

# Forecasting Cryptocurrency Prices Using Deep Learning: Integrating Financial, Blockchain, and Text Data

## Master Thesis

for acquiring the degree of  
Master of Science (M.Sc.)

in Business Economics  
at the Faculty of Economics  
of the Humboldt University Berlin

Submitted by:

Vincent B. Gurgul  
Matriculation no.: 573476

First examiner: Prof. Dr. Stefan Lessmann  
Second examiner: Prof. Dr. Benjamin Fabian  
Chair of Information Systems

Berlin, September 14, 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Literature Review</b>	<b>2</b>
2.1	Natural Language Processing (NLP) in the Context of Cryptocurrencies . . . . .	2
2.2	Target Variable Selection in Financial Forecasting . . . . .	3
2.3	Comparison of Cryptocurrency Price Forecasting Approaches . . . . .	4
<b>3</b>	<b>Methodology</b>	<b>7</b>
3.1	Deep Learning Approaches for Natural Language Processing (NLP) . . . . .	7
3.2	Choice of Target Variables and Trading Strategy . . . . .	9
3.3	Time Series Modelling with Neural Networks . . . . .	11
<b>4</b>	<b>Experimental Design</b>	<b>16</b>
4.1	Data Collection and Preprocessing . . . . .	16
4.2	Time Series Forecasting Models . . . . .	18
4.3	Performance Metrics and Model Evaluation . . . . .	19
<b>5</b>	<b>Results and Analysis</b>	<b>21</b>
5.1	Comparison of Forecasting Performance . . . . .	21
5.2	Analysis of Feature Importance . . . . .	26
5.3	Market Efficiency Throughout Time . . . . .	29
<b>6</b>	<b>Conclusion</b>	<b>31</b>
6.1	Summary of Findings . . . . .	31
6.2	Implications . . . . .	32
<b>References</b>		<b>34</b>
<b>Appendix A: Disaggregated Results</b>		<b>39</b>
<b>Appendix B: Documentation</b>		<b>45</b>

# List of Figures

3.1	Sample of local extreme points of the Bitcoin (BTC) price . . . . .	10
3.2	The standard Recurrent Neural Network (RNN) model architecture . . . . .	12
3.3	Long Short-Term Memory (LSTM) cell architecture . . . . .	13
3.4	Comparison of information decay in Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks . . . . .	13
3.5	Convolutional Neural Network (CNN) time series model architecture . . . . .	14
3.6	Temporal Fusion Transformer (TFT) model architecture . . . . .	15
4.1	Overview of the experimental design . . . . .	16
4.2	The applied time series cross-validation approach . . . . .	20
5.1	Comparison of Multi-Layer Perceptron (MLP) profit and AUC ROC by dataset . .	22
5.2	Comparison of Multi-Layer Perceptron (MLP) profit and AUC ROC by target variable . . . . .	23
5.3	Comparison of Multi-Layer Perceptrons (MLPs) trained on daily price change and $\pm 7$ day extreme points . . . . .	24
5.4	Distribution of trading profits of the 10 most profitable Multi-Layer Perceptron (MLP) models throughout time . . . . .	30

# List of Tables

2.1	Overview of cryptocurrency price forecasting approaches . . . . .	6
3.1	Highest-scoring r/Bitcoin subreddit posts by NLP model . . . . .	9
5.1	Reference points for the analysis of trading profit . . . . .	21
5.2	Feature categories ranked by their importance for predicting Bitcoin (BTC) price movements . . . . .	27
5.3	Feature categories ranked by their importance for predicting Ethereum (ETH) price movements . . . . .	28
A.1	Average performance by time series model using all NLP features . . . . .	39
A.2	The ten most profitable Bitcoin price movement regression models . . . . .	39
A.3	The ten most profitable Ethereum price movement regression models . . . . .	39
A.4	The ten most profitable Bitcoin price movement classification models . . . . .	40
A.5	The ten most profitable Ethereum price movement classification models . . . . .	40
A.6	The ten most profitable Bitcoin $+/- 7$ day extrema classification models . . . . .	40
A.7	The ten most profitable Ethereum $+/- 7$ day extrema classification models . . . . .	41
A.8	The ten most profitable Bitcoin $+/- 14$ day extrema classification models . . . . .	41
A.9	The ten most profitable Ethereum $+/- 14$ day extrema classification models . . . . .	41
A.10	The ten most profitable Bitcoin $+/- 21$ day extrema classification models . . . . .	42
A.11	The ten most profitable Ethereum $+/- 21$ day extrema classification models . . . . .	42
A.12	The 50 most influential features for predicting Bitcoin (BTC) price movements using XGBoost . . . . .	43
A.13	The 50 most influential features for predicting Ethereum (ETH) price movements using XGBoost . . . . .	44
B.1	Search ranges of the hyperparameter optimisation for the RoBERTa fine-tuning . .	45
B.2	Search ranges of the hyperparameter optimisation for the time series models . . . .	46
B.3	Overview and description of the Bitcoin/Ethereum features . . . . .	47

# List of Abbreviations

<b>AI</b>	Artificial Intelligence
<b>AUC ROC</b>	Area Under the Receiver Operating Characteristic Curve
<b>BCE</b>	Binary Cross-Entropy
<b>BTC</b>	Bitcoin
<b>CNN</b>	Convolutional Neural Network
<b>ETH</b>	Ethereum
<b>FNN</b>	Feed-Forward Neural Network
<b>GRN</b>	Gated Residual Network
<b>GRU</b>	Gated Recurrent Unit
<b>LLM</b>	Large Language Model
<b>LSTM</b>	Long Short-Term Memory
<b>ML</b>	Machine Learning
<b>MLP</b>	Multi-Layer Perceptron
<b>MSE</b>	Mean Squared Error
<b>NLP</b>	Natural Language Processing
<b>OLS</b>	Ordinary Least Squares
<b>RNN</b>	Recurrent Neural Network
<b>SVM</b>	Support Vector Machine
<b>TFT</b>	Temporal Fusion Transformer

# 1 Introduction

The cryptocurrency market has emerged as an alternative financial system over the last decade, attracting substantial attention from researchers and practitioners. This unique decentralized market is characterised by high volatility and ample data availability, making it a compelling field for the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques. In particular, the availability of vast public sentiment data, primarily from social networks, opens up new avenues for the integration of Natural Language Processing (NLP) into the forecasting of cryptocurrency price movements.

The research presented in this work assesses the influence of news from various cryptocurrency related outlets and social media posts from Twitter and Reddit on the valuations of the two largest cryptocurrencies by market capitalisation, Bitcoin (BTC) and Ethereum (ETH). Thus far, the majority of papers in the cryptocurrency domain apply only dictionary-based approaches to analyse the potential effects of news and social media (see Section 2.1 “NLP in the Context of Cryptocurrencies”). We do justice to the complexity of human language by applying an array of deep learning NLP techniques instead. In this scope, we introduce a novel methodology for converting the textual data into numerical representations of market sentiment.

Leung, Daouk, and Chen (2000) demonstrate that treating price forecasting as a classification rather than a regression problem results in higher accuracy and, notably, higher profit. In the realm of cryptocurrency price analysis, this is usually the binary classification of daily price movement (up or down). However, some studies successfully apply the classification of changepoints for stock market prediction (Ferraz and Moura, 2017; Liang, Wang, and Wu, 2022; Yoo et al., 2021). Changepoints, such as local extrema, are fundamentally less noisy than daily price movements. We hypothesise that by classifying local minima and maxima our ML models obtain enhanced prediction performance, both with respect to classification metrics and trading profit. In our analysis of the impact of market sentiment we therefore aim to establish whether the textual data can aid in the prediction of local extreme points with various observational time frames, in addition to the forecasting of daily price movements. This also enables the examination of the longevity of the effects of sentiments disseminated through social media.

We compare the predictive performance of various ML methods – including deep learning and sequential models – using different approaches for quantifying market sentiment, to models that do not include textual data. This comparison is conducted across five different target variables. Our evaluation is not solely reliant on traditional metrics for assessing binary classification performance, but also incorporates a trading simulation to determine the profits the models could potentially generate.

We further expand upon the existing body of literature by examining the evolution of market efficiency throughout time, especially with respect to the growing prevalence of textual data. To

ensure a comprehensive analysis, we leverage the entire available time frame and incorporate a wide range of publicly available information, ranging from on-chain transaction data to GitHub and Google Trends.

