic returns in a four-hour time frame.

The best-performing models in this time frame include LightGBM, TBATS, ARIMA, RNN, and TCN. However, TCN's weak performance stands out in Fig. 5.6, where it consistently

falls among the least accurate models, especially in its predictions for BTC, ADA, TRX, and EOS.

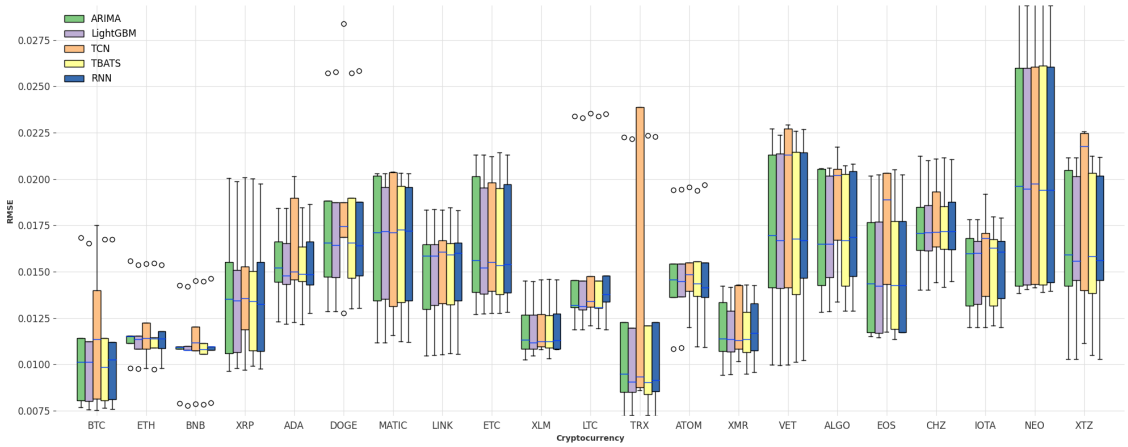

Figure 5.6. Comparative analysis of RMSE metrics among top-performing forecasting models using logarithmic returns in a four-hour time frame.

The performance of TCN in forecasting TRX over the four-hour time frame, as depicted in Fig. 5.7, is suboptimal. In the initial test phase, TCN's predictions significantly stray from the actual values, leading to a high RMSE. This trend continues in the following test phases, marked by considerable differences between the predicted and real data. Further investigation is required to understand the reasons behind this underperformance, potentially stemming from inconsistencies in the volatility data during the training and testing periods. The TCN model also shows poor performance with other cryptocurrencies, especially BTC. This is evident in the first and fourth periods, where the model's predictions significantly diverge, suggesting high volatility at these times. We investigate these findings further in Section § 5.3.1.

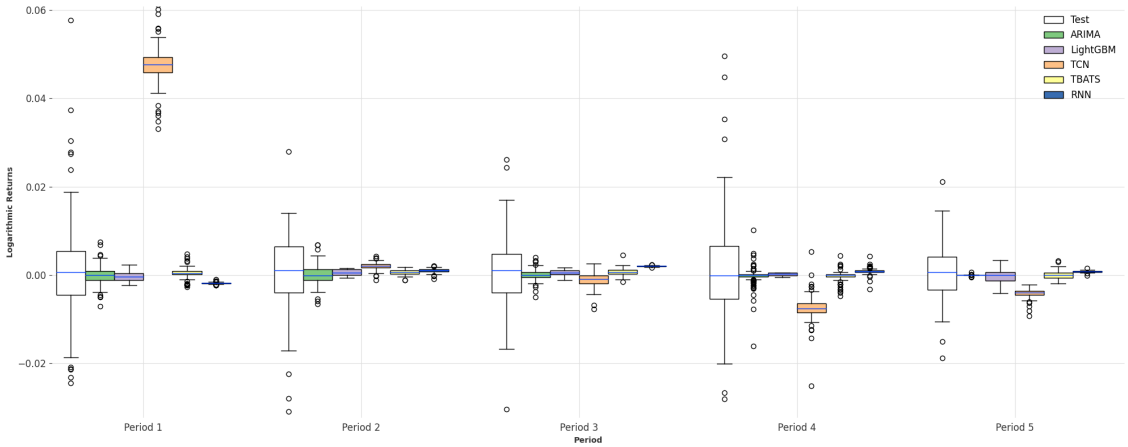

Figure 5.7. Predictive performance of top forecasting models on TRX on the four-hour time frame.

In the four-hour time frame, similar trends to the daily analysis are seen, with the Prophet model getting better, as shown in Fig. 5.5. Using logarithmic returns continues to give the best predictions, while the scaled dataset does not help improve results in this longer time frame. However, some models, like N-BEATS, TCN, TFT, and N-HiTS, show higher error rates, with TCN's results dropping compared to its daily performance. On the other hand, LightGBM, TBATS, ARIMA, and RNN keep doing well. TCN struggles particularly with forecasting certain cryptocurrencies, such as TRX and BTC, indicating that high volatility might be affecting it. These issues point to the need for a deeper look into volatility in Section § 5.3.1.

5.1.3 *Analysis of the Fifteen-Minute Time Frame*

Fig. 5.8 shows a different trend compared to previous time frames. Specifically, fewer forecasting models perform poorly on scaled data relative to logarithmic returns. Especially deep-learning forecasting models perform better on this dataset. Also, traditional and decomposition-based time series models appear largely unaffected by the dataset type.

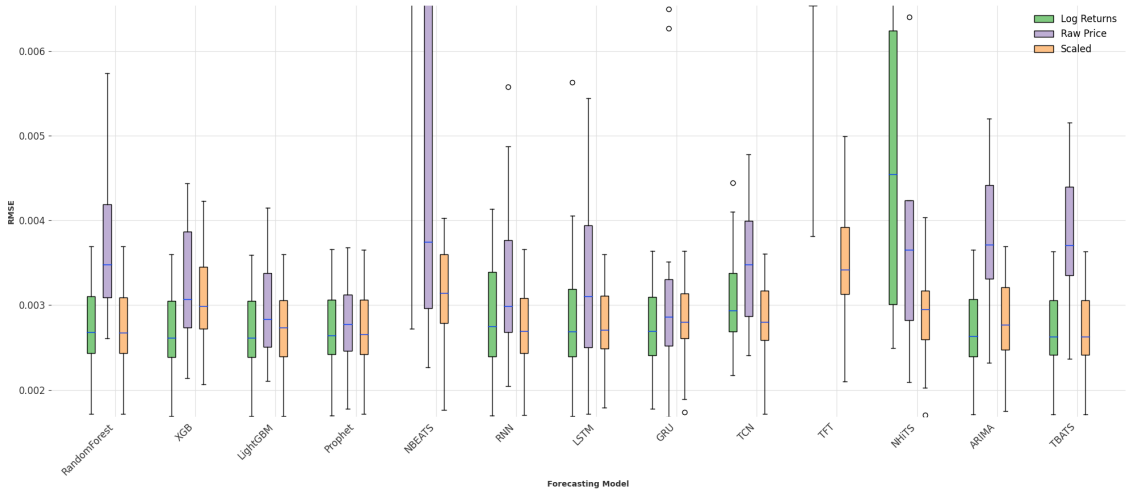

Figure 5.8. Comparative evaluation of RMSE metrics among top forecasting models using logarithmic returns in a fifteen-minute time frame.

Logarithmic returns continue to dominate, with XGBoost leading in accuracy, followed by TBATS, ARIMA, RNN, and TCN. Notably, TCN and RNN show better results with scaled data, therefore we perform the analysis of these models using the scaled dataset.

When comparing performance across cryptocurrencies, models like ARIMA, XGBoost, and TBATS display consistent patterns. In contrast, GRU and TCN exhibit substantial variations, particularly in predictions for MATIC and IOTA. To understand these disparities, we examine the detailed performance metrics for IOTA in Fig. 5.10.

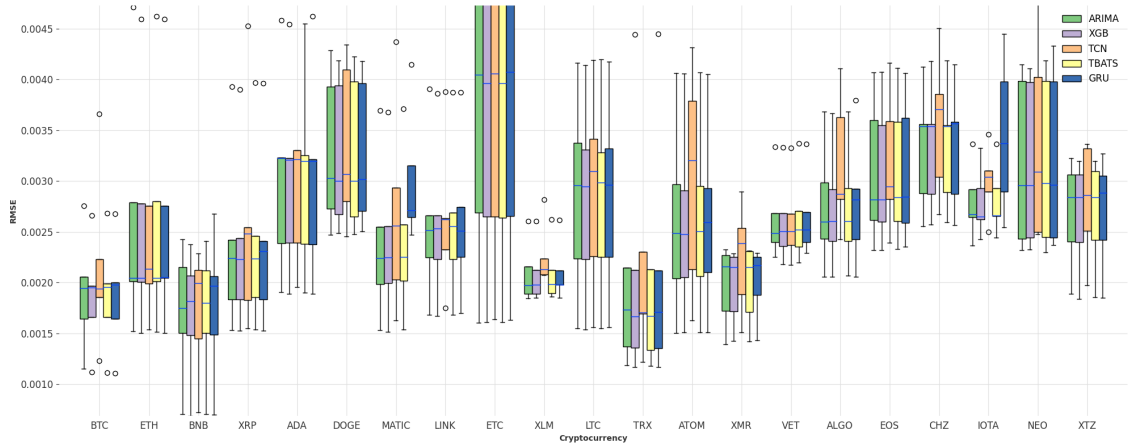

Figure 5.9. Comparative analysis of RMSE values among models using logarithmic returns on a fifteen-minute time frame.

The high RMSE on IOTA for the GRU model stands out in the initial test phase, largely due to differing volatility levels between the training and testing sets. The training set had more volatility, while the test set showed less, hampering the GRU model's ability to make accurate predictions. This pattern is also evident in other cryptocurrencies like BTC and MATIC, with GRU's poorest performance in the first period. Interestingly, the other models also exhibit worse performance during the first period, as we can see by the difference between the actual mean and the mean of each model.

The TCN model started stronger than the GRU but saw decreased accuracy in later periods, especially the third and fourth. Here, we see an increase in predictions far from the actual mean that led to a higher RMSE. This inconsistency aligns with our previous findings on TCN's performance.

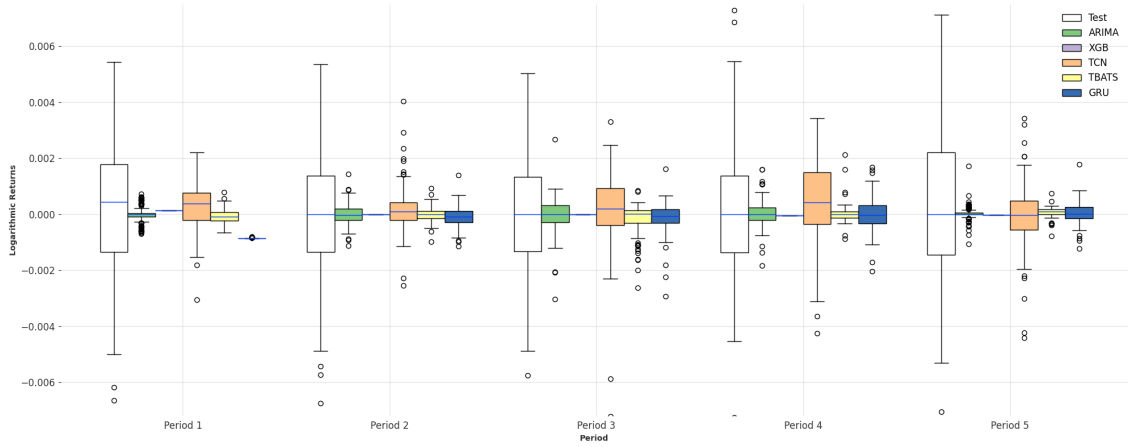

Figure 5.10. Predictive performance of top forecasting models on IOTA on the fifteen-minute time frame.

In the fifteen-minute frame, Fig. 5.8 reveals deep-learning models performing better on scaled data, while traditional models are stable across datasets. XGBoost tops the accuracy chart, with ARIMA, TBATS, RNN, and TCN following; the latter two excel on scaled data. Despite consistent performance from ARIMA, XGBoost, and TBATS, GRU and TCN vary, notably for MATIC and IOTA, as seen in Fig. 5.10. GRU's initial high RMSE on IOTA could be due to training-testing volatility differences. TCN, though starting strong, falters later, similar to our previous results with this model.

5.1.4 *Analysis of the One-Minute Time Frame*

Fig. 5.11 reveals performance metrics in the one-minute time frame, the most detailed in our study. Notably, the gap in performance grows between models trained on logarithmic returns and those on scaled data, especially in RNN-based, deep learning-based, and hybrid forecasting categories. Therefore, we use the scaled data for the top-performing models from those categories.

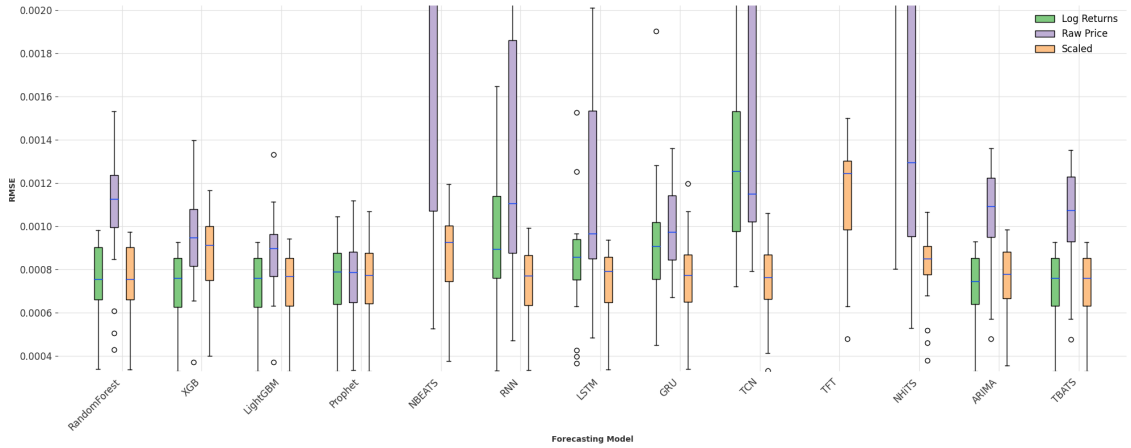

Figure 5.11. Comparative assessment of RMSE metrics across top forecasting models using logarithmic returns in a one-minute time frame.

Fig. 5.11 shows that in the one-minute time frame, the mean RMSE values for LightGBM, XGBoost, and ARIMA are similar, suggesting consistent performance. LSTM leads the RNN-based models, and TCN stands out in the deep learning category. The top five models—LightGBM, ARIMA, TBATS, LSTM, and TCN—mirror earlier findings.

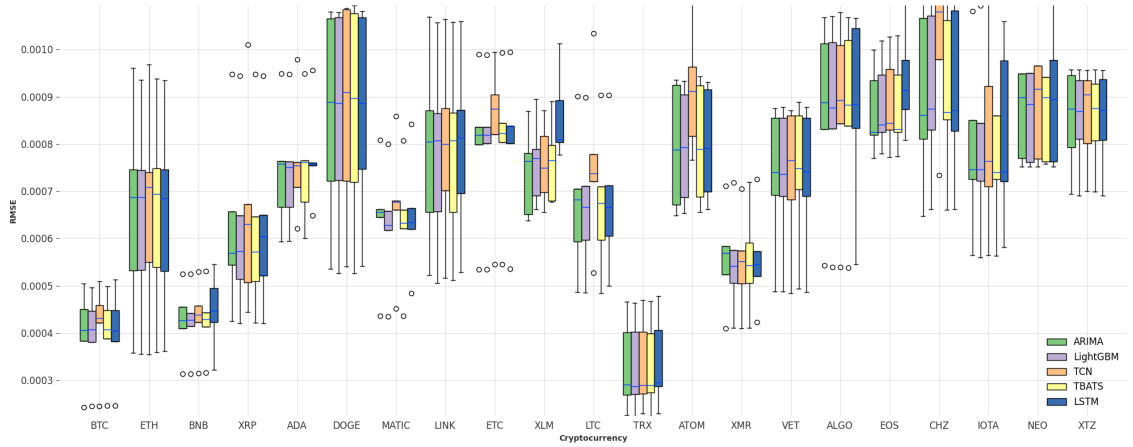

Figure 5.12. Comparative analysis of the RMSE of forecasting models using logarithmic returns in a one-minute time frame.

Despite their standings, LSTM and TCN struggle to predict price changes accurately for certain cryptocurrencies, such as LTC, as shown in Fig. 5.13. The LSTM model tends to produce stagnant, near-zero predictions early on, while the TCN model, though more reactive, often makes errors, generating predictions that deviate significantly from real values. This trend is not unique to LTC, as similar forecasting issues occur with other cryptocurrencies, including BTC and XRP, especially in the fifth period.

These inconsistencies may stem from a volatility mismatch between the training and testing datasets. For instance, the fourth period's training dataset has low volatility, in contrast to its testing counterpart. This disparity challenges both LSTM and TCN, leading to less accurate forecasts and higher RMSE values.