The paper is structured in the following way: In Section 2, we review the use of NLP, the selection of target variables and the choice of forecasting methods in the existing literature. In Section 3, we give a general introduction to the methodology that we apply. In Section 4, the experimental design is outlined. This includes the data processing, the time series models and our evaluation approach. In Section 5, we present and interpret the performance of the models. Finally, conclusions and implications are found in Section 6.

## 2 Literature Review

### 2.1 NLP in the Context of Cryptocurrencies

In the landscape of cryptocurrency price forecasting, a predominant trend is the application of dictionary-based sentiment analysis methodologies, where each word is assigned a score from a hand-labelled dictionary without considering its context or potential ambiguities. Among the commonly employed dictionaries, the open-source library VADER stands out as a frequent choice (Abraham, Higdon, Nelson, Ibarra, and Nelson, 2018; Mittal, Dhiman, Singh, and Prakash, 2019; Parekh et al., 2022; Valencia, Gómez-Espinosa, and Valdés-Aguirre, 2019; Wołk, 2019), with the library Textblob being the second most used tool (Jain, Tripathi, Dwivedi, and Saxena, 2018; Li, Chamrajnagar, Fong, Rizik, and Fu, 2019; Raju and Tarif, 2020). Karalevicius, Degrande, and De Weerdt (2018) differentiate themselves by comparing the sentiment scores obtained using the Harvard Psychosocial Dictionary and the Loughran-McDonald finance-specific dictionary, which not only includes a dictionary of sentiment polarity but also dictionaries of “negators” and “intensifiers”, allowing for a more nuanced analysis. Negators reverse the sentiment of words, while intensifiers amplify or weaken sentiment.

Such dictionary-based methods have demonstrated their merits in the earlier literature; however, they inherently rely on pre-defined lexicons, failing to capture any long-term dependency and therefore omitting contextual nuances and relationships. Only a handful of papers have integrated neural networks to harness textual data for predictions. One study leveraged pre-trained embeddings (Gensim Word2Vec) alongside a bag-of-words framework for encoding textual information, subsequently inputting this data into a Recurrent Neural Network (RNN) for forecasting purposes (Pant, Neupane, Poudel, Pokhrel, and Lama, 2018). In addition, Long Short-Term Memory (LSTM) networks, a specialised variant of RNNs, have been applied for sentiment analysis in the cryptocurrency price forecasting domain by Inamdar, Bhagtani, Bhatt, and Shetty (2019).

While these efforts signify a shift towards more sophisticated modelling, it is surprising that only three studies have explored the potential of state-of-the-art language models based on the Transformer model (Vaswani et al., 2017) for cryptocurrency price forecasting. Unlike dictionary-based methodologies, which fall short in grasping the dynamic nuances of language, Transformers have the capability to capture complex contextual relationships. Moreover, when compared to embeddings and traditional RNNs, Transformers manifest a superior ability in capturing long-range dependencies in text, making them more proficient at understanding intricate grammatical and logical patterns. Ortú, Uras, Conversano, Bartolucci, and Destefanis (2022) employ a pre-trained BERT model to categorise emotions and sentiments in comments about cryptocurrencies on GitHub and Reddit. In addition to sentiment polarity, these comments are also assessed for specific emotional reactions, such as joy, anger, and sadness. This data is used to predict hourly and daily price increases/decreases. Ider and Lessmann (2022) make use of BERT and FinBERT models for the analysis of the sentiment of cryptocurrency-related news and social media posts. This research also introduces an innovative dimension by fine-tuning these pre-trained models using a weak-labelling approach. The third paper, Kim, Kim, Kim, and Lim (2023), fine-tunes BERT, DeBERTa, and RoBERTa models on a manually labeled cryptocurrency-specific corpus of news articles, compares their performance to pre-trained mBERT and XLM-RoBERTa models, and subsequently builds a trading bot based on those results.

It is worth highlighting that outside the narrow boundary of price forecasting, there exists a broader literature on the application of advanced sentiment analysis techniques in the realm of cryptocurrencies (Dwivedi, 2023; Raheman, Kolonin, Fridkins, Ansari, and Vishwas, 2022; Widianto and Cornelius, 2023). Yet, their potential for improving forecasting performance remains relatively uncharted. In this context, we delve deeper into the utilisation of deep learning in cryptocurrency forecasting by harnessing the potential of Transformer models and their fine-tuning. Further, we introduce a novel approach for converting textual data into numerical representations of market sentiment by constructing a Transformer-based model that predicts whether a piece of text expresses optimism with regards to the price development of a cryptocurrency or not (see Section 3.1 “Deep Learning Approaches for NLP”).

## 2.2 Target Variable Selection in Financial Forecasting

The challenge of forecasting asset prices has long been at the centre of academic interest (Fama, 1970). Traditionally, the predominant approach in stock price forecasting literature involves regressing onto the subsequent day’s price or its relative change. Moreover, several research endeavours explore the possibility of binary classification, wherein the goal is to predict whether the price will experience an increase or decrease in the forthcoming period (see Obthong, Tantisantiwong, Teamwatthanachai, and Wills, 2020). These trends persist within the cryptocurrency domain (see overview in Table 2.1).

Leung et al. (2000) show that approaching stock price analysis as a classification rather than a regression problem results in superior forecasting performance, both in terms of classification metrics as well as trading profit. Several studies document similar findings in the realm of cryptocurrency price forecasting (Ider and Lessmann, 2022; Mudassir, Bennbaia, Unal, and Hammoudeh, 2020). This suggests that a more straightforward, discrete target is easier for ML models to associate with the explanatory variables.

Some studies successfully apply the classification of changepoints for stock market prediction (Ferraz and Moura, 2017; Liang et al., 2022; Yoo et al., 2021). However, this remains an unexplored area within the cryptocurrency realm. The only notable effort in this direction is the detection of structural breaks in the cryptocurrency price movements (Kim, Shin, Choi, and Lim, 2022; Thies and Molnár, 2018). Broadening the discourse, we introduce an approach encompassing changepoint analysis in tandem with daily log price changes and daily price movements, represented as binary variables, allowing us to analyse and compare the efficacies of these different approaches. While changepoints could provide a clearer, less noisy signal for prediction, the daily metrics contain more information through capturing the granular shifts in the market.

For the changepoint analysis, we delve into an approach centred on local extrema. We construct two binary variables that indicate whether a given timepoint is a local minimum or maximum respectively. These variables become the target for two distinct binary classification models. The forecasts of these two models are then used to construct a trading strategy, which aims at purchasing the asset at the troughs and selling it at the peaks.

By virtue of conveying less granular information compared to daily price fluctuations, local extrema as target variables imply a ceiling on potential profits. However, it is essential to reconceptualise this perspective. Given that local extrema are less susceptible to noise in comparison to daily price changes, a plausible argument can be made that they present a more stable and distinct pattern for machine learning approaches to model. Thus, even though these extrema-based models might be associated with lower potential profit, their heightened accuracy could translate into greater profitability in practice. This contrasts with models trained on daily variations which, while encapsulating more information, might be impeded by their inherent noise and variability, leading to less efficient predictions that only generate a fraction of the potential profit. Furthermore, the prediction of local extreme points allows us to examine the impact of textual data across varying observational time frames, giving insight into the longevity of the effects of sentiments propagated through news and social media.

## 2.3 Comparison of Cryptocurrency Price Forecasting Approaches

In the realm of cryptocurrency price forecasting, the status quo in time series modelling predominantly revolves around a mix of non-sequential linear and non-linear models. Ordinary

Least Squares (OLS), Support Vector Machines (SVMs), and ARIMA, are extensively explored. At the same time, tree-based models like Random Forests and XGBoost also garner plenty of attention. However, it is the neural networks that consistently demonstrate superior performance. In this context, we see the application of simple Multi-Layer Perceptrons (MLPs) as well as Convolutional Neural Networks (CNNs) and RNNs, particularly LSTM networks, Gated Recurrent Unit (GRU) networks, and ensembles thereof (a more comprehensive discussion of the various neural network architectures for time series forecasting is found in Section 3.3 “Time Series Modelling with Neural Networks”).

In some cases the MLP is reported to have achieved the best forecasting performance (Ortu et al., 2022), in some cases it is the sequential models (Murray, Rossi, Carraro, and Visentin, 2023) and in some cases they performed on par (Mudassir et al., 2020). Only a single paper ventures into the application of the Temporal Fusion Transformer (TFT), however, with rather disappointing results (Murray et al., 2023). This underscores the nascent stage of the TFT in the area of cryptocurrency forecasting, despite it showing promising capabilities in other domains.