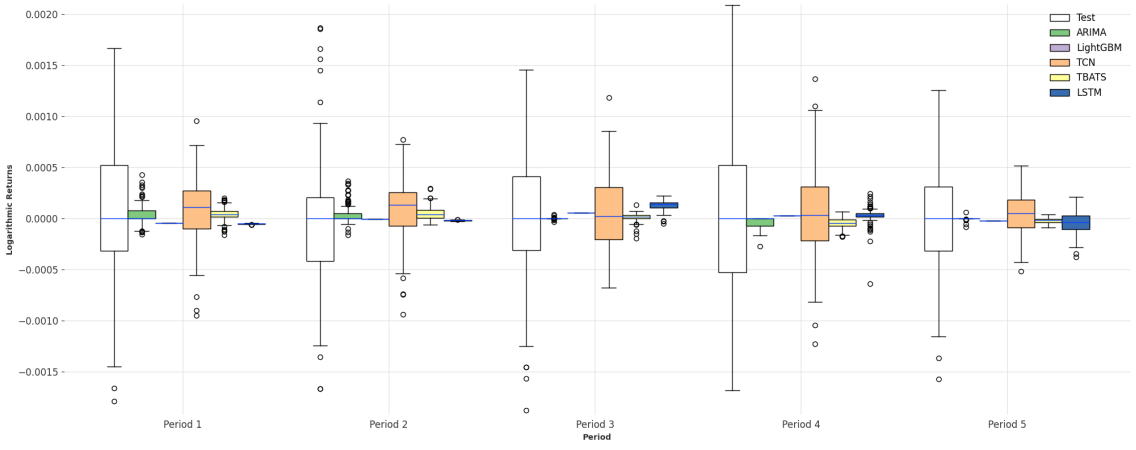

Figure 5.13. Predictive performance of top forecasting models on LTC on the one-minute time frame.

In the one-minute time frame, a significant performance gap exists between models trained on logarithmic returns versus scaled data, particularly in RNN, deep learning, and hybrid models. This discrepancy guides the selection of scaled data for subsequent analyses. LightGBM, XGBoost, and ARIMA display uniform performance, while LSTM and TCN lead their respective categories, maintaining a top-five model consistency seen in prior assessments. Despite their overall efficacy, LSTM and TCN encounter prediction challenges for cryptocurrencies like LTC, detailed in Fig. 5.13. LSTM often delivers stagnant forecasts, whereas TCN, though active, tends to deviate significantly. This predictive instability, also noted in other cryptocurrencies, is potentially caused by volatility differences between the training and testing phases, particularly in the fourth period, leading to increased RMSEs.

5.1.5 *Synthesis of Model Performances and Comparative Analysis*

This section synthesizes the findings of previous subsections, connecting them to existing research and presenting a collective overview of forecasting models' outcomes in Table 5.2. By averaging the RMSE for each cryptocurrency, we compile a comprehensive summary of model performances. The subsequent table ranks models, highlighting in bold those whose RMSEs match or exceed ARIMA's. We also draw attention to the models that showed enhanced performance with the scaled dataset by underlining them. Remarkably, only a handful surpass ARIMA, underscoring its continued relevance as a robust baseline, comparable to more complex models like TBATS and RNN-based ones. Additionally, the scaled dataset did not boost the effectiveness of the majority of the high-performing models. However, it proved beneficial for shorter time frames, specifically for fifteen and one-minute forecasts.

In conclusion, our analysis illuminates the efficacy of various forecasting models, drawing parallels with observations noted in the related work section. Consistent with earlier studies [46], LSTM slightly outperforms RNN in stock price predictions, a trend reflected in our data, especially in daily and one-minute intervals, as shown in Table 5.2. These minor differences confirm LSTM's edge in cryptocurrency forecasting over RNN, although variations exist across different intervals.

TCN stands out among deep learning models in our study, despite inconsistent performance across various datasets and intervals. This inconsistency matches prior research on TCN's application in financial forecasting [58, 59]. Conversely, N-BEATS and TFT face difficulties, more so in shorter intervals, even though they improve with dataset alterations. The literature presents a divided view on N-BEATS [63, 65] and a generally unfavorable opinion on TFT's forecasting abilities [60, 68], aligning with our observations.

Gradient-boosting models, namely XGBoost and LightGBM, exhibit strong performance, supporting their suitability for cryptocurrency forecasting as suggested in existing literature [75, 77, 78]. In contrast, Random Forest shows adequate, yet inferior performance compared to its gradient-boosting peers, a finding in agreement with previous research [70].

Model	Daily	Four-Hour	Fifteen-Minute	One-Minute
TBATS	**0.048765**	**0.014883**	**0.002659**	0.000716
ARIMA	0.048853	0.014922	0.002662	0.000713
LightGBM	0.048965	**0.014841**	**0.002643**	**0.000713**
LSTM	0.049481	0.014968	0.002728	0.000729
RNN	0.049587	0.014948	0.002727	0.000734
TCN	0.049644	0.015312	0.002819	0.000741
Random Forest	0.050304	0.015334	0.002703	0.000745
Prophet	0.051061	0.015276	0.002678	0.000740
GRU	0.052105	0.015007	0.002728	0.000754
N-HiTS	0.052677	0.016749	0.002880	0.000815
XGBoost	0.055058	0.014977	**0.002640**	**0.000713**
N-BEATS	0.057311	0.018082	0.046524	0.000855
TFT	0.079329	0.020879	0.003495	0.001111

Table 5.2. Aggregated model performance on the logarithmic returns, with models outperforming or matching ARIMA in RMSE highlighted in **bold** and those improved by using the scaled dataset underlined.

N-HiTS, an extension of N-BEATS, shows promise despite the absence of a cryptocurrency-specific benchmark. Its enhanced performance over N-BEATS suggests a promising avenue for future model refinement in cryptocurrency forecasting, despite its overall lower ranking.

TBATS excels in our analysis, confirming its potential noted in other studies to surpass ARIMA in cryptocurrency forecasting [81], though it requires significant computational resources. Prophet, on the other hand, yields consistent but modest outcomes, resonating with some research in stock forecasting [83, 84] but not fully aligning with specific studies on Bitcoin [85].

In addressing $RQ1$ concerning the comparison between state-of-the-art machine learning models and traditional ones in forecasting cryptocurrency prices, our analysis reveals a complex picture. State-of-the-art models, including LSTM, LightGBM, and TBATS, demonstrate strong and reliable performances. However, they do not consistently surpass the traditional model, ARIMA, which often matches or even exceeds the forecasting accuracy of these more advanced models.

This analysis highlights that despite the complexity of advanced machine learning models, their performance is not guaranteed to be superior to that of traditional models within the unpredictable realm of cryptocurrency prices. The success of a model is significantly determined by its ability to adapt to market fluctuations and the specific traits of the data, reinforcing the importance of comprehensive empirical testing when choosing a model.

The assessment of N-HiTS shows us new ways hybrid models could develop. The mixed results from models like N-BEATS and TFT highlight how important real-world testing is when choosing a model. This information not only strengthens our basic understanding of using these models in the cryptocurrency market but also shows us where we might focus our future studies. Specifically, we see a need to improve hybrid and deep-learning models for better financial forecasting.

5.2 IMPACT OF DATA PROPERTIES ON FORECASTING PERFORMANCE

This section examines how various data properties, detailed in Section § 3.2, affect the performance of distinct forecasting models. We have previously established that models perform better with logarithmic returns compared to raw price data. This approach effectively tackles non-stationarity, removing the necessity for an independent discussion on stationarity here.

Since the RMSE dataset does not meet the criteria for normality and homogeneity of variances, we choose non-parametric statistical methods, specifically the Mann-Whitney U test, for evaluating model performance. These methods rely on data ranks rather than raw values, providing a robust alternative for our analysis and ensuring a more reliable statistical inference.

5.2.1 *Impact of Autocorrelation on Forecasting Performance*

Here we discuss the analysis from Section § 3.2.2, where the Breusch-Godfrey and Ljung-Box tests identified autocorrelation in 36 and 37 cryptocurrencies, respectively. For thoroughness, we consider datasets autocorrelated if either test indicates so, leading to 41 such datasets. Others are labeled non-autocorrelated.

Building on research that emphasizes correcting autocorrelation to improve forecasting [95, 96], we hypothesized that non-autocorrelated datasets would perform better. We also expected differences in volatility between autocorrelated and non-autocorrelated groups, reflecting findings from the European stock market [97].

Our initial analysis investigated volatility differences between groups, using average volatility values in the Mann-Whitney U test. However, the results show no significance, thus indicating no meaningful difference between autocorrelated and non-autocorrelated groups regarding volatility.

We further examined autocorrelation's effect on forecasting accuracy, using the Mann-Whitney U test for each model, with the hypothesis that non-autocorrelated datasets would have a lower RMSE. Yet, p-values consistently above 0.05 collectively support the conclusion that autocorrelation does not significantly impact predictive accuracy, contrary to initial assumptions. This outcome could be due to forecasting models effectively handling autocorrelation, such as the Auto ARIMA model identifying appropriate parameters for

all datasets. Alternatively, the classification method might be imprecise, as different tests, like the Durbin-Watson, yielded varying results.

Examining autocorrelation's influence, particularly on smaller, less liquid cryptocurrencies, shows a significant impact [98]. However, our study mainly includes larger, more liquid cryptocurrencies. The mean of our autocorrelation test p-values is 0.34, aligning with prior research indicating a 0.35 p-value for highly liquid groups [159]. Future research might consider exploring the influence of autocorrelation on the RMSE in the context of forecasting more illiquid coins than those addressed in this thesis.

5.2.2 *Impact of Trend on Forecasting Performance*

This section delves into the exploration of trend components within our dataset to understand their effect on forecasting performance. We apply four statistical tests to identify trends, combining their outcomes via majority voting. This process classifies 72 datasets as 'no trend,' 7 as 'increasing trend,' and 5 as 'decreasing trend.' To address the category imbalance, 'increasing' and 'decreasing' trends are merged into a single 'trending' category.

While current research has not definitively linked trending prices to volatility, a connection is often assumed [105]. We expect a difference in volatility between the trending and non-trending groups, using the Mann-Whitney U test for analysis. Despite the daily time frame's p-value nearing 0.05, it slightly exceeds it, indicating that trend presence does not significantly affect volatility in our datasets.

Model	Fifteen-Minute	Four-Hour	Daily
Random Forest	0.524	0.077	0.012
XGBoost	0.476	0.077	0.012
LightGBM	0.476	0.103	0.016
Prophet	0.429	0.089	0.020
N-BEATS	0.333	0.455	0.177
RNN	0.429	0.089	0.020
LSTM	0.952	0.117	0.020
GRU	0.429	0.047	0.007
TCN	0.476	0.027	0.016
TFT	0.905	0.455	0.104
N-HiTS	0.095	0.022	0.007
ARIMA	0.429	0.089	0.010
TBATS	0.476	0.103	0.010

Table 5.3. Mann-Whitney U test p-values under the hypothesis that the trending category has lower RMSE, excluding the one-minute time frame due to significant class imbalance.

We also assess trend presence's impact on RMSE using the Mann-Whitney U test, hypothesizing that datasets without trends would have lower RMSEs. However, we mostly find insignificant results, with some p-values nearing 0.99. Altering our hypothesis produces different outcomes, especially on the daily time frame, as shown in Table 5.3. Except for N-BEATS and TFT, all models showed p-values below 0.05, indicating that, unexpectedly, the 'trending' category performed better than the 'no trend' group, a pattern also seen in several models on the four-hour time frame. Concluding that the groups classified a having a trend were easier to forecast, especially in higher time frames. These results starkly contrast with existing literature that underscores the reduced efficacy of machine-learning forecasts on trending data [106].

This discrepancy could be because several models, including TBATS, Auto ARIMA, and Prophet, can detect trends and adjust predictions, enhancing their performance and explaining the significant p-values. Yet, the observed significance could stem from class imbalance. A closer look at cryptocurrencies showing a daily trend—BTC, ETH, BNB, XRP, and IOTA—reveals that the first four had lower-than-average RMSEs, potentially influencing our test results. Further analysis is required to definitively ascertain the relationship between trends and predictability.

5.2.3 *Impact of Seasonality on Forecasting Performance*

This section assesses the effect of seasonality on volatility and our models' forecasting performance through two regression tests. The first test investigates a possible linear relationship between seasonality and volatility, inspired by previous findings in the Arabian stock market [112]. Similar effects might occur in the cryptocurrency market during particular times, like New Year's Eve or weekends. However, due to the cryptocurrency market's global nature, these effects might be less noticeable than in regional stock markets. Moreover, past studies suggest a lack of seasonality in cryptocurrencies [5, 114], a pattern we expect to see in our results.

The first test uses Ordinary Least Squares (OLS) regression with seasonality strength values from Section § 3.2.4, we find no significant linear relationship between seasonality and volatility across all time frames. The second test examines the potential linear relationship between seasonality strength and RMSE. Existing literature suggests that machine-learning forecasts often perform poorly on seasonal data [106]. If this is accurate, we would expect a clear linear relationship, especially on the daily time frame. However, our OLS regression shows mostly insignificant results, except for N-HiTS on the fifteen-minute frame and N-BEATS on the four-hour frame. The significance in these cases might stem from the weaker performance of these models, possibly skewing the results. Notably, N-HiTS is an extension of N-BEATS, which could explain the similar findings [79]. Thus, our models' performance shows no significant link with the seasonality strength of the time series.

These outcomes concur with previous research arguing that seasonality is generally absent in cryptocurrency markets. Even though some seasonal strength was detected in section § 3.2.4, especially on the daily time frame, it was not strong enough to significantly affect volatility or forecasting performance, as indicated by the mostly insignificant findings. We conclude that seasonality has no significant impact on the forecasting performance of the models and data used in this thesis.

5.2.4 *Impact of Heteroskedasticity on Forecasting Performance*

This section examines how conditional and unconditional heteroskedasticity affect the forecasting accuracy of different predictive models. For conditional heteroskedasticity, its limited presence in our dataset narrows the analytical range. Specifically, only nine of the 84 datasets showed conditional homoskedasticity, limiting the strength of any conclusions about its effect on volatility and RMSE. Despite this limitation, we applied the Mann-Whitney U Test to this small subset. The initial analysis aimed to identify any volatility differences between groups, but the results were insignificant across all time frames. This suggests that the conditional heteroskedasticity presence did not lead to varying volatilities.

Further tests, based on the assumption that homoskedastic datasets would perform better, mostly showed no significant results, except for N-BEATS on the fifteen-minute frame.

This suggests that conditional heteroskedasticity had little effect on our forecasting models' performance, contrary to past studies suggesting models like ARIMA might struggle with heteroskedastic time series [32]. The discrepancy in our findings could be due to the class imbalance, as only ten percent of the datasets were conditionally homoskedastic.

In the context of unconditional heteroskedasticity, the dataset exhibited a more balanced class distribution, providing a foundation for a comprehensive analysis. We performed two distinct tests with the Mann-Whitney U Test to assess the statistical significance of unconditional heteroskedasticity, with the first focused on its influence on volatility. A significant result was identified in the fifteen-minute time frame when applying the Goldfeld-Quandt test, indicating that the cryptocurrencies deemed heteroskedastic by this test display markedly higher volatility. Regrettably, substantial imbalances in the one-minute and daily time frames precluded the possibility of conducting tests in these intervals.

Model	Fifteen-Minute	Four-Hour	Daily
Random Forest	0.018	0.169	0.800
XGB	0.018	0.322	0.767
LightGBM	0.018	0.297	0.857
Prophet	0.015	0.273	0.881
N-BEATS	0.652	0.542	0.386
RNN	0.151	0.297	0.857
LSTM	0.007	0.250	0.881
GRU	0.009	0.207	0.767
TCN	0.107	0.727	0.829
TFT	0.322	0.187	0.905
N-HiTS	0.703	0.322	0.829
ARIMA	0.025	0.297	0.857
TBATS	0.018	0.297	0.829

Table 5.4. Mann-Whitney U test p-values assessing the impact of Breusch-Pagan test results on the RMSE.

We also investigated unconditional heteroskedasticity's impact on our models' forecasting performance, hypothesizing that the heteroskedastic datasets would have worse results. Here, the fifteen-minute frame was significant again, but this time for the Breusch-Pagan test. The details in Table 5.4 show several significant findings, except for models like N-BEATS, RNN, TCN, TFT, and N-HiTS. Unfortunately, other time frames and the Goldfeld-Quandt test showed no significance, suggesting that heteroskedasticity's influence is minimal on our datasets concerning volatility and forecasting performance. This was somewhat expected, given that most research points to difficulties for models like ARIMA when dealing with heteroskedastic time series.

5.2.5 *Impact of Stochasticity on Forecasting Performance*

In Section § 3.2.6, we determine that most of our datasets have a Hurst index exceeding 0.55, indicating a positive correlation. However, a portion falls within the 0.45 to 0.55 range, typically linked with Brownian motion. Research suggests that a Hurst index away from 0.5 signals increased predictability [124]. Therefore, we hypothesize that cryptocurrencies deviating from Brownian motion should, in theory, be more predictable.

Our first analysis investigates a potentially significant relationship between volatility and the Hurst index categorization. Using the Mann-Whitney U test, we find the results across all time frames insignificant, leading us to conclude there is no apparent correlation between the Hurst index categorization and volatility.

We then analyze the effect of stochasticity on predictive models with the Mann-Whitney U test. Segmenting the test by time frame, we only find a significant p-value within the fifteen-minute frame for the TFT model. This finding aligns with our previous observations in Section § 3.2.6, where the greatest presence of Brownian motion occurred within the fifteen-minute time frame. However, its restriction to one model and one time frame leads us to conclude that stochasticity does not significantly affect RMSE across the models studied. The limited impact of stochasticity on model performance might be related to the Hurst index results, as there are no H-values below 0.45, indicating a class imbalance in our datasets.

We also use OLS regression to determine the linear relationship between the H-value and volatility. The results are consistently insignificant, suggesting no correlation between the Hurst index and volatility. Additionally, we use OLS regression to examine a potential inverse relationship between the H-value and the RMSE, as prior research implies [124]. Nonetheless, we only encounter two significant results: for the TFT model on the fifteen-minute time frame and for N-BEATS on the four-hour time frame. This inconsistency might result from the different types of financial data used, as the existing research was based on stock market data.

5.2.6 *Summary of Findings on Data Properties*

In addressing research question *RQ2a* regarding the impact of data properties on predictive performance. We conclude that autocorrelation, seasonality, conditional heteroskedasticity, and stochasticity do not affect volatility and forecasting performance. Conversely, we identify trend and unconditional heteroskedasticity as significantly impacting forecasting performance. As detailed in Section § 5.2.2, datasets characterized as 'trending' demonstrated superior performance over their non-trending counterparts, notably in the four-hour and daily time frames. This indicates that trends exert a more pronounced impact on longer time frames. Furthermore, our analysis in Section § 5.2.4 reveals that homoskedastic datasets outperform heteroskedastic ones in the fifteen-minute time frame, as evidenced by the Mann-Whitney U test results applied to the Breusch-Pagan test outcomes. This ana-

lysis shows the nuanced role of specific data properties in forecasting, highlighting the importance of trend and unconditional heteroskedasticity in influencing predictive accuracy across various time frames.

5.3 EFFECT OF MARKET FACTORS ON PREDICTIVE ACCURACY

In this section, we address *RQ2b*, which examines the influence of market factors, specifically volatility and capitalization, on prediction accuracy. We first analyze volatility's impact on forecasts in Section § 5.3.1, followed by an exploration of how market capitalization correlates with predictive precision in Section § 5.3.2.

5.3.1 *Effect of Volatility on Predictive Accuracy*

To classify volatility in categories, we use the volatility categories outlined in Section § 3.2.7 and assign each segment of our test and training data to a specific category. Through these classifications, we assess the correlation between train and test volatility on the RMSE. We anticipate improved performance within the same testing volatility class when training and testing volatility are similarly classified, assuming the model captures comparable patterns under these conditions. To explore this, we generate heatmaps (Fig. 5.14) displaying the performance of all forecasting models, with the exception of outliers such as TFT, N-BEATS, and N-HiTS. Regrettably, our data does not encompass all possible combinations of training and testing volatility. For example, we observed no instances of high training volatility paired with low testing volatility, as this abrupt transition is uncommon.

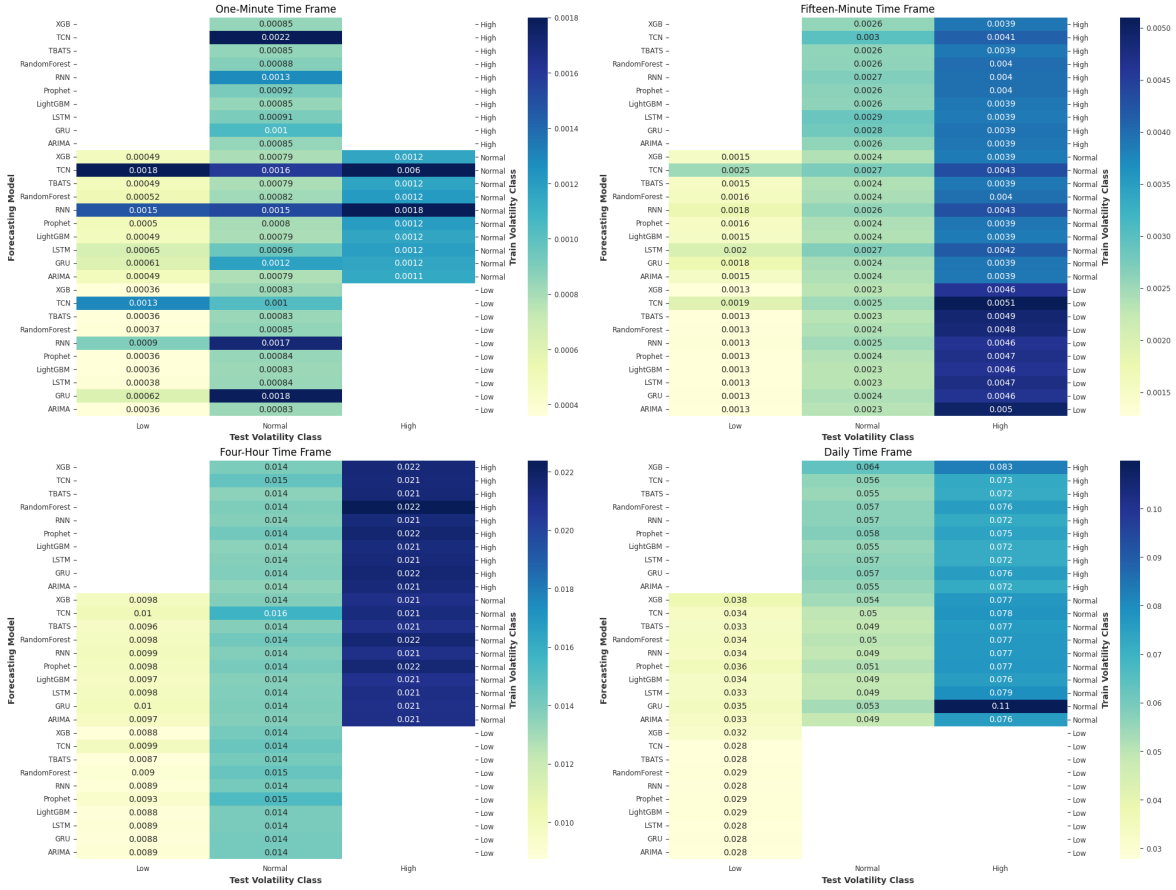

Figure 5.14. Impact of train and test volatility on the RMSE of the logarithmic returns dataset.

In Fig. 5.14, our analysis somewhat confirms our expectations for each time frame heatmap. For example, each heatmap shows that the best performance consistently occurs when both testing and training volatility are low. This trend also holds for normal volatility but is less pronounced. Notably, in the fifteen-minute and four-hour time frames, the difference in performance between low training and normal testing volatility is nearly indistinguishable from that of normal training and normal testing volatility. As anticipated, higher testing volatility generally yields poorer results, particularly evident in the fifteen-minute and daily time frames. Specifically, training on low volatility and testing on high volatility results in less accurate forecasts than training and testing within the same volatility category.

The heatmaps also lead us to conclude that across all time frames, testing on low volatility consistently produces the best performance. This trend is logical since low volatility allows models to make more reliable predictions close to the mean, resulting in a lower RMSE. Conversely, higher testing volatility corresponds to a higher RMSE.

To confirm our hypothesis, we applied the Mann-Whitney U test across different time frames, evaluating performance within each test volatility category. We partitioned the data into two sets: one with matching training and testing volatility categories, and another

with differing categories. Our hypothesis suggested that identical volatility in training and testing would yield a lower RMSE. The findings, however, were inconsistent across time frames.

In the one-minute time frame, almost all models, excluding N-BEATS, GRU, TCN, TFT, and N-HiTS, exhibited significant p-values. Nonetheless, the high testing volatility class couldn't be evaluated due to the presence of only one group.

In contrast, the fifteen-minute time frame showed significance solely within the low testing volatility class. This result might stem from using three distinct training classes, while the Mann-Whitney U test typically involves just two. When we tested for significance in the normal test volatility class using only normal and high-volatility training, we noted some significance for XGBoost, LightGBM, LSTM, and GRU. However, the high test volatility class showed no significant effect. Further testing, this time excluding normal training volatility, resulted in lower p-values for all models, but none were significant, potentially due to the rarity of high volatility testing with low volatility training in our datasets.

For the four-hour time frame, most results were insignificant, except for the normal test volatility class and TFT. The heatmap for this time frame supports these results, as performance values for each test volatility class are almost uniform.

The daily time frame, however, showed extensive significance. All models, except N-HiTS, were significant in the low test volatility class. In the normal test volatility class, nearly all p-values were close to zero, with N-BEATS slightly above 0.05, marking it as insignificant. The high test volatility class showed no significant results, a surprising finding considering the distinct differences in model performances within this category.

To further explore how training and testing volatility impact RMSE, we use OLS regression, the results of which are presented in Table 5.5. High R^2 values exceeding 0.99 indicate a strong correlation between the model and the data, a conclusion supported by the F-statistic. Notably, training volatility typically does not have a substantial effect on RMSE, whereas testing volatility maintains a consistent and significant relationship, evidenced by p-values well below the 0.05 threshold. This trend is apparent for all forecasting models, except for N-BEATS, N-HiTS, and TFT. We attribute this deviation to these specific models' inability to adequately fit the data and generate precise predictions, as previously discussed in Section § 5.1.

Our literature review reveals a lack of research specifically examining the effect of volatility on the predictive accuracy of forecasting models in our field of study. This gap in the literature hinders our ability to compare our results directly with those in existing studies. However, our findings echo the conclusions of a study on the role of volatility in demand forecasting [19]. That research emphasized that volatility significantly affects the precision of time series forecasting models because changes in volatility can alter data patterns, thereby impacting forecast accuracy. This is consistent with our observations that models tend to perform more predictably in environments characterized by low testing volatility.

Model	Intercept	Train Coef	Test Coef	R^2	P|t| Train	P|t| Test	F-statistic
ARIMA	0.000053	-0.006020	0.195703	0.995	0.230	1.04×10^{-39}	10769.477
GRU	-0.000263	-0.002439	0.206703	0.952	0.381	7.21×10^{-7}	2111.702
LSTM	0.000175	-0.008922	0.200931	0.993	0.120	2.75×10^{-35}	6446.670
LightGBM	0.000073	-0.006655	0.196835	0.995	0.421	5.05×10^{-37}	10852.985
N-BEATS	0.036994	-0.132873	0.270784	0.074	0.578	0.392	3.428
N-HiTS	0.004862	0.018729	0.161668	0.778	0.556	0.045	387.864
Prophet	0.000043	0.002839	0.192866	0.992	0.578	6.55×10^{-26}	5301.568
RNN	0.000436	-0.001317	0.190240	0.989	0.500	8.95×10^{-19}	5667.581
Random Forest	-0.000032	-0.013363	0.212446	0.992	0.301	9.00×10^{-34}	7564.038
TBATS	0.000067	-0.008055	0.198044	0.995	0.274	6.17×10^{-37}	12793.444
TCN	0.001069	-0.004033	0.193046	0.977	0.411	8.56×10^{-12}	3660.332
TFT	0.008468	0.150346	0.099149	0.726	0.123	0.436	126.987
XGBoost	-0.000375	0.018251	0.189358	0.994	0.071	4.21×10^{-29}	7830.359

Table 5.5. OLS regression results on train and test volatility.

In responding to a segment of *RQ2b*, we provide an in-depth examination of how volatility influences the performance of cryptocurrency forecasting models. We conclude that volatility, particularly testing volatility, plays a substantial role in determining the accuracy of predictions, with models generally performing better in situations of low volatility. Although training volatility does affect model performance—especially when testing volatility is low—this influence becomes more complex and challenging to manage in high-volatility conditions. Despite the scarcity of similar studies for comparison, our research sheds light on the intricate relationship between training and testing volatilities and paves the way for future investigations. By uncovering these dynamics, we hope to inspire further research and contribute to a more nuanced comprehension of volatile financial markets.

5.3.2 *Effect of Market Capitalization on Predictive Accuracy*

Market capitalization is a crucial characteristic inherent to each cryptocurrency. However, existing literature shows a noticeable gap, primarily concentrating on well-known cryptocurrencies with significant capitalization. We examine the relationship between market capitalization, volatility, and the accuracy of forecasting models within the cryptocurrency sector.

A common belief in financial analytics suggests that assets with larger market capitalizations tend to have more stable and predictable price trends, indicating reduced volatility [10, 11]. However, current research does not strongly substantiate the connection between volatility and market capitalization. For instance, a study on the Nairobian stock exchange documented only a marginal relationship between these elements [23]. In this section, we seek to determine whether a similar scenario exists in the cryptocurrency market.

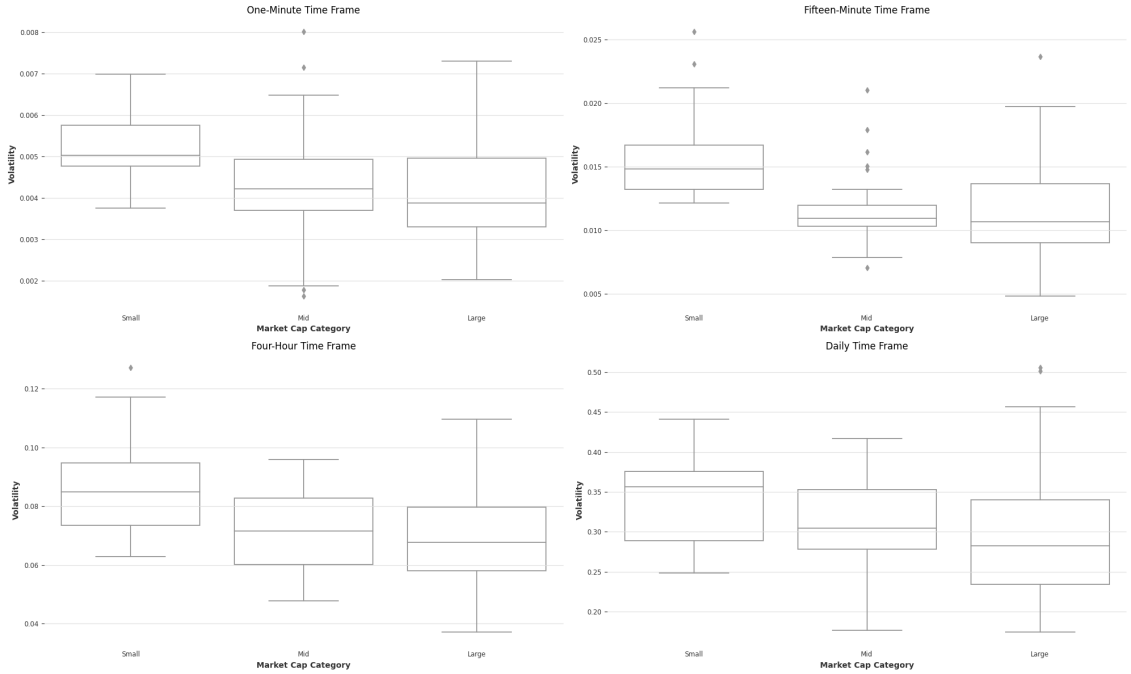

Figure 5.15. Boxplot of volatility values for each market capitalization category.

First, we examine the potential correlation between market capitalization and volatility. Grouping our volatility values by market capitalization category reveals a certain relationship between these two variables, as depicted in the boxplots in Fig. 5.15. We hypothesize that this trend may be due to the larger market capitalization contributing to more stable price movements for an asset. Our next step is to ascertain the statistical significance of this relationship within our datasets.

To determine if there is any significant distinction in volatility among different market capitalization categories, we employ the Kruskal-Wallis test. Notably, the results vary significantly when categorizing by time frame. Without considering time frames, the p-value stands at 0.184, indicating no significance. However, when sorted by time frame, a significant p-value emerges for the fifteen-minute interval, as detailed in Table 5.6. This finding implies a substantial disparity in volatility across the three market capitalization categories specifically in the fifteen-minute and four-hour time frames. Conversely, other time frames yield p-values above 0.05, suggesting no significant volatility difference among the market capitalization category groups.

Time Frame	p-value
1m	0.13670
15m	0.01316
4h	0.02989
1d	0.33959

Table 5.6. Kruskal-Wallis test results for market capitalization category and volatility.

To further investigate the notable p-values, we apply the Mann-Whitney U test, contrasting each market capitalization category. We hypothesize that categories with larger market capitalizations will demonstrate lower volatility. Our findings support this hypothesis, especially when comparing mid and small, as well as large and small market capitalization categories within both fifteen-minute and four-hour time frames, as detailed in Table 5.7. Interestingly, a significant value also emerges when comparing large and small categories in the one-minute time frame. However, we find no significant difference in volatility between large and mid-sized categories, suggesting that category impact is noticeable only when contrasting the extremes, specifically within the fifteen-minute and four-hour time frames.

Time Frame	Mid vs. Small	Large vs. Small	Large vs. Mid
One-Minute	0.111	0.048	0.210
Fifteen-Minute	0.008	0.004	0.500
Four-Hour	0.028	0.008	0.274
Daily	0.155	0.111	0.579

Table 5.7. Mann-Whitney U test results for market capitalization category comparison on volatility.

To assess the importance of the actual market capitalization value, we use OLS regression. We construct regression models with data from the final period, treating market capitalization as the independent variable and volatility as the dependent variable. We adopt this strategy because market capitalization data is available only for the final period. Despite minimal changes in market capitalization over time, using a consistent value for all periods might undermine the OLS regression results reliability.

When the results are aggregated, not separated by time frame, the OLS regression results indicate insignificance, reflected by a p-value of 0.531. Yet, when results are considered by individual time frames, a significant contrast emerges. The OLS regression, conducted for each distinct time frame, consistently reveals significant p-values, as outlined in Table 5.8. These findings suggest that the specific value of market capitalization holds more predictive power over volatility than merely the category of market capitalization. Across all time frames, a negative coefficient is present, signifying an inverse relationship between market capitalization and volatility: as market capitalization rises, volatility tends to decrease. These results are consistent with previous studies indicating a similar trend [10, 11]. However, the uniform significance across all time frames deviates from findings in the Nairobi stock exchange research [23], potentially due to the fundamentally different market dynamics between the cryptocurrency market and the Nairobi stock exchange.