Turning our attention to the explanatory variables employed in these forecasting models, we observe several trends. A significant portion of the literature traditionally relies on financial data, often resorting to technical indicators or purely autoregressive analyses (Bakar, Rosbi, and Uzaki, 2020; Derbentsev, Matviychuk, and Soloviev, 2020; Livieris, Kiriakidou, Stavroyiannis, and Pintelas, 2021; Pintelas, Livieris, Stavroyiannis, Kotsilieris, and Pintelas, 2020; Rafi et al., 2023). While the majority of studies incorporate text data in some way, it is noteworthy that only a select few venture into the realm of blockchain data (Ider and Lessmann, 2022; Jay et al., 2020; Kim et al., 2022; Mittal et al., 2019; Mudassir et al., 2020; Wołk, 2019). This includes valuable information sources such as transaction data and technical data intrinsic to the blockchain. Additionally, some approaches harness the power of Google Trends to gauge public interest (Abraham et al., 2018; Jay et al., 2020; Mittal et al., 2019; Wołk, 2019). One study utilises data from the GitHub repositories of the cryptocurrencies it aims to forecast, offering a unique perspective on developer activity (Ortu et al., 2022). However, no existing study integrates the entire ensemble of text, financial data, blockchain data, Google Trends, and GitHub data, leaving a potential gap in holistic analysis. An extensive overview of the cryptocurrency price forecasting methodology in the existing literature is found in Table 2.1 on the next page.

In light of the aforementioned observations, our main contribution to the existing literature is the further exploration of the TFT for cryptocurrency forecasting, particularly with the incorporation of binary classification problems. The TFT offers a novel approach to time series forecasting, potentially outperforming the benchmark set by LSTMs in capturing complex temporal dynamics. Nevertheless, its efficacy in cryptocurrency price forecasting has not been particularly impressive thus far. This discrepancy makes it an intriguing domain for further exploration. Moreover, we seek to ensure a comprehensive analysis by using an extensive set of explanatory variables, including transaction data, account balance data, metrics of developer activity, and online search trends.

Table 2.1: Overview of cryptocurrency price forecasting approaches

Paper	Forecasting Methods			NLP Methods			Targets	Text sources	Features				Cryptocurrencies*
	Linear Models	Non-linear Models	Sequential Models	Dictionary	Embeddings	RNN	Transformers		Blockchain	Financial	GitHub	Google Trends	
Colianni et al. (2015)	✓	✓		✓			Daily binary	Twitter					b
Abraham et al. (2018)	✓			✓			Daily price	Twitter			✓		be
Jain et al. (2018)	✓			✓			2-hourly price	Twitter					bl
Karalevicius et al. (2018)				✓			Daily binary	News					b
Pant et al. (2018)		✓		✓			Daily price	Twitter					b
Inamdar et al. (2019)		✓			✓		Daily price	News, Twitter					b
Li et al. (2019)		✓		✓			Daily price	Twitter		✓			z
Mittal et al. (2019)	✓	✓	✓	✓			Daily price	Twitter	✓		✓		b
Valencia et al. (2019)	✓	✓		✓			Daily binary	Twitter		✓			berl
Wojtk (2019)	✓	✓		✓			Daily price	Twitter	✓	✓		✓	bermnc
Bakar et al. (2020)	✓						Daily price						r
Derbentsev et al. (2020)		✓					Daily price, daily log price						ber
Jay et al. (2020)		✓	✓				Daily price		✓	✓		✓	bel
Mudassir et al. (2020)		✓	✓				Daily binary, daily price		✓	✓			b
Pintelas et al. (2020)	✓	✓	✓				Daily price						ber
Raju and Tarif (2020)	✓		✓	✓			Daily price	Twitter		✓			b
Livieris et al. (2021)			✓				Daily relative price change						ber
Ider and Lessmann (2022)	✓	✓				✓	Daily relative price change	News, Reddit, Twitter	✓	✓			be
Kim et al. (2022)			✓				Volatility-adjusted daily price		✓	✓			bt
Ortu et al. (2022)		✓	✓			✓	Daily binary, hourly binary	Reddit, GitHub		✓	✓		be
Parekh et al. (2022)			✓	✓			Daily price	Twitter					bld
Kim et al. (2023)		✓	✓			✓	4-hourly binary	News		✓			b
Murray et al. (2023)	✓	✓	✓				Daily absolute price change			✓			berlm
Rafi et al. (2023)			✓				Daily price						be
The proposed approach	✓	✓	✓			✓	Daily and extrema, see Section 3.2	News, Reddit, Twitter	✓	✓	✓	✓	be

\* b = Bitcoin, e = Ethereum, r = Ripple, l = Litecoin, m = Monero, z = ZClassic, n = Electroneum, c = ZCash, d = Dash, t = Tether

## 3 Methodology

### 3.1 Deep Learning Approaches for NLP

Deep learning methods have revolutionised the field of NLP, offering context understanding, handling of linguistic ambiguities, reduced necessity for manual feature engineering, and improved generalisation capabilities. In doing so, they have enabled breakthroughs in end-to-end learning, transfer learning, multimodal integration and multilingual processing (Ruder, Vulić, and Søgaard, 2019; Young, Hazarika, Poria, and Cambria, 2017).

Traditional dictionary-based NLP approaches often treat words as independent entities. This fails to capture the nuances and complexities of language, specifically the contextual relationships between words. This issue is addressed in deep learning models, where the Transformer stands out as particularly important, alongside RNNs and CNNs.

In deep learning, words are represented as high-dimensional vectors, called embeddings, which are capable of capturing semantic and syntactic similarities (Mikolov, Chen, Corrado, and Dean, 2013). This is done through approaches like Word2Vec or GloVe, that learn from raw text, or through backpropagating loss of Transformer models like BERT, when trained on a specific task. These word embeddings are subsequently fed into an RNN, CNN, or an Attention-based neural network (for an in-depth overview of neural network architectures for NLP see Goldberg, 2017).

These models aim to capture context by modelling long-term dependencies between words. This approach addresses language ambiguity, that is, the fact that the same word can have multiple meanings. Ultimately, this enables deep learning models to encode the meaning of a sentence, or even an entire piece of text, in a context-aware fashion (Reimers and Gurevych, 2019).

Deep learning models reduce the need for extensive feature engineering (like part-of-speech tagging or named entity recognition), a common requirement in traditional NLP. Furthermore, they can learn useful features from raw text, removing the need for hand-labelled dictionaries and thus making them more scalable (LeCun, Bengio, and Hinton, 2015). This allows for end-to-end learning, where a single model processes raw text and directly outputs the final task results, such as classifications or translations, eliminating the need for complex multi-step pipelines common in traditional NLP (Bahdanau, Cho, and Bengio, 2014).

Furthermore, deep learning models have demonstrated significant efficacy in transfer learning applications within the realm of NLP. Large Language Models (LLMs) like BERT or GPT, which are pre-trained on vast corpora, can be fine-tuned on specific tasks with relatively small datasets, leveraging knowledge learned from the large-scale text collections. The models first train on a corpus of unstructured and unlabeled text data, for example, by trying to predict the next word in a sequence. This allows early layers to extract general language features, such as syntax rules or semantic relationships and acts as a basic language understanding. During fine-tuning, this

pre-trained model is adjusted to perform a specific task, like sentiment analysis. The early layers, already skilled in general language understanding, remain largely unchanged, while the later layers (e.g. a classification head) adapt to map the general language features to the specific task (Howard and Ruder, 2018).

In this work, we apply three different deep-learning-based LLMs. The first is the pre-trained sentiment analysis model Twitter-RoBERTa-Base (version from 25.01.2023). It consists of an encoder from a Transformer model (Vaswani et al., 2017), which was first pretrained on 161 GB of raw text data to become RoBERTa-Base (Liu et al., 2019), and then fine-tuned for sentiment analysis on a manually labelled dataset of 124 million tweets (Loureiro, Barbieri, Neves, Espinosa, and Camacho-Collados, 2022). With this volume of training data, it stands out as the most exhaustive sentiment analysis model tailored for social media posts. The labels are positive, neutral, and negative, which we merge into a single sentiment score.