In the following analysis, we examine if the findings from prior statistical tests reflect in the RMSE of our models. We first focus on the relationship between RMSE and market capitalization category, as shown in Fig. 5.16. Consistent with our expectations, the results

Time Frame	Intercept	Coef	R^2	P-value	F-statistic
1m	0.003972	-3.41×10^{-15}	0.26103	0.01794	6.711446
15m	0.015091	-7.57×10^{-15}	0.138826	0.09623	3.062917
4h	0.083589	-5.32×10^{-14}	0.215386	0.034064	5.215716
1d	0.257607	-1.48×10^{-13}	0.23849	0.024693	5.950434

Table 5.8. OLS regression results for market capitalization and volatility.

reveal a trend: larger market capitalization categories tend to have lower RMSE values, indicating improved predictive performance.

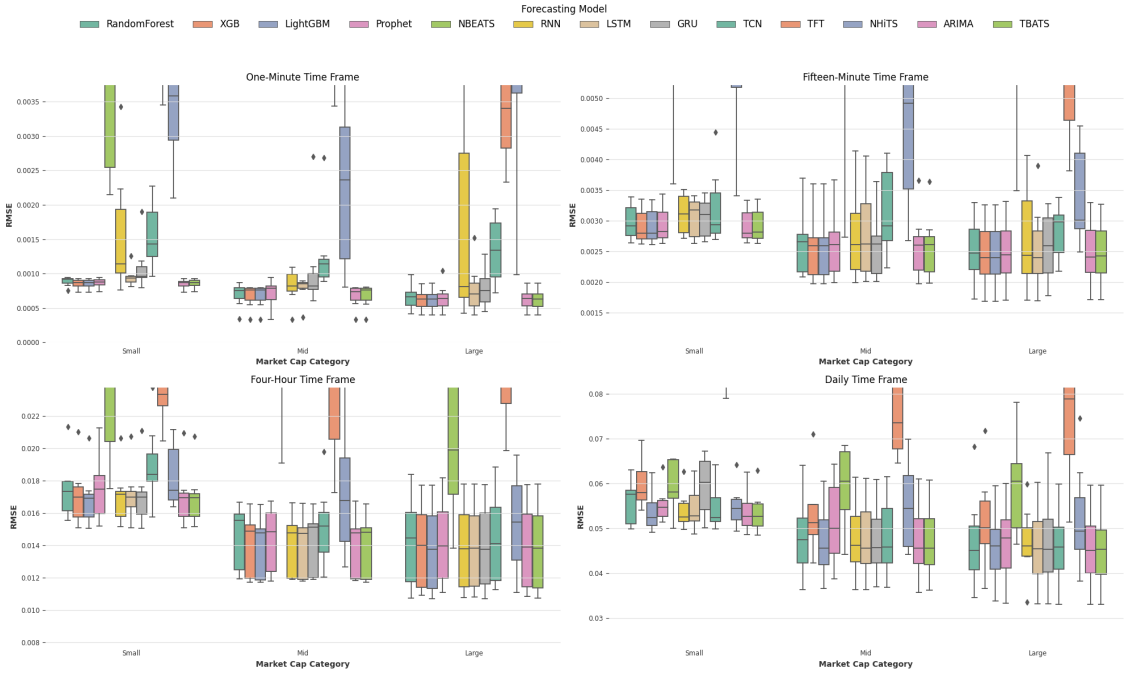

Figure 5.16. Boxplot of RMSE values by market capitalization category.

To assess the effect of the market capitalization category on RMSE, we begin our analysis with a Kruskal-Wallis test, considering the three distinct market capitalization categories. We group the results by time frame, as shown in Table 5.9 and notice several significant results across different time frames, especially in the one-minute and four-hour segments. Notably, the TFT model seems unaffected by market capitalization in all time frames, posing an exception to the general trend. This analysis both confirms our understanding and reveals additional complexity in how market capitalization subtly impacts the performance of forecasting models over different time periods.

We also examined the potential connection between market capitalization value and the RMSE using OLS regression. The count of significant p-values per time frame aligns with our past observations; notably, the one-minute time frame shows a comparable number of significant forecasting models. In contrast, the fifteen-minute time frame reveals the fewest

Model	One-Minute	Fifteen-Minute	Four-Hour	Daily
Random Forest	**0.0247**	0.1220	**0.0388**	0.1502
XGBoost	**0.0118**	0.1063	**0.0401**	0.0955
LightGBM	**0.0118**	0.1063	**0.0395**	0.1285
Prophet	0.0736	0.1220	**0.0479**	0.1641
N-BEATS	**0.0256**	0.9013	0.2864	0.8717
RNN	0.1897	0.3237	**0.0399**	0.1112
LSTM	0.0790	0.1426	**0.0328**	0.1629
GRU	0.1755	0.1128	**0.0482**	**0.0399**
TCN	0.4089	0.6196	**0.0237**	0.1629
TFT	0.0640	0.1524	0.6600	0.5282
N-HiTS	**0.0458**	**0.0345**	0.2267	0.5341
ARIMA	**0.0124**	0.1112	**0.0479**	0.1436
TBATS	**0.0079**	0.1063	**0.0395**	0.1175

Table 5.9. P-values from the Kruskal-Wallis test for market capitalization and RMSE per time frame. Significant p-values are shown in **bold**.

p-values, with none being significant, calling for more detailed exploration as different time segments may produce varying outcomes.

The results for the four-hour time frame met our expectations. Interestingly, we found a significantly higher number of meaningful values in the daily time frame compared to when we used the market capitalization category. This suggests that the actual value of market capitalization has a stronger effect than market capitalization categories, making the market capitalization value in dollars a more critical factor.

In conclusion, our data show a consistent negative coefficient across all models, implying that an increase in market capitalization tends to improve model performance. This finding is consistent with our earlier theoretical expectations and supports the premise we set at the beginning of this section.

The results of this section enable us to respond to *RQ2b*, regarding the impact of market factors like volatility and market capitalization on predictive performance. Section § 5.3.1 establishes that the volatility of the testing period significantly influences the accuracy of most of our forecasting models. Furthermore, the results of this section indicate a notable correlation between market capitalization and volatility, as evidenced in Table 5.8. Additionally, the findings from Table 5.9 suggest that market capitalization notably affects the RMSE of numerous forecasting models, particularly in the one-minute and four-hour time frames. Consequently, we deduce that market factors indeed play a crucial role in the predictive accuracy of our forecasting models.

5.4 THE INFLUENCE OF TIME FRAMES ON FORECASTING ACCURACY

In this section, we explore the impact of different time frames on forecasting accuracy, contributing to the answer for *RQ2c*. A substantial shift in certain models' effectiveness is evident when comparing Fig. 5.1 and Fig. 5.11. Furthermore, Table 5.2 indicates that most forecasting models' performance is not uniform across all time frames. Similar outcomes have emerged in prior sections, illustrating that results can vary significantly across different time frames, for example in Section § 5.3.2. This section seeks to determine whether these performance disparities are statistically meaningful. To do so, we need to be able to compare model performance across time frames. Unfortunately, we cannot use the RMSE as it is usually lower for shorter periods due to higher values in longer time frames. Therefore, we use the percentage difference between the forecasting model and the baseline to compare the performances across time frames.

In initial analyses, most forecasting models displayed competence in predicting logarithmic returns over various time frames, with a marked decline in the one-minute time frame. This decline is evident in Fig. 5.12, where the models struggled to account for the quick changes typical of high-frequency trading data. A closer look at certain models, such as XGBoost and LightGBM, revealed an interesting pattern: these models, though less effective on daily time frames, performed well on shorter intervals, likely because they are skilled at detecting brief periods of volatility [75, 160]. Notably, both XGBoost and LightGBM utilize gradient-boosting techniques, potentially explaining their similar performance patterns across various time frames.

We calculate the mean and standard deviations of the performance improvement compared to the baseline, presented in percentages. Our analysis, presented in Table 5.10, shows that most forecasting models have an increased mean and lower standard deviation when using the scaled dataset. The high mean and standard deviation values for the N-BEATS model arise from a significant outlier in its predictions. It is crucial to understand that an increased mean does not necessarily suggest superior performance of these models over those using logarithmic returns. Instead, it signifies these models surpassed ARIMA and maintained more consistent performance across different time frames with the scaled dataset. This pattern emerges as ARIMA's efficacy decreases with the scaled dataset, leading to elevated averages for models that remain stable or have better performance with data scaling, such as TBATS.

To confirm the increased stability across time frames of models using the scaled dataset, we conduct a Kruskal-Wallis test on the percentage differences relative to ARIMA for both datasets. The results, found in Table 5.11, show that there are significant differences across time frames for most models using logarithmic returns. Notably, models like XGBoost, Prophet, and N-BEATS show particularly low p-values, signaling substantial variations in performance over different time frames. On the other hand, models such as Random Forest and TBATS exhibit more consistent performance, evidenced by their higher p-values.

Model	Logarithmic Returns		Scaled Dataset	
	Mean	Standard Deviation	Mean	Standard Deviation
Random Forest	-2.983958	0.967409	1.683653	0.6447525
XGBoost	-3.183445	5.703899	-13.12606	1.953999
LightGBM	0.186235	0.394218	2.598802	1.244567
Prophet	-2.992266	1.619885	1.672939	0.9960628
N-BEATS	-1,885.523740	2,140.444330	-9,672,615	16,753,430
RNN	-30.239770	47.033778	0.6637787	1.423125
LSTM	-7.509715	7.288476	1.893680	0.6911745
GRU	-13.286828	16.989423	-1.888708	1.225226
TCN	-50.824817	70.404857	0.7236629	1.498435
TFT	-421.372721	423.041524	-44.41053	10.34075
N-HiTS	-372.564386	566.791545	-5.826387	2.732547
TBATS	0.013840	0.223743	4.481298	0.4217933

Table 5.10. Mean and standard deviation of the forecasting models' performance improvement compared to ARIMA in percentage.

This consistency is further explained in Table 5.10, where it is evident that the standard deviation for these models remains relatively unchanged between datasets.

Scaling logarithmic returns significantly altered the performance metrics of the majority of the forecasting models, a change observable in the third column of Table 5.11. Most models registered higher p-values, pointing to a less pronounced distinction in performance across time frames. This shift could be due to the scaling process itself, which may modify fundamental data patterns and, consequently, impact model performances. Certain models, including Random Forest and TBATS, maintained steady performance regardless of whether the data was logarithmic returns or scaled, indicating their resilience to changes in data preprocessing. In contrast, LightGBM, TCN, TFT, and N-HiTS continued to exhibit significant p-values, signifying ongoing disparities in performance across various time frames.

Using the results of this section we can address *RQ2c* regarding the extent of influence that time frames have on predictive performance. We show the significant role of time frames in prediction accuracy. Most models struggle with the one-minute interval, suggesting high-frequency trading data poses forecasting challenges. However, certain models, like XGBoost and LightGBM, excel in these shorter periods due to their adeptness at capturing quick volatility. Data scaling, while generally improving forecasts, yields varied effects depending on the model and time frame. Random Forest and TBATS, for instance, show consistent performance regardless of scaling, denoting robustness. In contrast, models like LightGBM, TCN, TFT, and N-HiTS exhibit ongoing performance disparities, indicating sensitivity to time frame adjustments. In conclusion, time frames significantly influence

Model	Logarithmic Returns	Scaled
Random Forest	0.1429	0.1978
XGBoost	2.0893×10^{-11}	0.0841
LightGBM	0.0309	0.0091
Prophet	6.0733×10^{-4}	0.0819
N-BEATS	1.0326×10^{-10}	0.7844
RNN	3.5502×10^{-7}	0.6036
LSTM	2.4267×10^{-5}	0.4237
GRU	1.0555×10^{-6}	0.4324
TCN	1.9287×10^{-10}	0.0338
TFT	1.6790×10^{-11}	5.8291×10^{-11}
N-HiTS	7.2994×10^{-10}	0.0051
TBATS	0.1099	0.1198

Table 5.11. P-values of the Kruskal-Wallis test results for the logarithmic returns and scaled logarithmic returns datasets.

cryptocurrency price predictions, with shorter intervals proving more challenging. Data scaling's effectiveness also varies across models, necessitating careful consideration of both time frame and model adaptability when forecasting cryptocurrency prices.

5.5 IMPACT OF DATA TIME SPAN

This section examines the effect of data quantity on model performance. We investigate this to be able to answer *RQ*3. As discussed in Section § 4.1, we divide our data into five periods, aiming to understand how expanding the training dataset enhances model accuracy. We also explore the impact of a gap between training and testing data on performance.

Our initial step involves analyzing the volatility of each training and testing period, intending to use this insight to explain performance disparities. We hypothesize that training models with data encompassing various volatility types will enhance forecasting capabilities compared to using data from a single period characterized by one volatility type.

Concerns arise when the gap between training and testing data widens, potentially degrading performance, especially if training and testing volatility differ significantly. For example, the initial training period on the daily time frame exhibits rising volatility, with frequent spikes above the 75th percentile. This pattern is absent in subsequent periods, leading us to anticipate a more substantial performance gap for this time frame.

We present a detailed view of the training and testing periods along with volatility statistics in Fig. 5.17. This illustration shows the five periods under consideration. The blue and red lines at the top indicate the data range used for training and testing. Beneath these lines, the corresponding volatility for each period is depicted, highlighting percentiles and

average volatility. We use this data in subsequent subsections to correlate our observations with the volatility specific to each period.

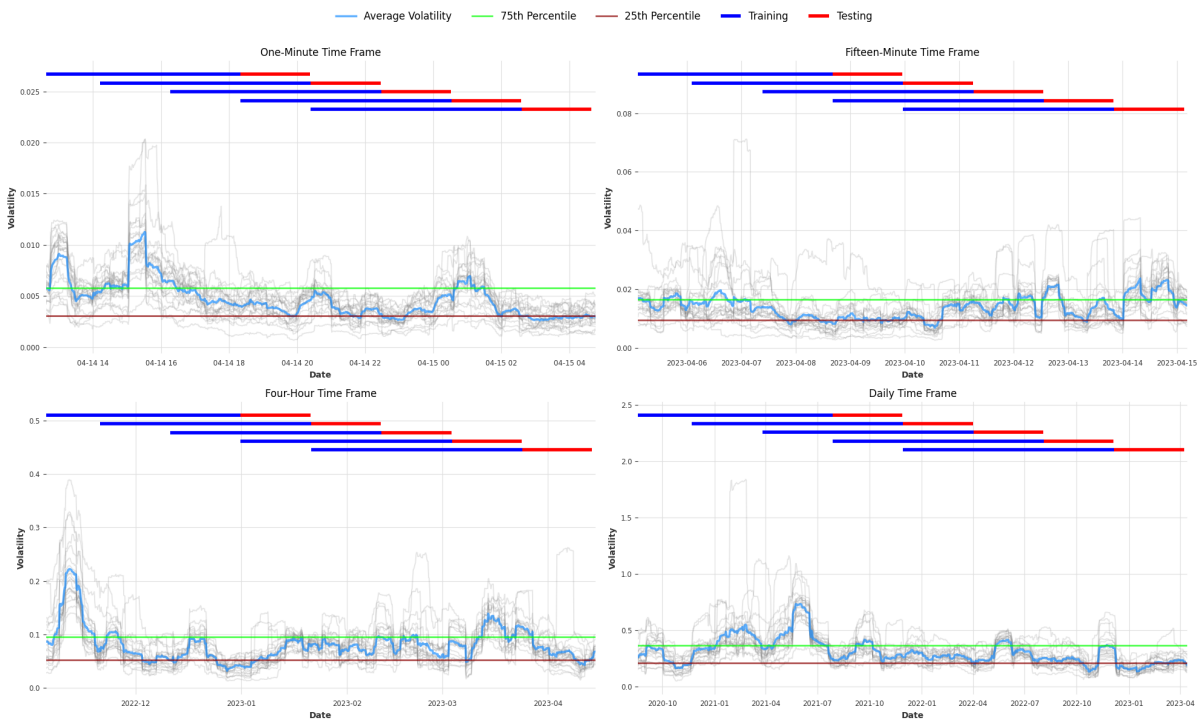

Figure 5.17. Overview of volatility during periods for each time frame.

5.5.1 *Amplification of Training Data*

Section § 5.1 highlighted the suboptimal outcomes shown by deep-learning-based forecasting models. We suggested that this weaker performance could be linked to the amount of training data these models receive. Our goal is to identify an optimal data threshold that might enhance the effectiveness of deep-learning and RNN-based forecasting models, as they currently fall short compared to other types of models. To gauge the influence of training data size, we examine how the models perform under various conditions by keeping the testing dataset consistent while progressively increasing the training data.

We retrained the models, starting with different dates for the training data. This means that the training data for period one includes the largest volume of data, while period five contains the same data volume as outlined in Section § 5.1. Fig. 5.18 presents RMSE boxplots for each time frame and the chosen forecasting models.

Despite augmenting training data, significant improvements were largely absent in the one-minute time frame, with the exception of some models like RNN, LSTM, and TCN, which showed slight performance enhancements but still did not surpass the ARIMA model's effectiveness. N-BEATS, TFT, and N-HiTS remained stagnant, reinforcing their previously discussed underperformance in Section § 5.1. Analysis of the final testing period revealed

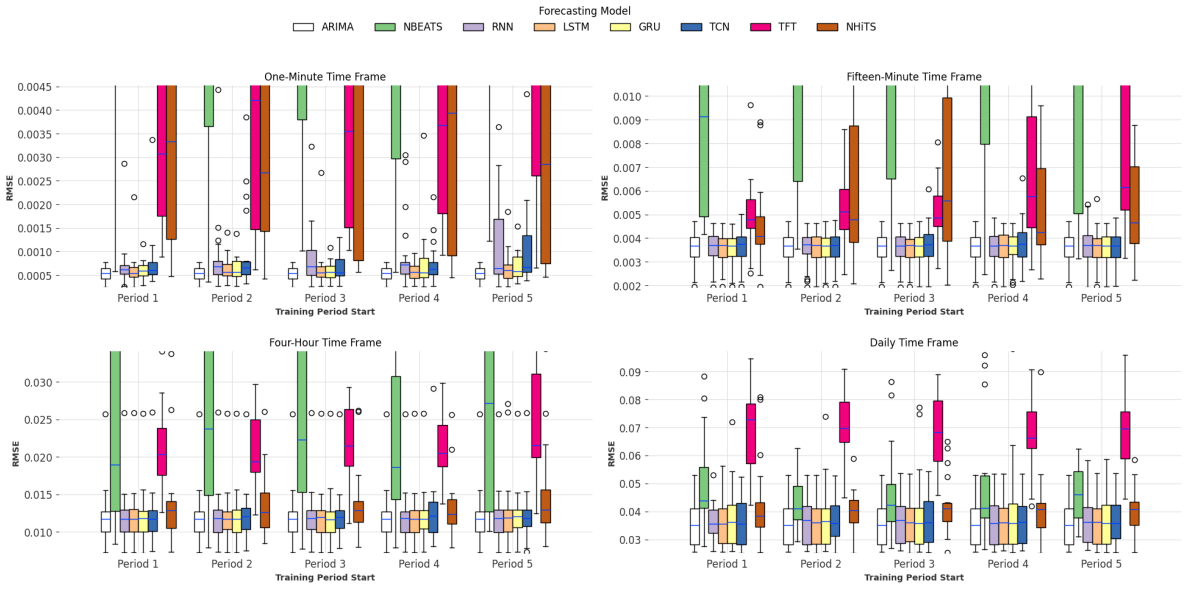

Figure 5.18. Boxplot of RMSE values per starting period of training data.

predominantly low volatility, with minimal similarity between training and testing data, as detailed in Fig. 5.17.

This pattern was also evident in the fifteen-minute time frame, although TFT and N-HiTS improved noticeably. However, most models showed no response to the increased training data. Notably, the final testing period displayed high volatility, similar to the first and last training periods, suggesting that additional training data did not contribute significantly to performance enhancement.

For the four-hour time frame, minor improvements were observed in the RNN, LSTM, GRU, and TCN models. The testing data began with high volatility but shifted to lower levels, a trend adequately reflected in the initial training period, explaining the minimal performance gains despite expanded training data.

In the daily time frame, slight improvements were seen, particularly in the RNN and LSTM models, compared to the final period. The testing period predominantly showed low volatility, similar to the one-minute time frame. Although the final training period captured this characteristic, the initial training period also contained a phase of low volatility, suggesting why extending the training data to the initial period only marginally enhanced performance.

The limited improvement, despite increased training data, might be linked to the hyperparameters used, especially those concerning input chunk length and training length. Since the hyperparameters were initially optimized based on the first data period, they may not be ideal for longer data sequences, potentially affecting model performance. We leave this deeper analysis of the impact of the hyperparameters to future work.

To further investigate if more training data improves forecasting accuracy, we employed the Mann-Whitney U test, comparing the RMSE between the first and fifth periods. The expectation was that more training data would reduce the RMSE, especially since models trained until the fifth period had the most data. However, the results did not strongly support this assumption for most models. Significant p-values were scarce, with a notable exception in the fifteen-minute time frame for the TFT model, which had a p-value of 0.0468, indicating a significant increase in RMSE.

Further analysis comparing each period against the fifth did not reveal additional significant results for the TFT model within the fifteen-minute time frame. This suggests that simply increasing training data does not consistently enhance performance for the models used in this study.

While the division between training and testing data was informed by previous studies [136], there is no definitive rule for the exact amount of data necessary for these segments. Even if such an optimal quantity were determined, it would likely vary based on several factors, including data features, model types, and hyperparameter settings. Unfortunately, this study's exploration was limited, as hyperparameter optimization was based solely on the initial training data size, restricting further inquiry into this aspect.

Addressing the first part of *RQ*3, regarding the sensitivity of the forecasting models to the quantity of training data, our findings demonstrate that simply increasing the amount of training data does not uniformly enhance the accuracy of these models. Though some models and time frames showed minor improvements, the majority remained largely unaffected when presented with additional data.

This outcome underscores the complexity inherent in these models' responses to data volume changes. It also suggests that factors other than data quantity—such as model-specific hyperparameters and the inherent characteristics of the data itself—play crucial roles in forecasting accuracy. The assumption that an increase in training data would automatically result in reduced RMSE values found limited support.

We conclude that the sensitivity of state-of-the-art forecasting models to changes in training data quantity is not straightforward and that merely expanding data volume does not assure improved performance. Further research is essential to unravel the specific conditions and configurations that optimize each model's performance.

5.5.2 *Efficacy in Long-term Forecasting*

The field of time series forecasting is significantly affected by the time dimension in which models are expected to produce predictions. Long-term forecasting proves to be especially challenging due to inherent uncertainties and potential changes in patterns as the forecast horizon expands. This subsection seeks to evaluate the effectiveness of forecasting models for long-term predictions. A central concern is determining whether models, after being trained on the volatility typical of a specific period (referred to as period 1), maintain their

predictive accuracy when faced with periods that have different volatility profiles and are further along the time dimension. This investigation is important as it provides insights into the models' robustness and adaptability amid changing market dynamics, an aspect critical for stakeholders who depend on these forecasts for decision-making. To assess the models' long-term predictive capabilities, we evaluate their RMSE for each starting period of the testing data. Fig. 5.19 presents the boxplots for all forecasting models and time frames.

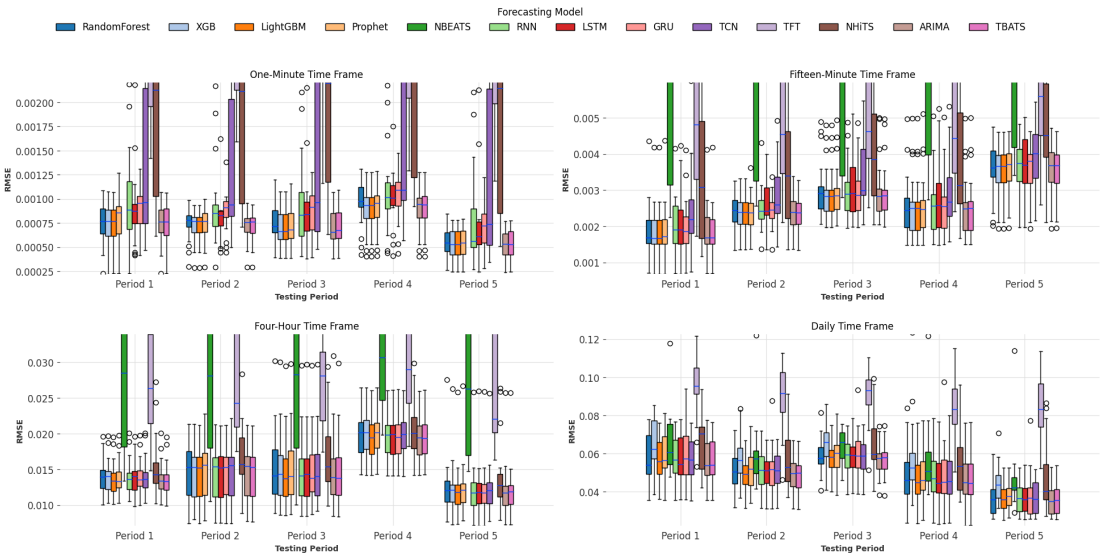

Figure 5.19. Boxplot of RMSE values per starting period of testing data.

The findings are somewhat unexpected. One might assume that models trained on the first period would excel when tested on a similar period. However, the data show other periods surpassing the first in testing performance. Most models show similar performance metrics, with N-BEATS, TFT, and N-HiTS standing out due to their tendency to produce outliers, as previously noted in Section § 5.1. The more effective models maintain their relative performance, as seen in the second period of the daily time frame, where LSTM, TCN, ARIMA, and TBATS all show comparable boxplots.

Analysis of the one-minute time frame indicates the best performance in the fifth period, furthest from our training data. Further investigation into Fig. 5.17 clarifies this occurrence. The fifth period's testing data mostly aligns with the 25th percentile, indicating a low volatility phase. This finding is consistent with insights from Section § 5.3.1, highlighting the significant role of test volatility on forecast accuracy.

This reasoning extends to other time frames as well. For example, the fifteen-minute time frame shows peak performance in the first period, correlating with the lowest test volatility. Similarly, test volatility for both the four-hour and daily time frames decreases in the final periods, aligning with the highest performance points for these intervals.

To confirm the statistical relevance of these findings, we will conduct a Mann-Whitney U test across various time frames, using the initial observations as hypotheses. The sub-

sequent results reveal widespread significance, thus providing statistical support for our findings.

In the one-minute time frame results, most models show significant findings, except for some like N-BEATS, TCN, TFT, and N-HiTS. These results show that the fifth period was the easiest to forecast for most models. This consistency aligns with our initial observations, suggesting these models' limitations in capturing time series data effectively within this time frame.

In the fifteen-minute time frame, we see a lot of significant results again, meaning that the first period was statistically the easiest to predict. The results largely reflect the previous time frame regarding the significant results of the forecasting models. However, this time TCN shows significance in the comparison of its performance between the first and fifth periods.

In the four-hour time frame, a change is evident as all results for N-HiTS and TCN are significant. This shift is due to these models' slightly enhanced performance at higher time frames. Thus, we can state that the fifth period was statistically easier to predict compared to the other periods for almost all models, with the exception of N-BEATS and TFT.

In the daily time frame, numerous significant results are found. Almost all models show significance, except for TFT, its results are only significant if we compare the performance in the fifth period with the first. This shows that our observation that the fifth period would be the easiest to predict due to its low volatility is statistically correct, reaffirming the essential role of test volatility in forecast model performance.

In addressing the second part of *RQ*3, which asks if models maintain their prediction quality over time, our study finds big differences in how well time series forecasting models perform under various time lengths and volatility situations. Models like LSTM, TCN, and ARIMA are steady and perform well, adjusting easily to different situations. On the other hand, N-BEATS, TFT, and N-HiTS are less stable, often doing poorly when conditions change from what they were trained on. There's a strong connection between less volatility and better predictions, confirmed by the Mann-Whitney U test, showing these models' performance in different settings. So, while many models can predict over long times, how well they do largely depends on the volatility during the tested times, supporting our earlier findings in Section § 5.3.1.

5.6 LIMITATIONS AND FUTURE WORK

In this section we examine essential elements of our research methodology, focusing on limitations, possible enhancements, and areas for future study. The following sections analyze crucial aspects of our thesis, including the quality and applicability of our data, the effectiveness of our models, and the limitations we met during hyperparameter tuning.

5.6.1 *Data Robustness and Representativeness*

This discussion begins by addressing a fundamental limitation: we rely on only 1000 timesteps for each time frame. This restriction results in the absence of certain volatility combinations during the training and testing stages, as evident in the heatmaps of Fig. 5.14. A larger data set would help overcome this limitation by filling the identified gaps. Nevertheless, practical challenges arise, particularly for the daily time frame, as many cryptocurrencies have not yet existed for more than 1000 days. An alternative approach, given more time, could involve employing a 12-hour time frame instead, effectively doubling the number of data points and potentially revealing a broader range of volatility combinations. Furthermore, increasing the number of data points offers a potential solution to the current imbalance in data attributes. This study reveals pronounced disparities among the classes of conditional heteroskedasticity, trend, and stochasticity. Introducing more data could enable a more even distribution across these categories, enhancing the robustness of future analyses.

We use a thirty-time-step window to calculate historical volatility, as described in Section § 3.2.7. For volatility classification, we employ percentiles, forming three unique groups. Future studies could explore varying window sizes to better understand both short-term and long-term volatility trends. Another approach might involve expanding the classifications for volatility. For example, earlier research has established twenty volatility categories using multiple percentiles [132], providing a pathway for a more detailed volatility analysis within the dataset. Beyond the percentile-based system, there is a variety of other volatility classification methods. Upcoming research could consider more advanced techniques, such as the Z-score method or classifications based on machine learning, to achieve a more comprehensive understanding of volatility categorization.

5.6.2 *Model Optimization and Performance*

In this thesis, we choose a mix of different time series forecasting models, detailed in Chapter 4. However, most of these models were not specifically designed for cryptocurrency time series. Further research might consider including more models suited for this unique context like the CNN-LSTM model [161] and the deep state space model [162] crafted for cryptocurrency forecasting. Although numerous models are available, accessing their code can be challenging, often compelling researchers to develop these models from scratch.

In order to employ a variety of recognized forecasting models, we selected univariate models for our analysis, thereby constraining our data to a single variable, such as price or logarithmic returns. We theorize that using multivariate models, which consider multiple variables like closing, opening, highest, and lowest prices, could improve forecasts. Previous studies indicate that regulatory events can negatively affect cryptocurrency re-

turns [163]. Collecting such data through news or social media could strengthen model performance, leading to more precise forecasts.

For hyperparameter optimization, this study used the RMSE, necessitating a single performance metric for our models. However, the realm of forecasting metrics is wide, with alternatives like Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Mean Squared Error (MSE) each having their strengths and weaknesses. While this study focused on one metric, employing a mix of metrics could provide a richer analysis in future studies.

One enhancement for upcoming research could be transitioning from a regression model to a classification framework. While this may reduce forecast detail, it might boost predictive performance. Future work could create forecast classifications and judge model performance based on these classes. For example, instead of predicting a precise percentage increase, models could forecast general price trends—increasing, decreasing, or stabilizing. This method, requiring less specific predictions, could offer valuable insights to investors while potentially improving prediction accuracy. A useful starting point could be Akyildirim et al.'s work (2021), which used binary classification for cryptocurrency forecasts [164].

Additionally, using a model ensemble is an intriguing possibility, as previous studies suggest this can enhance performance [68]. Combining the top five models from this thesis might yield better results than one standalone model. Future research could integrate models in such a way that each model's strengths and weaknesses are complemented by others, potentially boosting overall forecast efficacy.

5.6.3 *Limitions of Hyperparameter Tuning*

In Section § 4.2, we showed that our search space for hyperparameter tuning was grounded in existing research. Nonetheless, we identify opportunities for enhancement within this process. Primarily, the consideration of a more expansive search space, especially regarding the default parameters illustrated in Table 4.1, emerges as a viable point of refinement. This proposition stems from our belief that augmenting the look-back period could enhance performance, particularly on elongated time series, as discussed in Section § 5.5. Furthermore, integrating a greater number of hyperparameters represents another potential avenue for improvement. The scope of the present research excluded exhaustive hyperparameter optimization due to time constraints. However, we hypothesize that broadening the search space could facilitate superior model performance. Subsequent research might also consider amplifying the number of trials in the hyperparameter tuning phase. While this would invariably demand additional time, it may also offer the advantage of pinpointing the optimal hyperparameter combination.

Hyperparameter tuning was conducted solely on the initial period of each dataset within the scope of our research. A prospective enhancement to this approach might involve the utilization of all periods or an alteration in the testing method during hyperparameter

tuning to encompass all periods. Employing this strategy would integrate more training data and various volatility durations, potentially elevating the overall model performance. Another potential methodology for enhancing model performance involves activating the "retrain" parameter for all models. Enabling this feature would compel the Darts library to retrain the model at each time step during one-step forecasting. Despite the potential for improved outcomes, we elected not to employ this strategy due to the additional time required to fit and train the models. However, future research might achieve heightened performance by retraining the models at each time step.

5.7 DISCUSSION

Hereafter, we summarize our results with respect to the initial research questions.

Summary for *RQ*1**:** *How do the performances of state-of-the-art forecasting models compare to traditional models in forecasting cryptocurrency prices?*

Answer: Our selection of state-of-the-art models demonstrate strong and reliable performances. However, they do not consistently surpass the traditional model, ARIMA, which often matches or even exceeds the forecasting accuracy of these more advanced models. In conclusion, the majority of the state-of-the-art models were not able to outperform the traditional model. However, TBATS and LightGBM performed better in three out of the four time frames. XGBoost did perform better on half of time frames. Thus, showing that a small selection of state-of-the-art models is able to outperform it.

Summary for *RQ*2**:** *What effects do data properties and market factors have on cryptocurrency prediction models?*

Answer: Regarding the data properties, we observe that while autocorrelation, seasonality, conditional heteroskedasticity, and stochasticity do not impact volatility or forecasting effectiveness, trend, and unconditional heteroskedasticity emerge as significant factors. Specifically, datasets with a 'trending' nature show superior forecasting performance, particularly in extended time frames. Conversely, homoskedastic datasets outshine in shorter durations, a finding substantiated by Mann-Whitney U tests. Our analysis confirms that the volatility inherent in the testing periods markedly influences most forecasting models. A notable interplay exists between market capitalization and volatility. Furthermore, market capitalization distinctly affects the RMSE of various models, more pronouncedly in one-minute and four-hour time frames, underlining the critical role market conditions play in forecasting precision. We identify significant disparities in predictive accuracy due to time frames. High-frequency, one-minute trading intervals pose substantial challenges for most models, highlighting the intricacies of such forecasts. In conclusion, market conditions and time frames affected the models the most, whereas only a few of the data properties had a significant impact on predictive performance.