The second model is the pre-trained zero-shot classifier BART-Large MNLI (Lewis et al., 2019). This model utilises the encoder of a pre-trained BART-Large model and is fine-tuned on the MultiNLI dataset, which contains 433 thousand sentence pairs annotated with textual entailment information. Each data point consists of (i) a “premise” – a specific piece of text, (ii) a “hypothesis” that may or may not refer to this piece of text, and (iii) a label that indicates whether the hypothesis is true, false, or unrelated to the premise. For our methodology, we input our textual data into the model as the premise. For the hypothesis, we use the sentence “This example is bullish for Bitcoin” or its Ethereum equivalent. The model then produces a score that reflects the probability of this hypothesis being true. Introducing this approach for assessing market sentiment is our primary contribution to the realm of NLP methods, going beyond what has been applied in existing financial forecasting literature.

Another contribution is the further exploration of fine-tuning LLMs for price prediction. For the third model we take a hands-on approach by fine-tuning a RoBERTa-Base model (Liu et al., 2019) directly on the cryptocurrency price. As the target for the training, we opt for daily price movements represented as a binary variable. This choice is informed by the superior performance achieved using this target in our time series models, indicating it is the easiest for ML models to predict effectively. We strategically utilise the available textual data: half of the data from each day is allocated to the training process, while the remaining half is employed to compute the final scores. We fine-tune the hyperparameters of the RoBERTa model for each text source and each coin individually, given the significant differences in text length and stylistic characteristics. To this end, we employ the Bayesian optimisation framework Optuna (Akiba, Sano, Yanase, Ohta, and Koyama, 2019). The hyperparameter search entails 240 iterations with the Area Under the Receiver Operating Characteristic Curve (AUC ROC) as the objective function. The AUC ROC is a binary classification metric that evaluates the quality of ranking positive instances above negative ones. The search ranges of the hyperparameter tuning are outlined in Table B.1 in Appendix B.

All three models handle emoticons (e.g. “:)”) and unicode (e.g. emojis) well and no additional vocabulary had to be added given that they already contain all relevant cryptocurrency related words in their pre-trained vocabulary. The cleaning of the textual data therefore only entails the removal of html elements and hyperlinks.

Table 3.1: Highest-scoring r/Bitcoin subreddit posts by NLP model

Model	Highest-scoring r/Bitcoin post
Twitter-RoBERTa (sentiment model)	So excited I finally own 50 btc!!   Thank you Bitcoin community!
BART MNLI (bullishness model)	Foreign Exchange scandal will promote Bitcoin use! Getting screwed again by banks...   German watchdog plans to step up FX probe at Deutsche. Britain’s Financial Conduct Authority began a formal investigation into possible manipulation in the \$5.3 trillion-a-day global foreign exchange market.
RoBERTa fine-tuned directly on the cryptocurrency price	Ineligible to use the Coinbase platform   I tried buying some coins just to hold on to, and I got an automated email saying my transaction was cancelled for security reasons. So I contacted support and they said: “Unfortunately a manual review has determined that you are ineligible to use the Coinbase platform to purchase Bitcoin. We’re sorry for any inconvenience that this may cause.” Has this happened to anyone else?

In the above table, we present the top-scoring post for each of the three NLP models, derived from all contributions to the r/Bitcoin subreddit. The Twitter-RoBERTa sentiment model effectively selected a notably positive post, while the BART MNLI bullishness classifier successfully chose a post that conveys optimism with regards to the future price of Bitcoin. It is noteworthy that the second post does not explicitly include terms like “bullish” or “bull”, demonstrating the model’s capability to infer higher-level semantics from the presented hypothesis.

When evaluating the third model, it is important to recognise that predicting the daily price movements is a much more complex task than sentiment analysis. Therefore, the high-scoring posts of the fine-tuned RoBERTa LLM do not necessarily convey positivity or optimism. The top post presented above suggests that perhaps the model has picked up on individuals expressing their intent to purchase the respective cryptocurrency.

## 3.2 Choice of Target Variables and Trading Strategy

In our exploration of cryptocurrency price forecasting, we utilise the CryptoCompare price at midnight for the target creation. This choice is motivated by the robustness of their CCCAGG methodology, which averages prices from 301 cryptocurrency exchanges. The weighting of this average is influenced by both the 24 hour volume and the time elapsed since the last transaction, ensuring a comprehensive and timely representation of the market.

Our first forecasting target is the log price change for the subsequent day, treated as a continuous variable. This approach is also the predominant one in the cryptocurrency price forecasting literature (see Section 2.2 “Target Variable Selection in Financial Forecasting”). The underlying trading strategy is straightforward: we buy the asset if the forecasted price change is positive and sell it if it is negative. This method offers the advantage of simplicity, but it also hinges on the precision of the continuous forecast.

Subsequently, we consider a binary representation of the next day’s price change for our second target. Here, a price increase is coded as 1, while no change or a price decrease is represented as 0. The corresponding trading strategy is to buy if the prediction exceeds a certain threshold and to sell if it falls below. This binary perspective, while reducing the granularity of the forecast, is more resilient to minor fluctuations and noise in the data.

Venturing into less charted territories, we explore forecasting local extrema for the subsequent day, spanning observational intervals of  $+/- 7$  days,  $+/- 14$  days, or  $+/- 21$  days around the given timepoint. Two binary variables are created, denoting whether a specific timepoint represents a minimum or maximum within the given time interval. These variables then serve as the targets for two separate binary classification models. While some studies have successfully applied changepoint classification for stock market prediction, this remains an unexplored approach within the cryptocurrency domain (see Section 2.2).

We hypothesise that local extrema, being less vulnerable to daily noise, might offer a more stable pattern for machine learning models to depict. The forecasting model’s enhanced accuracy could then lead to superior profitability, even though the loss of information might suggest lower potential profit. Additionally, forecasting local extrema offers a perspective to examine the impact of textual data across various observational time frames, allowing the assessment of how long-lasting the effects of market sentiments are.

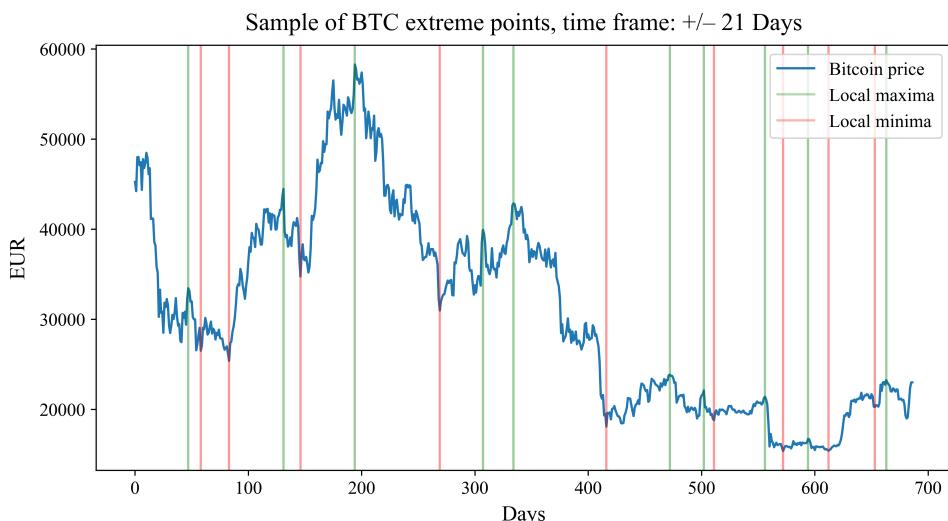


Figure 3.1: Sample of local extreme points of the BTC price  
(Author’s illustration)

For the local extrema predictions, the trading strategy is to buy if the probability of a local minimum occurring the next day exceeds a set threshold, provided the probability of a local maximum does not surpass the same threshold. Conversely, the strategy is to sell if the probability of a local maximum the next day exceeds the threshold, while the probability of a local minimum does not.

To optimise the efficacy of our strategies in the scope of the binary and change point forecasts, we determine classification thresholds using optimal Accuracy tuning. This proves to correlate well with maximum profit, while being significantly less computationally intensive than running a trading simulation. Experiments were also conducted using Balanced Accuracy, that is, the arithmetic mean of the true positive rate and the true negative rate. This resulted in the extreme point models becoming overly confident in their positive predictions though, subsequently achieving lower trading profits.

In summary, our approach to cryptocurrency price forecasting encompasses continuous, binary, and changepoint targets. While the continuous and binary methods are common practice, the changepoint analysis introduces a novel perspective to the cryptocurrency forecasting literature. This method, hypothesised to be less affected by daily market noise, aims to provide insights into the longevity of the impact of sentiments from news and social media. Our exploration underscores the importance of methodological diversity in capturing the intricate dynamics of the cryptocurrency market, as well as the potential trade-offs between simplicity, granularity, and resilience to market noise.

### 3.3 Time Series Modelling with Neural Networks

Traditional statistical methods, such as ARIMA and Exponential Smoothing, have been the go-to time series modelling techniques for decades. However, with the advent of deep learning, neural networks have emerged as a powerful alternative, offering the potential to capture more complex patterns and relationships in time series data (Zhang, Patuwo, and Hu, 1998). This has led research in the realm of financial analysis to increasingly turn to deep learning methodologies for price forecasting (Ozbayoglu, Gudelek, and Sezer, 2020). The following section provides a concise overview of neural-network-based time series modelling approaches, followed by a comprehensive examination of the TFT architecture.

The simplest form of a neural network is referred to as an MLP or Feed-Forward Neural Network (FNN). It comprises multiple layers of nodes, commonly known as neurons. Specifically, there is an input layer, one or more hidden layers, and an output layer. The strength of the connections between the nodes is defined by weights and biases, which are both adjusted using backpropagation. During backpropagation, the error of each prediction on the training dataset is circulated backward through the network, determining the contribution of each node to the overall discrepancy. By leveraging a differentiable objective function and an optimisation algorithm,

such as gradient descent, the weights and biases are iteratively tuned to minimise the error. In the case of time series data, past observations of the explanatory time series serve as input features, while future values of the time series of interest are used as the target for error computation (for an in-depth explanation of neural networks see Goodfellow, Bengio, and Courville, 2016).

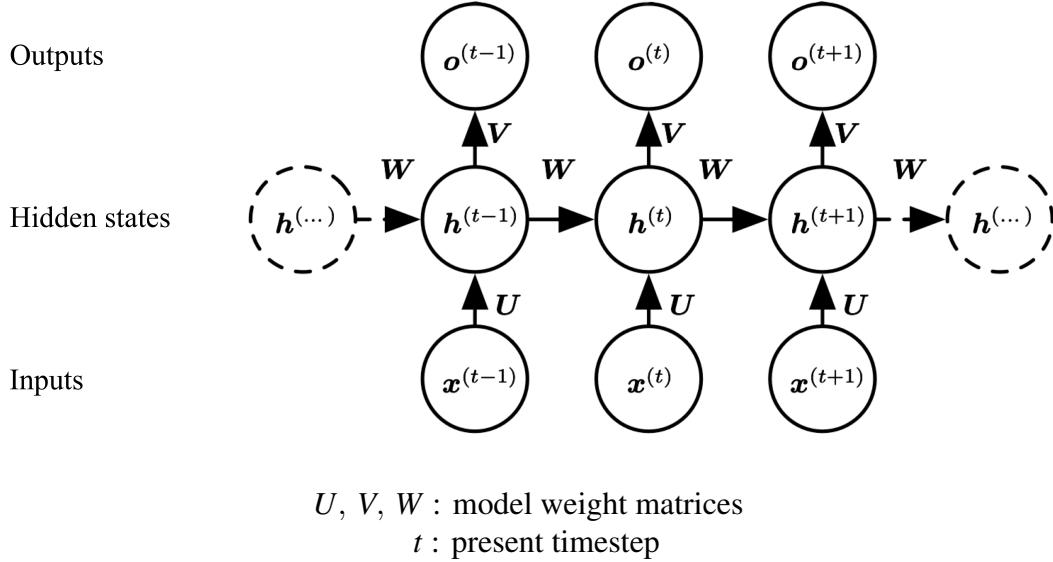


Figure 3.2: The standard RNN model architecture  
 (Adapted from Goodfellow et al., 2016)

Among the various neural network architectures, RNNs have gained prominence due to their inherent design tailored for sequential data, that is, data ordered based on the occurrence of events over the course of time. While FNNs consider each of the lagged features independently, an RNN operates in a more temporally aware fashion. Specifically, an RNN bases its predictions on the most recent lag and its internal hidden state, which ideally encapsulates historical lags and their intricate relationships. Conversely, FNNs, while utilising all provided lags, neglect the inherent sequential order and potential temporal patterns present in the data. When referring to an RNN with  $n$  neurons, we typically mean that the inputs are encoded as  $n$ -dimensional hidden states. Those hidden states can then be processed by passing them through an FNN to produce the desired output shape or by iterating the RNN architecture to generate forecasts multiple timesteps into the future.

RNNs are termed “recurrent” because their previous outputs – the past hidden states – are fed back into future iterations as input. However, RNNs are not recurrent in the sense that information ever loops back from a neuron to itself; it also passes linearly from the input to the output layer. Any RNN can, therefore, be unfolded in time and represented as an FNN for a fixed-length sequence. This representation enables the application of gradient-based techniques to train the network, a concept known as Backpropagation Through Time (see further details on RNNs in Goodfellow et al., 2016).

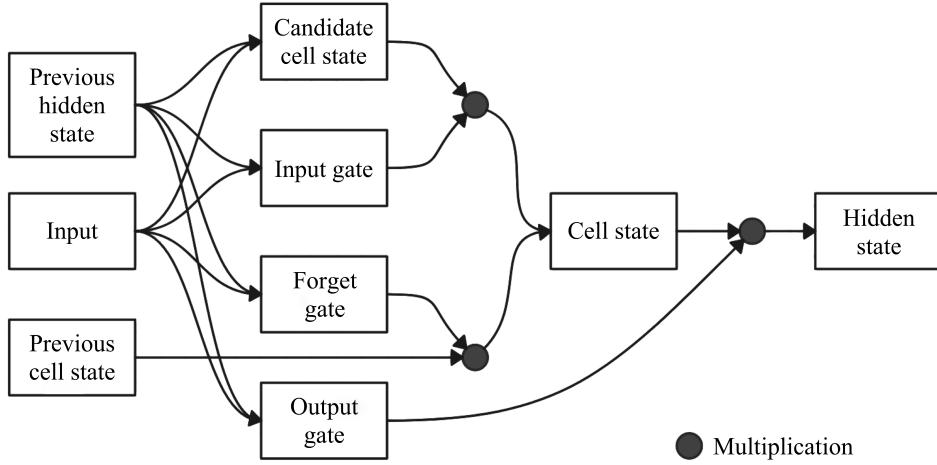


Figure 3.3: LSTM cell architecture  
(Author's illustration)

Advanced RNN architectures, notably LSTM and GRU networks, have emerged to address some intrinsic limitations of vanilla RNNs. One such challenge is the vanishing gradient problem, where the gradients diminish significantly across successive backpropagation steps, impeding the network's capacity to learn from distant past events. The LSTM architecture, first introduced by Hochreiter and Schmidhuber (1997) and then refined by Gers, Schmidhuber, and Cummins (2000), combats this issue through a more intricate design. In addition to a hidden state, it possesses a cell state, which serves as the memory of the network and can carry information untouched through many timesteps, complemented by three gates – input, forget, and output – that control the information flow.

Figure 3.4 illustrates the vanishing gradient problem and how the LSTM addresses it. In the standard RNN, the information decays over time as new inputs overwrite the hidden state. The LSTM cell state, on the other hand, passes the information from the first input as long as the forget gate is open and the input gate is closed. In practice, the gates are not strictly binary; instead, they can take on any continuous value between zero and one, allowing for nuanced modulation of the information flow.

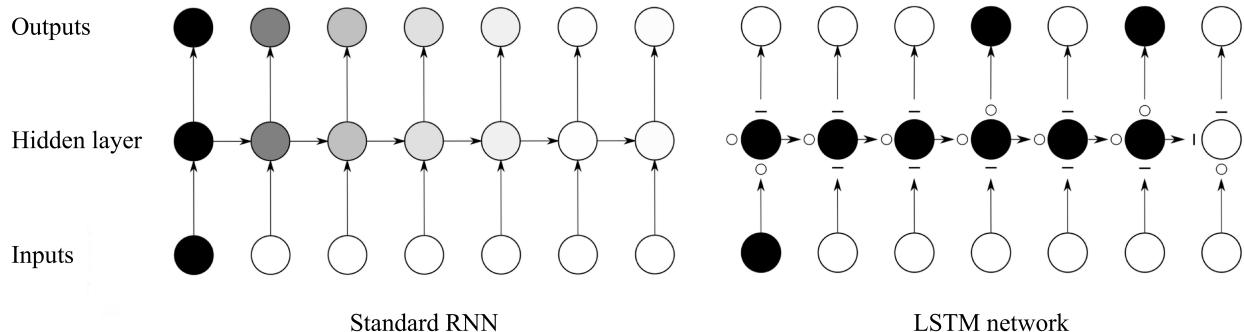


Figure 3.4: Comparison of information decay in RNNs and LSTM networks  
(Adapted from Graves, 2012)

GRUs offer a streamlined alternative to LSTMs by consolidating the cell and hidden states and employing just two gates – the update and reset gates. Despite their simplified architecture, they often match LSTMs in performance while being more computationally efficient (for a comprehensive overview of advanced RNN architectures see Graves, 2012).