Summary for *RQ*3**:** *How sensitive are state-of-the-art forecasting models to the quantity of training data, and to what degree is prediction performance sustained over time?*

Answer: Our study reveals complex dynamics in forecasting models' behavior with respect to training data volume and prediction sustainability over time. Increasing the amount of training data does not uniformly enhance model accuracy; instead, performance depends on intricate factors like model-specific parameters and data properties, indicating that mere data expansion does not assure improved precision. Regarding sustained performance, models like LSTM, TCN, and ARIMA demonstrate adaptability and consistency, contrasting with N-BEATS, TFT, and N-HiTS, which struggle under varying conditions. A distinct correlation between lower volatility and increased forecasting accuracy is evident, affirming that while models can predict over long periods, their effectiveness is largely influenced by the volatility of the testing periods.

6

CONCLUSIONS

In this thesis, we conduct a systematic benchmark study to evaluate the effectiveness of time series forecasting models in the unpredictable field of cryptocurrency price prediction. Unlike previous studies that focus on a single cryptocurrency or a limited set of models, our research covers twenty-one cryptocurrencies, thirteen forecasting models, and four time frames, filling a gap in the current literature.

We find that traditional models can sometimes equal or surpass the performance of advanced models, challenging the usual preference for state-of-the-art techniques. While many data properties and market factors have little impact on prediction accuracy, results are significantly influenced by factors such as heteroskedasticity, trends, market capitalization, and volatility. Notably, having more extensive training data does not consistently improve the performance of deep-learning and RNN-based models. Furthermore, longer forecasting periods do not necessarily reduce accuracy, with outcomes largely depending on market volatility.

Nevertheless, our study acknowledges several limitations that offer possibilities for future research. Using only 1000 timesteps per time frame restricted our dataset, potentially skewing volatility patterns. Larger datasets or shorter time frames could improve future studies, as could exploring other volatility classifications and multivariate models.

Our forecasting models, while varied, were not cryptocurrency-specific, highlighting a need for models tailored to this sector, like the CNN-LSTM or deep state space models, for more precise predictions. Switching from regression to classification might also simplify and improve forecasts.

Additionally, this thesis is the first to assess the N-HiTS model's performance in cryptocurrency price prediction. Therefore, providing a foundation for further examinations of its capabilities and constraints.

7

APPENDIX

7.1 STATISTICAL TEST RESULTS OF DATA PROPERTIES

Model	Daily	Four-Hour	Fifteen-Minute	One-Minute
TBATS	0.048765	0.014883	0.002659	0.000716
Random Forest	0.050304	0.015334	0.002703	0.000745
LSTM	0.050308	0.015209	0.002728	0.000729
TCN	0.050495	0.015312	0.002819	0.000741
LightGBM	0.050575	0.015214	0.002681	0.000718
Prophet	0.051061	0.015276	0.002681	0.000742
ARIMA	0.051457	0.015503	0.002767	0.000750
RNN	0.051816	0.015350	0.002727	0.000734
N-HiTS	0.052677	0.016749	0.002880	0.000815
GRU	0.052958	0.016086	0.002782	0.000754
N-BEATS	0.057311	0.018082	831.837011	0.000855
XGBoost	0.058538	0.017116	0.003060	0.000861
TFT	0.079329	0.020879	0.003495	0.001111

Table 7.1. Aggregated model performance on the scaled logarithmic returns.

Model	One-Minute	Fifteen-Minute	Four-Hour	Daily
Random Forest	-4.380903	-1.647853	-2.903341	-3.003736
XGBoost	0.107630	0.697849	-0.503819	-13.035441
LightGBM	0.118552	0.583836	0.471021	-0.428470
Prophet	-3.401823	-0.603883	-2.827096	-5.136261
N-BEATS	-5382.545693	-1873.084866	-256.216336	-30.248063
RNN	-111.575029	-7.384274	-0.305003	-1.694776
LSTM	-18.291009	-10.084308	-0.366565	-1.296976
GRU	-42.471012	-3.165716	-0.696855	-6.813727
TCN	-172.296525	-19.303487	-9.903662	-1.795594
TFT	-1130.092942	-359.297309	-115.995532	-80.105102
N-HiTS	-1352.225389	-105.716426	-18.976548	-13.339179
TBATS	-0.359171	0.086713	0.237237	0.090579

Table 7.2. Percentage difference in performance between forecasting model and ARIMA on logarithmic returns.

Model	One-Minute	Fifteen-Minute	Four-Hour	Daily
Random Forest	0.867147	2.374308	1.241000	2.252157
XGBoost	-15.582786	-11.14450	-11.276411	-14.500535
LightGBM	4.400550	3.065837	1.745192	1.183628
Prophet	1.528006	3.239212	1.454329	0.470210
N-BEATS	-15.285371	-38690410	-17.465447	-11.803511
RNN	2.024410	1.196445	1.161741	-1.727481
LSTM	2.809717	0.8617838	1.962632	1.940587
GRU	-0.448445	-1.118328	-3.662386	-2.325671
TCN	1.496034	-1.850945	1.367836	1.881726
TFT	-50.293794	-29.94132	-40.133357	-57.273634
N-HiTS	-9.540513	-3.894537	-7.253894	-2.616604
TBATS	4.645697	4.077644	4.104410	5.097440

Table 7.3. Percentage difference in performance between forecasting model and ARIMA on scaled logarithmic returns.

Model	One-Minute	Fifteen-Minute	Four-Hour	Daily
Random Forest	0.514	0.625	0.543	0.457
XGBoost	0.486	0.652	0.682	0.318
LightGBM	0.486	0.652	0.543	0.457
Prophet	0.458	0.702	0.514	0.244
N-BEATS	0.348	0.813	0.856	0.945
RNN	0.727	0.652	0.572	0.400
LSTM	0.625	0.514	0.543	0.344
GRU	0.598	0.570	0.628	0.486
TCN	0.953	0.625	0.888	0.400
TFT	0.458	0.119	0.732	0.457
N-HiTS	0.750	0.458	0.953	0.244
ARIMA	0.514	0.677	0.600	0.428
TBATS	0.514	0.652	0.543	0.457

Table 7.4. P-values from the Mann-Whitney U test for autocorrelation.

Test	Fifteen-Minute	Four-Hour	Daily
Breusch-Pagan	0.110	0.060	0.343
Goldfeld-Quandt	0.006	1.000	-

Table 7.5. Mann-Whitney U test p-values under Breusch-Pagan and Goldfeld-Quandt tests for volatility.

Model	Intercept	Coef	R^2	$P > \|t\|$	F-statistic
Random Forest	-0.040	0.108	0.897	3.14×10^{-42}	713.941
XGBoost	-0.044	0.119	0.897	2.76×10^{-42}	716.450
LightGBM	-0.039	0.105	0.906	8.69×10^{-44}	786.621
Prophet	-0.040	0.110	0.905	9.82×10^{-44}	784.029
N-BEATS	0.021	0.050	0.016	0.246	1.368
RNN	-0.038	0.106	0.908	3.59×10^{-44}	805.554
LSTM	-0.039	0.106	0.899	1.60×10^{-42}	727.081
GRU	-0.041	0.112	0.871	3.46×10^{-38}	552.535
TCN	-0.037	0.105	0.897	3.65×10^{-42}	711.015
TFT	-0.056	0.170	0.781	8.86×10^{-29}	292.566
N-HiTS	-0.034	0.104	0.783	5.95×10^{-29}	296.219
ARIMA	-0.038	0.105	0.904	1.77×10^{-43}	771.694
TBATS	-0.038	0.105	0.904	1.86×10^{-43}	770.658

Table 7.6. OLS regression results for seasonality.

Model	1m	15m	4h	1d
Random Forest	0.930	0.304	0.329	0.931
XGBoost	0.887	0.313	0.236	0.831
LightGBM	0.889	0.307	0.286	0.948
Prophet	0.696	0.335	0.272	0.725
N-BEATS	0.354	0.952	0.036	0.937
RNN	0.243	0.224	0.281	0.722
LSTM	0.283	0.170	0.269	0.920
GRU	0.296	0.486	0.265	0.642
TCN	0.327	0.501	0.084	0.880
TFT	0.416	0.939	0.604	0.368
N-HiTS	0.236	0.019	0.425	0.277
ARIMA	0.888	0.281	0.260	0.871
TBATS	0.861	0.303	0.286	0.929

Table 7.7. P-values of OLS regression on seasonal strength per time frame

7.2 STATISTICAL TEST RESULTS OF VOLATILITY

Model	Low	Normal
Random Forest	0.000208	0.031226
XGBoost	0.000329	0.024226
LightGBM	0.000329	0.024226
Prophet	0.000208	0.002802
N-BEATS	0.541104	0.811248
RNN	0.003423	0.092792
LSTM	0.000079	0.127163-
GRU	0.573694	0.154497
TCN	0.410183	0.004852
TFT	0.136959	0.195448
N-HiTS	0.922831	0.379640
ARIMA	0.000283	0.027155
TBATS	0.000383	0.023537

Table 7.8. P-values of Mann-Whitney U test for volatility categories on the one-minute time frame.

Model	Low	Normal	High
Random Forest	0.029637	0.075903	0.1
XGBoost	0.029637	0.045155	0.1
LightGBM	0.029637	0.038796	0.1
Prophet	0.029637	0.057592	0.1
N-BEATS	0.754579	0.981980	0.8
RNN	0.029637	0.054910	0.1
LSTM	0.009990	0.025331	0.1
GRU	0.021312	0.031472	0.1
TCN	0.114219	0.111236	0.2
TFT	0.040626	0.819122	0.6
N-HiTS	0.668998	0.848605	0.1
ARIMA	0.029637	0.054910	0.1
TBATS	0.029637	0.057592	0.1

Table 7.9. P-values of Mann-Whitney U test for volatility categories on the fifteen-minute time frame.

Model	Low	Normal	High
Random Forest	0.206349	0.736000	0.606494
XGBoost	0.142857	0.524666	0.606494
LightGBM	0.206349	0.653933	0.606494
Prophet	0.365079	0.378530	0.464286
N-BEATS	0.984127	0.200057	0.359091
RNN	0.206349	0.602510	0.606494
LSTM	0.277778	0.658486	0.640909
GRU	0.206349	0.663016	0.606494
TCN	0.452381	0.676462	0.535714
TFT	0.793651	0.020592	0.992857
N-HiTS	0.547619	0.975433	0.606494
ARIMA	0.206349	0.685302	0.640909
TBATS	0.206349	0.649358	0.606494

Table 7.10. P-values of Mann-Whitney U test for volatility categories on the four-hour time frame.

Model	Low	Normal	High
Random Forest	0.021189	0.000973	0.285714
XGBoost	0.016541	0.000006	1.000000
LightGBM	0.016541	0.000403	0.285714
Prophet	0.005195	0.000157	0.285714
N-BEATS	0.003691	0.059106	0.571429
RNN	0.021189	0.000063	0.142857
LSTM	0.016541	0.000157	0.142857
GRU	0.012850	0.007878	0.142857
TCN	0.016541	0.000285	0.142857
TFT	0.021189	0.000009	0.714286
N-HiTS	0.516473	0.002609	0.428571
ARIMA	0.016541	0.000234	0.285714
TBATS	0.021189	0.000544	0.142857

Table 7.11. P-values of Mann-Whitney U test for volatility categories on the daily time frame.

Model	Intercept	Coef	R^2	P-value	F-statistic
Random Forest	0.000774	-6.29e-16	0.216416	**0.033583**	5.247545
XGBoost	0.000742	-6.02e-16	0.223716	**0.030352**	5.475590
LightGBM	0.000742	-6.02e-16	0.223759	**0.030334**	5.476940
Prophet	0.000771	-6.54e-16	0.212125	**0.035631**	5.115493
N-BEATS	0.028000	4.81e-15	0.000186	0.953171	0.003541
RNN	0.001494	-1.70e-15	0.032509	0.434154	0.638434
LSTM	0.000847	-3.04e-16	0.024691	0.496355	0.481014
GRU	0.001047	-1.06e-15	0.083355	0.204344	1.727763
TCN	0.001914	-5.44e-16	0.001525	0.866536	0.029019
TFT	0.008538	-6.84e-15	0.023693	0.505304	0.461088
N-HiTS	0.006975	2.41e-14	0.064770	0.265576	1.315862
ARIMA	0.000741	-5.94e-16	0.226275	**0.029291**	5.556534
TBATS	0.000745	-6.03e-16	0.223796	**0.030318**	5.478110

Table 7.12. P-values of OLS regression for volatility on the one-minute time frame.

Model	Intercept	Coef	R^2	P-value	F-statistic
Random Forest	0.002769	-1.39e-15	0.135891	0.100103	2.987956
XGBoost	0.002706	-1.40e-15	0.139712	0.095092	3.085639
LightGBM	0.002710	-1.41e-15	0.139365	0.095537	3.076718
Prophet	0.002747	-1.47e-15	0.146009	0.087377	3.248489
N-BEATS	0.049227	-5.71e-14	0.004065	0.783640	0.077560
RNN	0.002910	-9.32e-16	0.032912	0.431275	0.646607
LSTM	0.003001	-2.05e-15	0.095418	0.173051	2.004170
GRU	0.002803	-1.58e-15	0.162455	0.070035	3.685359
TCN	0.003123	-1.21e-15	0.071572	0.241024	1.464708
TFT	0.012799	-1.92e-14	0.096206	0.171194	2.022492
N-HiTS	0.005521	-4.95e-15	0.047037	0.345005	0.937822
ARIMA	0.002727	-1.37e-15	0.134927	0.101407	2.963467
TBATS	0.002726	-1.42e-15	0.144329	0.089372	3.204801

Table 7.13. P-values of OLS regression for volatility on the fifteen-minute time frame.

Model	Intercept	Coef	R^2	P-value	F-statistic
Random Forest	0.015842	-1.03e-14	0.266172	0.016664	6.891630
XGBoost	0.015408	-9.11e-15	0.213966	**0.034738**	5.171980
LightGBM	0.015281	-9.30e-15	0.232596	**0.026819**	5.758809
Prophet	0.015707	-8.20e-15	0.168643	0.064429	3.854195
N-BEATS	0.052953	-3.56e-14	0.007531	0.708371	0.144180
RNN	0.015392	-9.37e-15	0.237665	**0.024981**	5.923416
LSTM	0.015411	-9.35e-15	0.230313	**0.027688**	5.685359
GRU	0.015459	-9.55e-15	0.240455	**0.024020**	6.014966
TCN	0.016694	-1.09e-14	0.196981	**0.043859**	4.660724
TFT	0.027441	2.98e-14	0.125412	0.115252	2.724506
N-HiTS	0.017921	-1.22e-14	0.163193	0.069342	3.705362
ARIMA	0.015356	-9.18e-15	0.224435	**0.030050**	5.498261
TBATS	0.015323	-9.30e-15	0.231145	**0.027368**	5.712075

Table 7.14. P-values of OLS regression for volatility on the four-hour time frame.

Model	Intercept	Coef	R^2	P-value	F-statistic
Random Forest	0.051896	-3.23e-14	0.226938	**0.029022**	5.577594
XGBoost	0.056762	-3.60e-14	0.285074	**0.012667**	7.576179
LightGBM	0.050347	-2.92e-14	0.252519	**0.020254**	6.418712
Prophet	0.052925	-3.62e-14	0.340770	**0.005465**	9.821485
N-BEATS	0.062627	-1.45e-14	0.020394	0.536882	0.395556
RNN	0.051037	-3.06e-14	0.292966	**0.011279**	7.872830
LSTM	0.051037	-3.29e-14	0.276642	**0.014324**	7.266400
GRU	0.053959	-3.92e-14	0.193773	**0.045823**	4.566567
TCN	0.051181	-3.25e-14	0.275251	**0.014616**	7.215965
TFT	0.088846	-6.49e-14	0.117056	0.128990	2.518919
N-HiTS	0.055074	-2.11e-14	0.097866	0.167354	2.061171
ARIMA	0.050341	-3.14e-14	0.273877	**0.014910**	7.166360
TBATS	0.050250	-3.14e-14	0.275621	**0.014538**	7.229374

Table 7.15. P-values of OLS regression for volatility on the daily time frame.

7.3 STATISTICAL TEST RESULTS ON THE EFFICIENCY OF LONG-TERM FORECASTING

Model	Period 1	Period 2	Period 3	Period 4
Random Forest	7.63×10^{-4}	2.59×10^{-4}	1.17×10^{-3}	2.34×10^{-6}
XGBoost	5.87×10^{-4}	2.14×10^{-4}	3.05×10^{-3}	6.01×10^{-6}
LightGBM	5.87×10^{-4}	2.14×10^{-4}	3.05×10^{-3}	6.01×10^{-6}
Prophet	7.63×10^{-4}	4.10×10^{-4}	4.13×10^{-3}	8.48×10^{-6}
N-BEATS	0.470	0.480	0.470	0.440
RNN	0.031	0.015	0.030	3.83×10^{-3}
LSTM	0.009	0.007	0.016	7.00×10^{-4}
GRU	0.010	0.021	0.025	3.75×10^{-4}
TCN	0.091	0.066	0.109	0.025
TFT	0.540	0.480	0.470	0.510
N-HiTS	0.372	0.353	0.410	0.257
ARIMA	3.12×10^{-4}	1.09×10^{-4}	1.50×10^{-3}	6.01×10^{-6}
TBATS	5.37×10^{-4}	1.95×10^{-4}	2.07×10^{-3}	5.36×10^{-6}

Table 7.16. P-values from the Mann-Whitney test comparing fifth period performance to other periods in the one-minute time frame.