CNNs also find a niche in time series forecasting. Unlike their primary use in image processing, where they identify spatial patterns, in time series, one-dimensional CNNs can identify and extract localised, shift-invariant patterns from the input data. The convolutional layers use sliding filters to learn temporal patterns, such as spikes, drops or specific shapes, with subsequent pooling and dense layers extracting dominant features and making predictions (see Ismail Fawaz, Forestier, Weber, Idoumghar, and Muller, 2019).

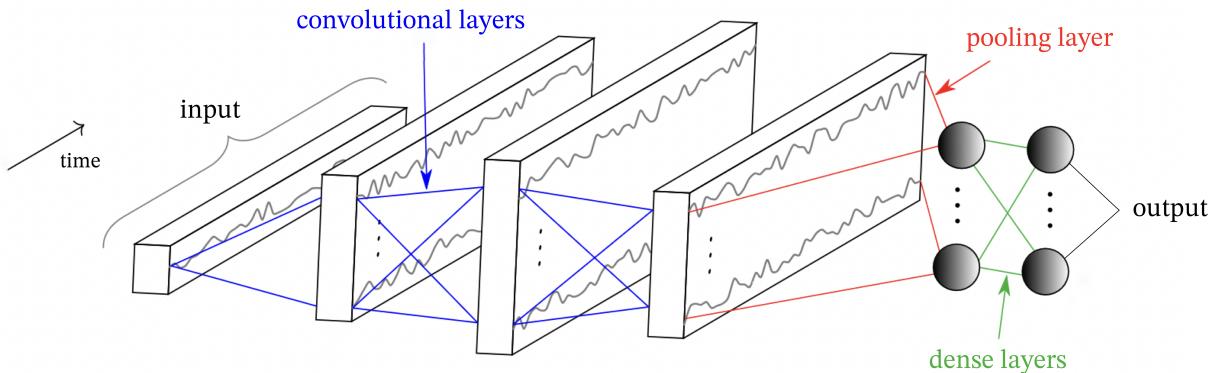


Figure 3.5: CNN time series model architecture  
(Adapted from Ismail Fawaz et al., 2019)

The TFT, introduced by Lim, Arik, Loeff, and Pfister (2020), is a notable evolution in time series modelling, blending aspects of recurrent structures with attention mechanisms. Conceptually, the TFT is best understood as a panel data model. Its design allows for the simultaneous modelling of individual-specific static metadata alongside the temporal component. This facilitates the simultaneous analysis of multiple entities across time, harnessing the full potential of both cross-sectional and time series data.

At the heart of the TFT architecture lies an LSTM encoder-decoder framework. This setup provides the model with the capability to transform input sequences into a compressed representation, which is subsequently decoded to produce the forecasted values. LSTMs, as detailed earlier, are adept at capturing sequential patterns and ensuring long-term dependencies are considered.

A distinguishing feature of the TFT is its incorporation of the Temporal Self-Attention mechanism. Borrowing insights from the Self-Attention of the Transformer language model, this mechanism enables the TFT to reweigh different points in the input sequence. By doing so, the model can discern long-spanning seasonal patterns and dependencies, particularly in situations where the influence of past events is not uniform, but varies based on context.

Integral to the TFT's design is its use of Gated Residual Networks (GRNs) for variable selection and in the later layers of the network. The potential of a GRN hinges on two key components. Firstly, the residual connection facilitates gradient propagation, allowing for the training of deeper networks without encountering the vanishing gradient problem. Secondly, the gating mechanism enables the GRN to dynamically alternate between the original input and the transformed input. This adaptability helps in selectively emphasizing certain features, proving useful for variable selection and facilitating the extraction of feature importances.

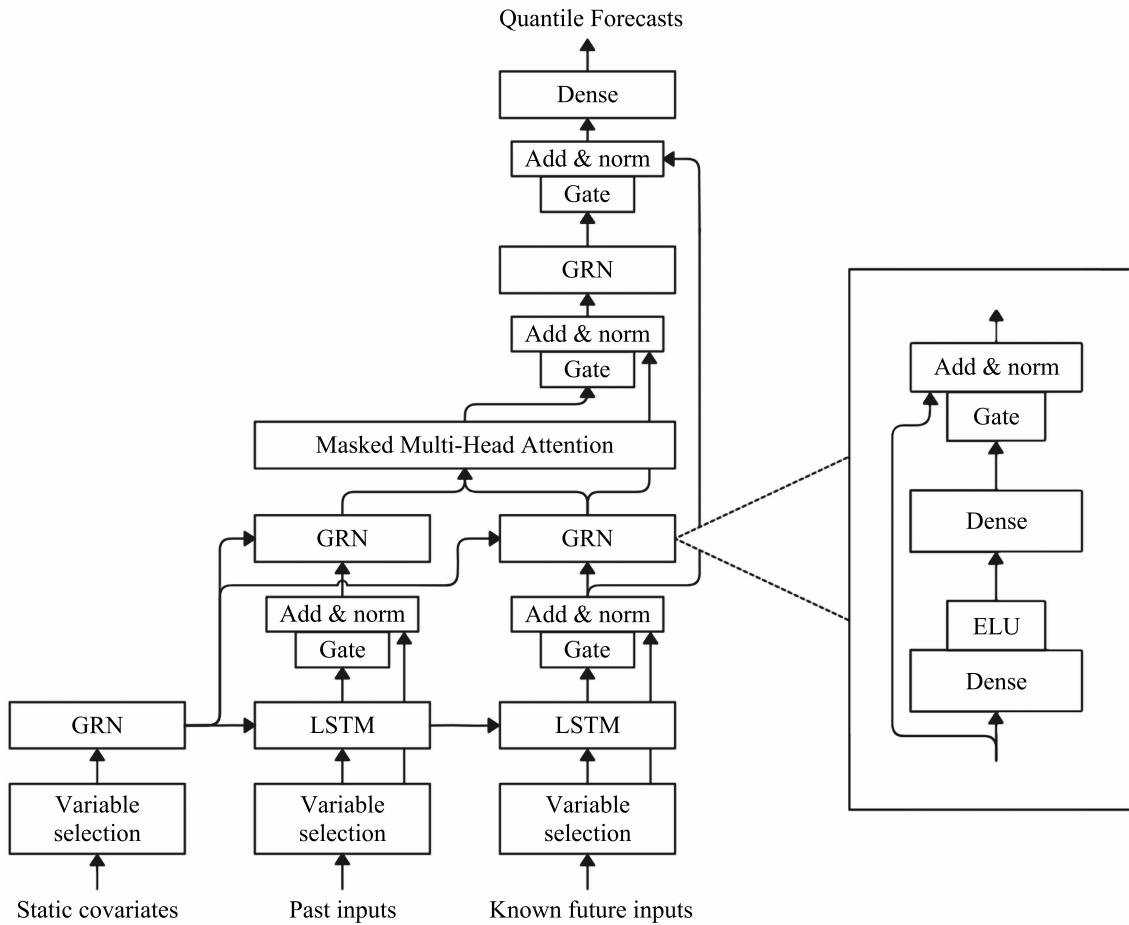


Figure 3.6: TFT model architecture  
(Author's illustration)

Yet, the merits of the TFT do not end with the capability of modelling more complex temporal relationships. Two benefits elevate its utility in practical applications: the generation of quantile forecasts and the model's inherent explainability. Instead of a single point prediction, quantile forecasts provide a range of values, akin to a confidence interval, enabling more informed decision-making strategies. The explainability through feature importances and attention weights, ensures that despite the model's complexity, the significant drivers of its predictions are discernible. In contexts where understanding the rationale behind predictions can be as critical as the forecasts themselves, those attributes contribute to the TFT's appeal.

Our research delves into the application of sequential deep learning time series models for the forecasting of cryptocurrency prices, alongside traditional non-sequential models. The complex dynamics among the explanatory time series, particularly in relation to the local extreme points, may demand a more flexible model like the LSTM network or the TFT. The TFT stands out as a prime candidate given its automatic variable selection and inherent explainability. With regard to sequential models, we also apply an LSTM network, as it has consistently demonstrated superior performance over other RNNs on a variety of different datasets (Greff, Srivastava, Koutnik, Steunebrink, and Schmidhuber, 2017). A detailed overview of all employed ML models is found in Section 4.2 “Time Series Forecasting Models”.

## 4 Experimental Design

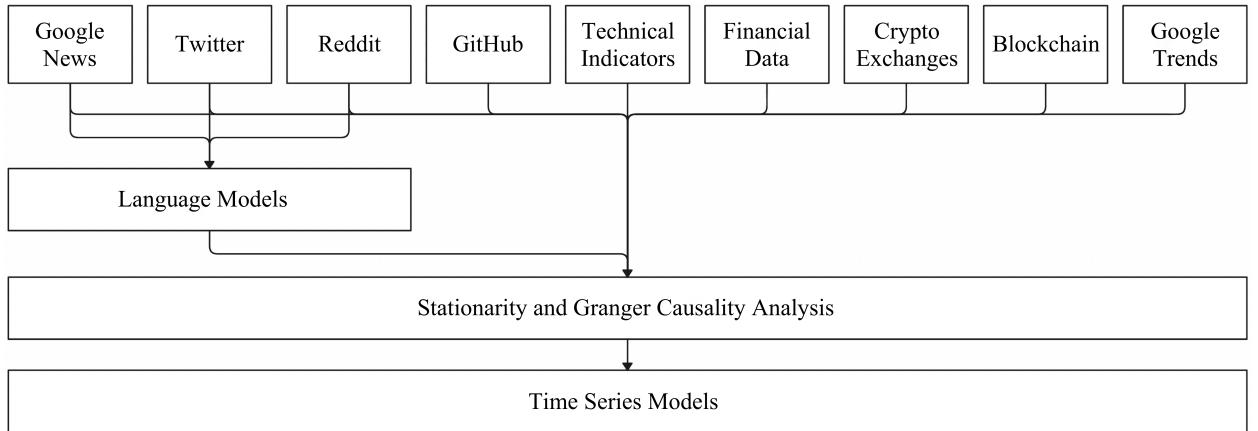


Figure 4.1: Overview of the experimental design  
(Author’s illustration)

### 4.1 Data Collection and Preprocessing

In our endeavour to forecast cryptocurrency prices, we utilise a very diverse range of data sources with the time frame of our dataset starting in August 2011 for BTC and August 2015 for ETH. To begin with, we collect text data from social media platforms and news outlets, focusing exclusively on English content. From Google News, we extract approximately 55 thousand news headlines, encompassing all articles from CoinDesk, Cointelegraph, and Decrypt that mention the keywords “Bitcoin” or “BTC” (and “Ethereum” or “ETH” respectively). On Reddit, we gather all posts from the r/Bitcoin and r/ethereum subreddits, totaling around 338 thousand threads. Finally, Twitter contributes the most to our dataset with nearly 1.9 million posts. We consider all tweets with more than five likes and two retweets that feature the hashtags #bitcoin or #btc (and correspondingly, #ethereum or #eth).

News headlines, with their formal and timely presentation of current affairs, offer a broad and credible overview of the latest events. Twitter, a popular platform among key influencers in the cryptocurrency domain, provides an unfiltered reflection of public opinion. With their concise format, tweets serve as a window into immediate personal reactions and insights, offering a snapshot of real-time sentiments. Meanwhile, Reddit posts, typically longer in nature, delve deeper into community-driven discussions, including background research and technical analysis. Together, these three sources provide a comprehensive blend of journalistic reporting, real-time reactions, and detailed community perspectives, making them invaluable for a holistic analysis of market sentiments and trends.

The text data is processed using the three LLMs detailed in Section 3.1. In addition to the textual information, we also incorporate numerical data from these platforms, such as the post count, the number of subscribers to the official Bitcoin/Ethereum Twitter accounts, and their respective subreddits, as well as the number of active users on Reddit.

Our study further integrates data from GitHub, specifically the repositories of the two cryptocurrencies we analyse: bitcoin/bitcoin and ethereum/go-ethereum. We consider the commit count, number of additions/deletions, forks, stars, and subscribers.

Shifting our focus to data more common in financial forecasting, we incorporate 48 different technical indicators based on past price and volume, including trend, momentum, volatility, and volume indicators. External financial data is also not overlooked; we include the price and volume of the S&P 500 index, the CBOE Volatility Index (VIX), COMEX gold price, and crypto indices such as the MarketVector Digital Assets 100 (MVDA) tracking the 100 largest cryptocurrencies and the Bitcoin Volatility Index (BVIN) that tracks the implied volatility of BTC using options data from Deribit. Data from cryptocurrency exchanges, detailing the volume of purchases and sales of our cryptocurrencies of interest, is also factored into our analysis.

From the blockchain, we extract data on the amount and size of transactions, account balance data, the number of newly created addresses, the number of zero balance addresses, and other technical data such as the hashrate and block size.

Lastly, we utilise Google Trends to gauge the increase or decrease in the number of searches for the queries “bitcoin”, “ethereum”, “cryptocurrency”, “blockchain”, and “investing”. An extensive overview and description of all variables can be found in Table B.3 in Appendix B.

In terms of preprocessing, we first identify variables with a unit root using the Dickey-Fuller, Phillips-Perron, and Kwiatkowski-Phillips-Schmidt-Shin tests. For such variables, we take differences. Heteroskedastic variables, identified using the White, Breusch-Pagan, and Goldfeld-Quandt tests, are logged. We consider variables as non-unit root or homoskedastic if at least two of the three corresponding tests suggest so. Outliers that seem likely to be erroneous are manually removed. For variable selection, we rely on Granger causality on lags of up to 14 days. Our decision to utilise 14 lags is underpinned by two primary observations: firstly, models trained

with this configuration demonstrated superior performance compared to those trained with 7 lags; secondly, the presence of causal lags diminished considerably beyond the 14th day.

Our Granger causality analysis reveals that both BTC and ETH exhibit sensitivity to their respective trading volumes, technical indicators, public sentiment, and broader economic trends. While both share common predictors, the BTC price showcases more pronounced sensitivity to its network metrics and large transactions, whereas ETH is affected more by its developmental community health. Examining the time frame of the impact of different features, both coins appear to be influenced by an interplay between real-time fluctuations and longer-term trends. Volume data and technical indicators have an immediate term impact of one or two days. Broader economic indicators, such as stock indices or the gold price, influence price movements over varying short-term intervals of one to two weeks. Meanwhile, NLP data has a more diverse impact, from nearly immediate to up to a week later, demonstrating the varying half-lives of different text sources in influencing cryptocurrency prices.

Additionally, we perform seasonality decompositions of the differenced price data using MSTL (Bandara, Hyndman, and Bergmeir, 2021) and Facebook’s Prophet model (Taylor and Letham, 2017). Both approaches reveal no significant daily, weekly, or monthly seasonality. Moreover, adding the day of the week or the month of the year as dummy variables does not enhance our model’s performance, and thus, we refrain from including them in the dataset. In total, we have at our disposal 137 variables, out of which between 52 and 84 are determined to be Granger causal, depending on the target.

## 4.2 Time Series Forecasting Models

In our analysis, we employ a range of sequential and non-sequential time series forecasting models. We begin with OLS-based models for benchmarking, specifically Ridge Regression for the regression problems and Logistic Regression with L2 regularisation for the binary classification problems. Our findings indicate that L2 regularisation is superior to L1 in this context, offering a more robust model fit.

Given their capacity to model complex non-linear relationships, we also apply Gradient Boosting as implemented in the XGBoost framework and a vanilla MLP. The objective functions for XGBoost are the Mean Squared Error (MSE) for regression problems and Binary Cross-Entropy (BCE) for binary classification problems. Regularisation measures comprise a combination of L1 and L2 regularisation on the leaf weights, a threshold for the addition of new leaves to a tree (also referred to as “gamma”) and subsampling.

We construct the MLP with a maximum of four layers and apply weight decay using the L2 norm to mitigate overfitting. Unlike most approaches discussed in Section 2.3 “Comparison of Cryptocurrency Price Forecasting Approaches”, we individually tune the number of neurons in each FNN layer rather than setting a uniform count across all layers. This approach provides the

models with an additional flexibility, optimising its adaptability to different data patterns. Aside from the neuron count, we tune the activation function, the batch size, the learning rate, the optimiser, and the type of scaling as hyperparameters.