Model	Period 2	Period 3	Period 4	Period 5
Random Forest	9.06×10^{-4}	6.75×10^{-6}	0.001	2.08×10^{-6}
XGBoost	0.001	1.48×10^{-5}	0.002	2.34×10^{-6}
LightGBM	0.001	1.85×10^{-5}	0.003	2.34×10^{-6}
Prophet	0.001	1.06×10^{-5}	0.001	2.08×10^{-6}
N-BEATS	0.372	0.344	0.307	0.210
RNN	0.004	6.57×10^{-5}	0.005	3.35×10^{-6}
LSTM	0.018	0.001	0.014	2.85×10^{-5}
GRU	0.001	3.17×10^{-5}	0.003	2.98×10^{-6}
TCN	0.035	0.002	0.054	2.36×10^{-4}
TFT	0.372	0.249	0.362	0.114
N-HiTS	0.241	0.100	0.189	0.014
ARIMA	0.001	1.33×10^{-5}	0.002	1.84×10^{-6}
TBATS	0.001	1.06×10^{-5}	0.002	2.08×10^{-6}

Table 7.17. P-values from the Mann-Whitney test comparing first period performance to other periods in the fifteen-minute time frame.

Model	Period 1	Period 2	Period 3	Period 4
Random Forest	0.010	0.006	0.007	4.10×10^{-7}
XGBoost	0.006	0.008	0.009	6.83×10^{-7}
LightGBM	0.006	0.006	0.007	5.30×10^{-7}
Prophet	0.022	0.012	0.015	1.13×10^{-6}
N-BEATS	0.410	0.290	0.381	0.095
RNN	0.005	0.006	0.010	5.30×10^{-7}
LSTM	0.006	0.007	0.010	4.66×10^{-7}
GRU	0.007	0.004	0.007	5.30×10^{-7}
TCN	0.009	0.008	0.016	8.78×10^{-7}
TFT	0.145	0.265	0.145	0.033
N-HiTS	0.046	0.014	0.021	1.66×10^{-5}
ARIMA	0.004	0.004	0.006	4.10×10^{-7}
TBATS	0.007	0.005	0.006	4.10×10^{-7}

Table 7.18. P-values from the Mann-Whitney test comparing fifth period performance to other periods in the four-hour time frame.

Model	Period 1	Period 2	Period 3	Period 4
Random Forest	1.44×10^{-6}	5.35×10^{-5}	3.17×10^{-7}	0.003
XGBoost	2.64×10^{-6}	9.84×10^{-5}	1.27×10^{-6}	0.004
LightGBM	2.78×10^{-7}	1.19×10^{-5}	7.30×10^{-8}	0.002
Prophet	2.44×10^{-7}	2.06×10^{-5}	1.10×10^{-7}	0.001
N-BEATS	8.48×10^{-6}	1.09×10^{-4}	8.48×10^{-6}	0.003
RNN	2.14×10^{-7}	2.56×10^{-5}	1.64×10^{-7}	0.001
LSTM	9.95×10^{-7}	5.35×10^{-5}	2.14×10^{-7}	0.003
GRU	9.49×10^{-6}	2.59×10^{-4}	2.64×10^{-6}	0.013
TCN	2.98×10^{-6}	6.57×10^{-5}	1.26×10^{-7}	0.003
TFT	0.023	0.100	0.069	0.420
N-HiTS	4.49×10^{-4}	0.004	1.95×10^{-4}	0.030
ARIMA	1.87×10^{-7}	2.06×10^{-5}	1.10×10^{-7}	0.001
TBATS	2.44×10^{-7}	3.17×10^{-5}	1.10×10^{-7}	0.002

Table 7.19. P-values from the Mann-Whitney test comparing fifth period performance to other periods in the daily time frame.

7.4 ETHICS AND PRIVACY

The Ethics and Privacy Quick Scan of the Utrecht University Research Institute of Information and Computing Sciences was conducted. It classified this research as low-risk with no fuller ethics review or privacy assessment required.

BIBLIOGRAPHY

[1] M. A. Ammer and T. H. Aldhyani, 'Deep learning algorithm to predict cryptocurrency fluctuation prices: Increasing investment awareness', *Electronics*, vol. 11, no. 15, p. 2349, 2022.

[2] S. Corbet, Y. G. Hou, Y. Hu, C. Larkin and L. Oxley, 'Any port in a storm: Cryptocurrency safe-havens during the covid-19 pandemic', *Economics Letters*, vol. 194, p. 109 377, 2020.

[3] J. Chun, J. Ahn, Y. Kim and S. Lee, 'Using deep learning to develop a stock price prediction model based on individual investor emotions', *Journal of Behavioral Finance*, vol. 22, no. 4, pp. 480–489, 2021.

[4] N. Mgadmi, A. Béjaoui and W. Moussa, 'Disentangling the nonlinearity effect in cryptocurrency markets during the covid-19 pandemic: Evidence from a regime-switching approach', *Asia-Pacific Financial Markets*, pp. 1–17, 2022.

[5] A. M. Khedr, I. Arif, M. El-Bannany, S. M. Alhashmi and M. Sreedharan, 'Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey', *Intelligent Systems in Accounting, Finance and Management*, vol. 28, no. 1, pp. 3–34, 2021.

[6] T. Dimpfl and F. J. Peter, 'Nothing but noise? price discovery across cryptocurrency exchanges', *Journal of Financial Markets*, vol. 54, p. 100 584, 2021.

[7] G. P. Bellocca, G. Attanasio, L. Cagliero and J. Fior, 'Leveraging the momentum effect in machine learning-based cryptocurrency trading', *Machine Learning with Applications*, vol. 8, p. 100 310, 2022.

[8] L. Ranaldi, M. Gerardi and F. Fallucchi, 'Crypto net: Using auto-regressive multi-layer artificial neural networks to predict financial time series', *Information*, vol. 13, no. 11, p. 524, 2022.

[9] P. Chaim and M. P. Laurini, 'Nonlinear dependence in cryptocurrency markets', *The North American Journal of Economics and Finance*, vol. 48, pp. 32–47, 2019.

[10] *Inverse relationship between market capitalization and volatility*, 2019. [Online]. Available: https://messari.io/report/messari-research-inverse-relationship-between-market-capitalization-and-volatility.

[11] S. Seth, *Market capitalization: What it is, formula for calculating it*, 2022. [Online]. Available: https://www.investopedia.com/investing/market-capitalization-defined/.

[12] A. Tealab, 'Time series forecasting using artificial neural networks methodologies: A systematic review', *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 334–340, 2018.

[13] J. Du Preez and S. F. Witt, 'Univariate versus multivariate time series forecasting: An application to international tourism demand', *International Journal of Forecasting*, vol. 19, no. 3, pp. 435–451, 2003.

[14] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[15] *Volatility*, 2023. [Online]. Available: https://corporatefinanceinstitute.com/resources/capital-markets/volatility-vol/.

[16] A. Hayes, *Heteroscedasticity definition: Simple meaning and types explained*, 2023. [Online]. Available: https://www.investopedia.com/terms/h/heteroskedasticity.asp.

[17] S. Pincus and R. E. Kalman, 'Irregularity, volatility, risk, and financial market time series', *Proceedings of the National Academy of Sciences*, vol. 101, no. 38, pp. 13 709–13 714, 2004.

[18] J. M. Maheu, T. H. McCurdy and Y. Song, 'Components of bull and bear markets: Bull corrections and bear rallies', *Journal of Business & Economic Statistics*, vol. 30, no. 3, pp. 391–403, 2012.

[19] M. Abolghasemi, E. Beh, G. Tarr and R. Gerlach, 'Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion', *Computers & Industrial Engineering*, vol. 142, p. 106 380, 2020.

[20] E. Sentana and S. Wadhwani, 'Feedback traders and stock return autocorrelations: Evidence from a century of daily data', *The Economic Journal*, vol. 102, no. 411, pp. 415–425, 1992.

[21] S. Park and H. W. Park, 'Diffusion of cryptocurrencies: Web traffic and social network attributes as indicators of cryptocurrency performance', *Quality & Quantity*, vol. 54, no. 1, pp. 297–314, 2020.

[22] M. P. Kumar and N. M. Kumara, 'Market capitalization: Pre and post covid-19 analysis', *Materials Today: Proceedings*, vol. 37, pp. 2553–2557, 2021.

[23] M. Chessar and V. Bellarmine, 'Effect of firms' market capitalization on stock market volatility of companies listed at the nairobi securities exchange', *A Research Project submitted for the degree of MBA, University of Nairobi*, 2015.

[24] A. Barone, *What are small-cap stocks, and are they a good investment?*, 2022. [Online]. Available: https://www.investopedia.com/terms/s/small-cap.asp.

[25] W. U. Din, 'Stock return predictability with financial ratios: Evidence from psx 100 index companies', *Available at SSRN 3077890*, 2017.

[26] J. Maverick, *How small cap stocks differ in risk vs. large cap stocks*, 2022. [Online]. Available: https://www.investopedia.com/ask/answers/032615/how-do-risks-large-cap-stocks-differ-risks-small-cap-stocks.asp.

[27] J. Fundora, *Multiple time frames can multiply returns*, 2022. [Online]. Available: https://www.investopedia.com/articles/trading/07/timeframes.asp.

[28] V. Manahov and A. Urquhart, 'The efficiency of bitcoin: A strongly typed genetic programming approach to smart electronic bitcoin markets', *International Review of Financial Analysis*, vol. 73, p. 101 629, 2021.

[29] S. Chatterjee and T. Adinarayan, *Buy, sell, repeat! no room for 'hold' in whipsawing markets*, 2020. [Online]. Available: https://www.reuters.com/article/us-health-coronavirus-short-termism-anal-idUSKBN24Z0XZ.

[30] M. J. McGowan, 'The rise of computerized high frequency trading: Use and controversy', *Duke L. & Tech. Rev.*, vol. 9, p. 1, 2010.

[31] *Which timeframes are the best for trading?*, 2023. [Online]. Available: https://fbs.com/analytics/tips/how-to-choose-a-timeframe-for-trading-10338.

[32] P. Whittle, *Hypothesis testing in time series analysis*. Almqvist & Wiksells boktr., 1951, vol. 4.

[33] G. Udny Yule, 'On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers', *Philosophical Transactions of the Royal Society of London Series A*, vol. 226, pp. 267–298, 1927.

[34] N. Wiener, N. Wiener, C. Mathematician, N. Wiener, N. Wiener and C. Mathématicien, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press Cambridge, MA, 1949, vol. 113.

[35] G. E. Box, G. M. Jenkins, G. C. Reinsel and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[36] S. Huang, D. Wang, X. Wu and A. Tang, 'Dsanet: Dual self-attention network for multivariate time series forecasting', in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 2129–2132.

[37] P. Newbold, 'Arima model building and the time series analysis approach to forecasting', *Journal of forecasting*, vol. 2, no. 1, pp. 23–35, 1983.

[38] M. Omran and A. Ragab, 'Linear versus non-linear relationships between financial ratios and stock returns: Empirical evidence from egyptian firms', *Review of Accounting and finance*, vol. 3, no. 2, pp. 84–102, 2004.

[39] Q. Cheng, X. Liu and X. Zhu, 'Cryptocurrency momentum effect: Dfa and mf-dfa analysis', *Physica A: Statistical Mechanics and its Applications*, vol. 526, p. 120 847, 2019.

[40] J. J. Hopfield and D. W. Tank, 'Computing with neural circuits: A model', *Science*, vol. 233, no. 4764, pp. 625–633, 1986.

[41] D. E. Rumelhart, G. E. Hinton and R. J. Williams, 'Learning representations by back-propagating errors', *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[42] J. Roman and A. Jameel, 'Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns', in *Proceedings of HICSS-29: 29th Hawaii international conference on system sciences*, IEEE, vol. 2, 1996, pp. 454–460.

[43] P. Tino, C. Schittenkopf and G. Dorffner, 'Financial volatility trading using recurrent neural networks', *IEEE transactions on neural networks*, vol. 12, no. 4, pp. 865–874, 2001.

[44] S. Hochreiter, 'The vanishing gradient problem during learning recurrent neural nets and problem solutions', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[45] S. Hochreiter and J. Schmidhuber, 'Long short-term memory', *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[46] S. Selvin, R Vinayakumar, E. Gopalakrishnan, V. K. Menon and K. Soman, 'Stock price prediction using lstm, rnn and cnn-sliding window model', in *2017 international conference on advances in computing, communications and informatics (icacci)*, IEEE, 2017, pp. 1643–1647.

[47] D. M. Gunarto, S. Sa'adah and D. Q. Utama, 'Predicting cryptocurrency price using rnn and lstm method', *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, pp. 1–8, 2023.

[48] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, 'Empirical evaluation of gated recurrent neural networks on sequence modeling', *arXiv preprint arXiv:1412.3555*, 2014.

[49] P. Dey, E. Hossain, M. I. Hossain, M. A. Chowdhury, M. S. Alam, M. S. Hossain and K. Andersson, 'Comparative analysis of recurrent neural networks in stock price prediction for different frequency domains', *Algorithms*, vol. 14, no. 8, p. 251, 2021.

[50] M. J. Hamayel and A. Y. Owda, 'A novel cryptocurrency price prediction model using gru, lstm and bi-lstm machine learning algorithms', *AI*, vol. 2, no. 4, pp. 477–496, 2021.

[51] Y. Xu, *Bitcoin Price Forecast Using LSTM and GRU Recurrent networks, and Hidden Markov Model*. University of California, Los Angeles, 2020.

[52] P. L. Seabe, C. R. B. Moutsinga and E. Pindza, 'Forecasting cryptocurrency prices using lstm, gru, and bi-directional lstm: A deep learning approach', *Fractal and Fractional*, vol. 7, no. 2, p. 203, 2023.

[53] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, 'Wavenet: A generative model for raw audio', *arXiv preprint arXiv:1609.03499*, 2016.

[54] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. v. d. Oord, A. Graves and K. Kavukcuoglu, 'Neural machine translation in linear time', *arXiv preprint arXiv:1610.10099*, 2016.

[55] Y. N. Dauphin, A. Fan, M. Auli and D. Grangier, 'Language modeling with gated convolutional networks', in *International conference on machine learning*, PMLR, 2017, pp. 933–941.

[56] J. Gehring, M. Auli, D. Grangier, D. Yarats and Y. N. Dauphin, 'Convolutional sequence to sequence learning', in *International conference on machine learning*, PMLR, 2017, pp. 1243–1252.

[57] S. Bai, J. Z. Kolter and V. Koltun, 'An empirical evaluation of generic convolutional and recurrent networks for sequence modeling', *CoRR*, vol. abs/1803.01271, 2018. arXiv: 1803.01271. [Online]. Available: http://arxiv.org/abs/1803.01271.

[58] Y. Liu, H. Dong, X. Wang and S. Han, 'Time series prediction based on temporal convolutional network', in *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, IEEE, 2019, pp. 300–305.

[59] W. Dai, Y. An and W. Long, 'Price change prediction of ultra high frequency financial data based on temporal convolutional network', *Procedia Computer Science*, vol. 199, pp. 1177–1183, 2022.

[60] E. J. C. Lopes and R. A. da Costa Bianchi, 'Short-term prediction for ethereum with deep neural networks', in *Anais do I Brazilian Workshop on Artificial Intelligence in Finance*, SBC, 2022, pp. 1–12.

[61] B. N. Oreshkin, D. Carpov, N. Chapados and Y. Bengio, 'N-beats: Neural basis expansion analysis for interpretable time series forecasting', *arXiv preprint arXiv:1905.10437*, 2019.

[62] S. Makridakis, E. Spiliotis and V. Assimakopoulos, 'The m4 competition: 100,000 time series and 61 forecasting methods', *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020.

[63] A. Bulatov, 'Forecasting bitcoin prices using n-beats deep learning architecture', 2020.

[64] A. El Majzoub, F. A. Rabhi and W. Hussain, 'Evaluating interpretable machine learning predictions for cryptocurrencies', *Intelligent Systems in Accounting, Finance and Management*, 2023.

[65] T. AGGRAWAL and M. Dhawan, 'State-of-the-art vs prominent models: An empirical analysis of various neural networks on stock market prediction',

[66] B. Lim, S. Ö. Arık, N. Loeff and T. Pfister, 'Temporal fusion transformers for interpretable multi-horizon time series forecasting', *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[67] X. Hu, 'Stock price prediction based on temporal fusion transformer', in *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, IEEE, 2021, pp. 60–66.

[68] K. Murray, A. Rossi, D. Carraro and A. Visentin, 'On forecasting cryptocurrency prices: A comparison of machine learning, deep learning, and ensembles', *Forecasting*, vol. 5, no. 1, pp. 196–209, 2023.

[69] T. K. Ho, 'Random decision forests', in *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, vol. 1, 1995, pp. 278–282.

[70] M. Chen, N. Narwal and M. Schultz, 'Predicting price changes in ethereum', *International Journal on Computer Science and Engineering (IJCSE) ISSN*, pp. 0975–3397, 2019.

[71] J. Chen, 'Analysis of bitcoin price prediction using machine learning', *Journal of Risk and Financial Management*, vol. 16, no. 1, p. 51, 2023.

[72] T. Chen and C. Guestrin, 'Xgboost: A scalable tree boosting system', *CoRR*, vol. abs/1603.02754, 2016. arXiv: 1603.02754. [Online]. Available: http://arxiv.org/abs/1603.02754.

[73] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, 'Lightgbm: A highly efficient gradient boosting decision tree', *Advances in neural information processing systems*, vol. 30, 2017.

[74] S. B. Jabeur, S. Mefteh-Wali and J.-L. Viviani, 'Forecasting gold price with the xgboost algorithm and shap interaction values', *Annals of Operations Research*, pp. 1–21, 2021.

[75] J. Wu, X. Guo, M. Fang and J. Zhang, 'Short term return prediction of cryptocurrency based on xgboost algorithm', in *2022 International Conference on Big Data, Information and Computer Network (BDICN)*, IEEE, 2022, pp. 39–42.

[76] P. C. Sekhar, M Padmaja, B. Sarangi *et al.*, 'Prediction of cryptocurrency using lstm and xgboost', in *2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, IEEE, 2022, pp. 1–5.

[77] Y. Yang, Y. Wu, P. Wang and X. Jiali, 'Stock price prediction based on xgboost and lightgbm', in *E3S Web of Conferences*, EDP Sciences, vol. 275, 2021, p. 01 040.

[78] X. Sun, M. Liu and Z. Sima, 'A novel cryptocurrency price trend forecasting model based on lightgbm', *Finance Research Letters*, vol. 32, p. 101 084, 2020.

[79] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza, M. Mergenthaler and A. Dubrawski, 'N-hits: Neural hierarchical interpolation for time series forecasting', *arXiv preprint arXiv:2201.12886*, 2022.

[80] A. M. De Livera, R. J. Hyndman and R. D. Snyder, 'Forecasting time series with complex seasonal patterns using exponential smoothing', *Journal of the American statistical association*, vol. 106, no. 496, pp. 1513–1527, 2011.

[81] I. Sadia, A. Mahmood, L. B. M. Kiah and S. R. Azzuhri, 'Analysis and forecasting of blockchain-based cryptocurrencies and performance evaluation of tbats, nnar and arima', in *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, IEEE, 2022, pp. 1–6.

[82] S. J. Taylor and B. Letham, 'Forecasting at scale', *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[83] N. Kumar, R. Chauhan and G. Dubey, 'Forecasting of stock price using lstm and prophet algorithm', in *Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2020*, Springer, 2021, pp. 141–155.

[84] S. Kaninde, M. Mahajan, A. Janghale and B. Joshi, 'Stock price prediction using facebook prophet', in *ITM Web of Conferences*, EDP Sciences, vol. 44, 2022, p. 03 060.

[85] I. Yenidoğan, A. Çayir, O. Kozan, T. Dağ and Ç. Arslan, 'Bitcoin forecasting using arima and prophet', in *2018 3rd international conference on computer science and engineering (UBMK)*, IEEE, 2018, pp. 621–624.

[86] R. G. Tiwari, A. K. Agarwal, R. K. Kaushal and N. Kumar, 'Prophetic analysis of bitcoin price using machine learning approaches', in *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, IEEE, 2021, pp. 428–432.

[87] A. Chaudhari, 'Forecasting cryptocurrency prices using machine learning', Ph.D. dissertation, Dublin, National College of Ireland, 2020.

[88] W. Marquering and M. Verbeek, 'The economic value of predicting stock index returns and volatility', *Journal of Financial and Quantitative Analysis*, vol. 39, no. 2, pp. 407–429, 2004.

[89] INFINOX, *What are time frames in trading?*, 2022. [Online]. Available: https://www.infinox.com/fsc/en/ix-intel/what-is-time-frame-analysis.

[90] CoinMarketCap, *Top cryptocurrency exchanges ranked by volume*, 2023. [Online]. Available: https://coinmarketcap.com/rankings/exchanges/.

[91] CoinMarketCap, *Top stablecoin tokens by market capitalization*, 2023. [Online]. Available: https://coinmarketcap.com/view/stablecoin/.

[92] D. Kwiatkowski, P. C. Phillips, P. Schmidt and Y. Shin, 'Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?', *Journal of econometrics*, vol. 54, no. 1-3, pp. 159–178, 1992.

[93] D. A. Dickey and W. A. Fuller, 'Distribution of the estimators for autoregressive time series with a unit root', *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.

[94] T. Smith, *Autocorrelation: What it is, how it works, tests*, 2023. [Online]. Available: https://www.investopedia.com/terms/a/autocorrelation.asp.

[95] A. Dotis-Georgiou, *Autocorrelation in time series data: What is autocorrelation?*, 2022. [Online]. Available: https://www.influxdata.com/blog/autocorrelation-in-time-series-data.

[96] J. H. F. Flores, P. M. Engel and R. C. Pinto, 'Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting', in *The 2012 International joint conference on neural networks (IJCNN)*, IEEE, 2012, pp. 1–8.

[97] G. G. Booth and G. Koutmos, 'Volatility and autocorrelation in major european stock markets', *The European Journal of Finance*, vol. 4, no. 1, pp. 61–74, 1998.

[98] W. C. Wei, 'Liquidity and market efficiency in cryptocurrencies', *Economics Letters*, vol. 168, pp. 21–24, 2018.

[99] J. Durbin and G. S. Watson, *Testing for serial correlation in least squares regression. II*. Springer, 1992.

[100] G. M. Ljung and G. E. Box, 'On a measure of lack of fit in time series models', *Biometrika*, vol. 65, no. 2, pp. 297–303, 1978.

[101] L. G. Godfrey, 'Testing for higher order serial correlation in regression equations when the regressors include lagged dependent variables', *Econometrica: Journal of the Econometric Society*, pp. 1303–1310, 1978.

[102] L. Pichl and T. Kaizoji, 'Volatility analysis of bitcoin', *Quantitative Finance and Economics*, vol. 1, no. 4, pp. 474–485, 2017.

[103] C. Brooks, *Introductory econometrics for finance*. Cambridge university press, 2019.

[104] T. Kim and H. Y. Kim, 'Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data', *PloS one*, vol. 14, no. 2, e0212320, 2019.

[105] L. Pines, *Do volatility indicators do what their name suggests? an expert explains*, 2022. [Online]. Available: https://commodity.com/technical-analysis/volatility/.

[106] G. P. Zhang and M. Qi, 'Neural network forecasting for seasonal and trend time series', *European journal of operational research*, vol. 160, no. 2, pp. 501–514, 2005.

[107] K. H. Hamed and A. R. Rao, 'A modified mann-kendall trend test for autocorrelated data', *Journal of hydrology*, vol. 204, no. 1-4, pp. 182–196, 1998.

[108] S. Yue and C. Y. Wang, 'Regional streamflow trend detection with consideration of both temporal and spatial correlation', *International Journal of Climatology: A Journal of the Royal Meteorological Society*, vol. 22, no. 8, pp. 933–946, 2002.

[109] S. Yue and C. Wang, 'The mann-kendall test modified by effective sample size to detect trend in serially correlated hydrological series', *Water resources management*, vol. 18, no. 3, pp. 201–218, 2004.

[110] S. Razavi and R. Vogel, 'Prewhitening of hydroclimatic time series? implications for inferred change and variability across time scales', *Journal of hydrology*, vol. 557, pp. 109–115, 2018.

[111] J. Brownlee, *How to identify and remove seasonality from time series data with python*, 2020. [Online]. Available: https://machinelearningmastery.com/time-series-seasonality-with-python/.

[112] F. J. Seyyed, A. Abraham and M. Al-Hajji, 'Seasonality in stock returns and volatility: The ramadan effect', *Research in International Business and Finance*, vol. 19, no. 3, pp. 374–383, 2005.

[113] R. B. Cleveland, W. S. Cleveland, J. E. McRae and I. Terpenning, 'Stl: A seasonal-trend decomposition', *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[114] L. Kaiser, 'Seasonality in cryptocurrencies', *Finance Research Letters*, vol. 31, 2019.

[115] A. Hayes, *Volatility: Meaning in finance and how it works with stocks*, 2023. [Online]. Available: https://www.investopedia.com/terms/v/volatility.asp.

[116] J. M. Wooldridge, *Introductory econometrics: A modern approach*. Cengage learning, 2015.

[117] A. Corhay and A. T. Rad, 'Conditional heteroskedasticity adjusted market model and an event study', *The Quarterly Review of Economics and Finance*, vol. 36, no. 4, pp. 529–538, 1996.

[118] M. J. Fleming and E. M. Remolona, 'What moves the bond market?', *Economic policy review*, vol. 3, no. 4, 1997.

[119] T. S. Breusch and A. R. Pagan, 'A simple test for heteroscedasticity and random coefficient variation', *Econometrica: Journal of the econometric society*, pp. 1287–1294, 1979.

[120] S. M. Goldfeld and R. E. Quandt, 'Some tests for homoscedasticity', *Journal of the American statistical Association*, vol. 60, no. 310, pp. 539–547, 1965.

[121] A. Phillip, J. S. Chan and S. Peiris, 'A new look at cryptocurrencies', *Economics Letters*, vol. 163, pp. 6–9, 2018.

[122] R. F. Engle, 'Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation', *Econometrica: Journal of the econometric society*, pp. 987–1007, 1982.

[123] B. B. Mandelbrot and J. W. Van Ness, 'Fractional brownian motions, fractional noises and applications', *SIAM review*, vol. 10, no. 4, pp. 422–437, 1968.

[124] B. Qian and K. Rasheed, 'Hurst exponent and financial market predictability', in *IASTED conference on Financial Engineering and Applications*, Proceedings of the IASTED International Conference Cambridge, MA, 2004, pp. 203–209.

[125] S. K. Mitra, 'Is hurst exponent value useful in forecasting financial time series', *Asian social science*, vol. 8, no. 8, pp. 111–120, 2012.

[126] H. E. Hurst, 'Long-term storage capacity of reservoirs', *Transactions of the American society of civil engineers*, vol. 116, no. 1, pp. 770–799, 1951.

[127] R. Baronas, 'Fractal analysis of time series of the cryptocurrencies price', Ph.D. dissertation, Vilniaus universitetas, 2022.

[128] S. A. David, C. Inacio Jr, R. Nunes and J. T. Machado, 'Fractional and fractal processes applied to cryptocurrencies price series', *Journal of Advanced Research*, vol. 32, pp. 85–98, 2021.

[129] D. Kasper *et al.*, 'Evolution of bitcoin-volatility comparisons with least developed countries' currencies', *Evolution of Bitcoin-Volatility Comparisons with Least Developed Countries' Currencies (October 13, 2017)*, 2017.

[130] J. Baffes and V. Kshirsagar, 'Sources of volatility during four oil price crashes', *Applied Economics Letters*, vol. 23, no. 6, pp. 402–406, 2016.

[131] M. B. Billings, R. Jennings and B. Lev, 'On guidance and volatility', *Journal of Accounting and Economics*, vol. 60, no. 2-3, pp. 161–180, 2015.

[132] P. Giot, 'Relationships between implied volatility indices and stock index returns', *Journal of Portfolio Management*, vol. 31, no. 3, pp. 92–100, 2005.

[133] F. Pedregosa *et al.*, 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[134] Q.-Q. He, P. C.-I. Pang and Y.-W. Si, 'Multi-source transfer learning with ensemble for financial time series forecasting', in *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, IEEE, 2020, pp. 227–233.

[135] J. Herzen *et al.*, 'Darts: User-friendly modern machine learning for time series', *CoRR*, vol. abs/2110.03224, 2021. arXiv: 2110.03224. [Online]. Available: https://arxiv.org/abs/2110.03224.

[136] A. Borovykh, S. Bohte and C. W. Oosterlee, 'Conditional time series forecasting with convolutional neural networks', *arXiv preprint arXiv:1703.04691*, 2017.

[137] R. Chowdhury, M. A. Rahman, M. S. Rahman and M. Mahdy, 'An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning', *Physica A: Statistical Mechanics and its Applications*, vol. 551, p. 124 569, 2020.

[138] S. K. Nayak, S. C. Nayak and S. Das, 'Modeling and forecasting cryptocurrency closing prices with rao algorithm-based artificial neural networks: A machine learning approach', *FinTech*, vol. 1, no. 1, pp. 47–62, 2021.

[139] *Tune: Scalable hyperparameter tuning*. [Online]. Available: https://docs.ray.io/en/latest/tune/index.html.

[140] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht and A. Talwalkar, *A system for massively parallel hyperparameter tuning*, 2020. arXiv: 1810.05934 [cs.LG].

[141] A. I. Cowen-Rivers, W. Lyu, R. Tutunov, Z. Wang, A. Grosnit, R. R. Griffiths, A. M. Maraval, H. Jianye, J. Wang, J. Peters *et al.*, 'Hebo: Pushing the limits of sample-efficient hyper-parameter optimisation', *Journal of Artificial Intelligence Research*, vol. 74, pp. 1269–1349, 2022.

[142] B. N. Oreshkin, G. Dudek, P. Pełka and E. Turkina, 'N-beats neural network for mid-term electricity load forecasting', *Applied Energy*, vol. 293, p. 116 918, 2021.

[143] F. Garza, M. M. Canseco, C. Challú and K. G. Olivares, *StatsForecast: Lightning fast forecasting with statistical and econometric models*, PyCon Salt Lake City, Utah, US 2022, 2022. [Online]. Available: https://github.com/Nixtla/statsforecast.

[144] P. Ghosh, A. Neufeld and J. K. Sahoo, 'Forecasting directional movements of stock prices for intraday trading using lstm and random forests', *Finance Research Letters*, vol. 46, p. 102 280, 2022.

[145] Y. Wang and X. S. Ni, 'A xgboost risk model via feature selection and bayesian hyper-parameter optimization', *arXiv preprint arXiv:1901.08433*, 2019.

[146] D.-n. Wang, L. Li and D. Zhao, 'Corporate finance risk prediction based on lightgbm', *Information Sciences*, vol. 602, pp. 259–268, 2022.

[147] Z. Xiaosong and Z. Qiangfu, 'Stock prediction using optimized lightgbm based on cost awareness', in *2021 5th IEEE International Conference on Cybernetics (CYBCONF)*, IEEE, 2021, pp. 107–113.

[148] Facebook, *Diagnostics*, 2023. [Online]. Available: https://facebook.github.io/prophet/docs/diagnostics.html.

[149] A. González Mata, 'A comparison between lstm and facebook prophet models: A financial forecasting case study', 2020.

[150] Y. Ning, H. Kazemi and P. Tahmasebi, 'A comparative machine learning study for time series oil production forecasting: Arima, lstm, and prophet', *Computers & Geosciences*, vol. 164, p. 105 126, 2022.

[151] T. Hollis, A. Viscardi and S. E. Yi, 'A comparison of lstms and attention mechanisms for forecasting financial time series', *arXiv preprint arXiv:1812.07699*, 2018.

[152] N. Buslim, I. L. Rahmatullah, B. A. Setyawan and A. Alamsyah, 'Comparing bitcoin's prediction model using gru, rnn, and lstm by hyperparameter optimization grid search and random search', in *2021 9th International Conference on Cyber and IT Service Management (CITSM)*, IEEE, 2021, pp. 1–6.

[153] X. Luo, W. Gan, L. Wang, Y. Chen and E. Ma, 'A deep learning prediction model for structural deformation based on temporal convolutional networks', *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–12, 2021.

[154] B. Lei, B. Zhang and Y. Song, 'Volatility forecasting for high-frequency financial data based on web search index and deep learning model', *Mathematics*, vol. 9, no. 4, p. 320, 2021.

[155] E. J. C. Lopes and R. A. da Costa Bianchi, 'Short-term prediction for ethereum with deep neural networks and statistical validation tests', in *Anais do II Brazilian Workshop on Artificial Intelligence in Finance*, SBC, 2023, pp. 1–12.

[156] J. Sappl, M. Harders and W. Rauch, 'Machine learning for quantile regression of biogas production rates in anaerobic digesters', *Science of The Total Environment*, vol. 872, p. 161 923, 2023.

[157] S. Bouktif, A. Fiaz, A. Ouni and M. A. Serhani, 'Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches', *Energies*, vol. 11, no. 7, p. 1636, 2018.

[158] N. I. Sapankevych and R. Sankar, 'Time series prediction using support vector machines: A survey', *IEEE computational intelligence magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[159] A. Mikhaylov, M. S. S. Danish and T. Senjyu, 'A new stage in the evolution of cryptocurrency markets: Analysis by hurst method', in *Strategic outlook in business and finance innovation: Multidimensional policies for emerging economies*, Emerald Publishing Limited, 2021, pp. 35–45.

[160] H. Lyu, 'Cryptocurrency price forecasting: A comparative study of machine learning model in short-term trading', in *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, IEEE, 2022, pp. 280–288.

[161] I. E. Livieris, N. Kiriakidou, S. Stavroyiannis and P. Pintelas, 'An advanced cnn-lstm model for cryptocurrency forecasting', *Electronics*, vol. 10, no. 3, p. 287, 2021.

[162] S. Sharma and A. Majumdar, 'Deep state space model for predicting cryptocurrency price', *Information Sciences*, vol. 618, pp. 417–433, 2022.

[163] A. Chokor and E. Alfieri, 'Long and short-term impacts of regulation in the cryptocurrency market', *The Quarterly Review of Economics and Finance*, vol. 81, pp. 157–173, 2021.

[164] E. Akyildirim, A. Goncu and A. Sensoy, 'Prediction of cryptocurrency returns using machine learning', *Annals of Operations Research*, vol. 297, pp. 3–36, 2021.