Our LSTM architecture consists of up to three LSTM layers and an optional dense head with up to three dense layers. The hyperparameter tuning is very similar to that applied with the MLP. Not only the LSTM layer sizes are individually tuned, but also the neuron counts of the feed-forward layers that rest on top of them. The main difference in the tuning approach is the utilisation of “dropout”, that is, the random deactivation of neurons during training, instead of L2 regularisation.

Finally, we explore the TFT as implemented in PyTorch Forecasting. This model is particularly challenging due to its extensive training time – a result of it being fed all variables and conducting variable selection using GRNs. This method is notably less efficient than the Granger causality approach we adopt for all other models. Due to these time constraints, we opt for a uniform neuron count across all TFT layers. Furthermore, we employ dropout as the regularisation technique and set the number of attention heads relatively high, anticipating the complex seasonal patterns of our input time series. This assumption is validated as models with 16 attention heads consistently deliver superior performance.

For the sake of reproducibility, we abstain from employing early stopping in the training of the MLP, LSTM, and TFT models. Instead, we treat the number of epochs as a tunable hyperparameter. For the regression task, we configure our neural-network-based models to use a linear activation in the output layer, with backpropagation driven by the MSE. On the other hand, for the classification tasks, we employ a sigmoid activation in the output layer and use the BCE loss for backpropagation. Given the inherently imbalanced nature of classifying local extrema, we apply reweighing to the minority class for all extreme point models.

For the hyperparameter tuning, we leverage the Bayesian optimisation framework Optuna (Akiba et al., 2019) using the trading profit as the objective function. The search ranges for this tuning are detailed in Table B.2 in Appendix B. For each target and model, the optimisation process is either terminated after 200 iterations or after six weeks, whichever comes first. Notably, only the TFT ends up being constrained by the time limit.

### 4.3 Performance Metrics and Model Evaluation

We employ several evaluation metrics to assess the performance of our cryptocurrency forecasting models. Firstly, we utilise the AUC ROC to measure the model’s capability to rank positive instances higher than negative ones. The ROC curve is a graphical representation that plots the True Positive Rate against the False Positive Rate at various threshold settings. The AUC ROC is therefore bound between zero and one with higher values indicating a better quality of ranking. Additionally, we measure the Accuracy of the model, which quantifies the proportion of correct predictions relative to the total number of predictions made. Beyond these traditional metrics, we

introduce a practical evaluation based on the profitability of the model within the context of a trading strategy. To this end, we compare the profit generated by our model-driven trading decisions to a buy-and-hold benchmark. This benchmark represents a passive investment strategy where an investor buys the asset and holds onto it for the entire duration of the time period.

For the trading strategy, we start with the assumption of a portfolio value of one euro. When our model anticipates a price increase or identifies a local minimum for the subsequent day, we invest the entire available capital to buy the asset. Conversely, if the model foresees a price drop or a local maximum the next day, we liquidate all held assets. The trading strategy does not involve short selling or investing in an alternative asset, after the cryptocurrency is sold. To further ensure simplicity and interpretability of our analysis, we do not account for transaction costs. This omission is justified by the emergence of off-chain systems, such as the Lightning and Raiden networks, which enable the trading of cryptocurrencies at significantly reduced transaction costs (Hafid, Hafid, and Samih, 2020).

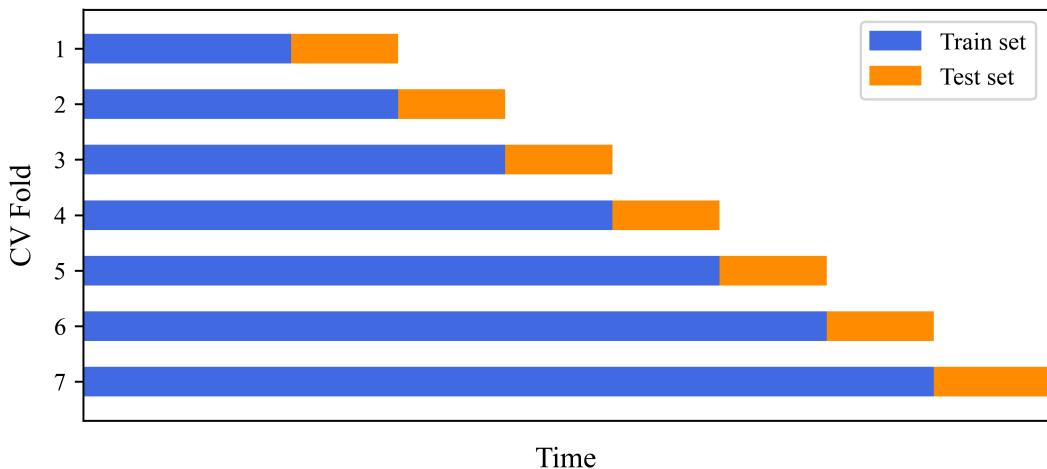


Figure 4.2: The applied time series cross-validation approach  
(Author's illustration)

All the metrics are computed as averages of a 7-fold rolling-window cross-validation with incrementally increasing training window sizes. The reasons for opting for increasing window sizes over a constant window, despite the higher computational demands, are twofold. Firstly, the increasing window approach is inherently more stable and results in lower variability of the computed metrics. Secondly, the models consistently exhibit superior performance when trained on the entirety of past data, as opposed to being limited to the most recent data points, suggesting that the underlying relationships have not changed significantly over time. The increasing window approach therefore provides a more accurate representation of model performance for the following comparative analysis.

# 5 Results and Analysis

## 5.1 Comparison of Forecasting Performance

In this section, we delve into a comprehensive examination of the BTC and ETH price forecasting performance. We apply a range of ML models, described in detail in Section 4.2 “Time Series Forecasting Models”, to five different target variables, explained in Section 3.2 “Choice of Target Variables and Trading Strategy”. Each model is trained once using financial, blockchain, GitHub, Google Trends, and numerical social media data, and then another time additionally incorporating various NLP features. The subsequent analysis not only illuminates the potential profitability of different trading strategies but also evaluates the predictive power of NLP models in the context of financial forecasting.

In framing our subsequent analysis of trading profits, we initiate by considering a few key reference points. The table presented below outlines the profits resulting from implementing a buy-and-hold trading strategy and the profit resulting from trading given perfect knowledge of the respective target variable. All values are arithmetic means of our time series cross-validation approach.

Table 5.1: Reference points for the analysis of trading profit

Trading strategy	Buy-and-hold		Perfect knowledge of target	
	Profit	Number of trades	Profit	Number of trades
<b>BTC</b>				
Price movement (binary)	279.56 %	2.0	151,887.02 %	247.1
Extrema (min/max) +/- 7 days	"	"	1,191.32 %	35.1
Extrema (min/max) +/- 14 days	"	"	1,266.22 %	16.6
Extrema (min/max) +/- 21 days	"	"	1,191.31 %	10.3
<b>ETH</b>				
Price movement (binary)	85.99 %	2.0	46,297.82 %	164.6
Extrema (min/max) +/- 7 days	"	"	327.47 %	22.6
Extrema (min/max) +/- 14 days	"	"	257.06 %	11.7
Extrema (min/max) +/- 21 days	"	"	213.23 %	8.3

The buy-and-hold benchmark represents a passive investment strategy where the asset is bought and held for the entire duration of the respective cross-validation split. On the other hand, when guided by perfect knowledge of a target variable, a trader would purchase the asset ahead of every price surge and liquidate it prior to any decline. Such a strategy represents the upper bound for the potential profit of a target variable.

A striking observation is the immense profit potential linked to daily price movements, a characteristic rooted in the inherent volatility of cryptocurrency prices. Since a substantial proportion of these daily fluctuations can be attributed to random noise, it becomes crucial to evaluate how effectively our time series models can distill the information contained in these variables. It is interesting to assess whether the daily price movements emerge as the most profitable in practice or whether the extrema prove more insightful, despite their constrained profit ceiling.

In our pursuit to understand the depth of the impact of NLP data on forecasting cryptocurrency movements, we incorporate sentiment scores from the pre-trained Twitter-RoBERTa model and bullishness classification scores from BART MNLI. Additionally, we fine-tune a RoBERTa model directly on the daily price movements. By integrating the NLP model outputs as features into our time series models, we observe a clear improvement in forecasting performance. Not only does this integration substantially enhance the profitability, it also improves the AUC ROC and the Accuracy.

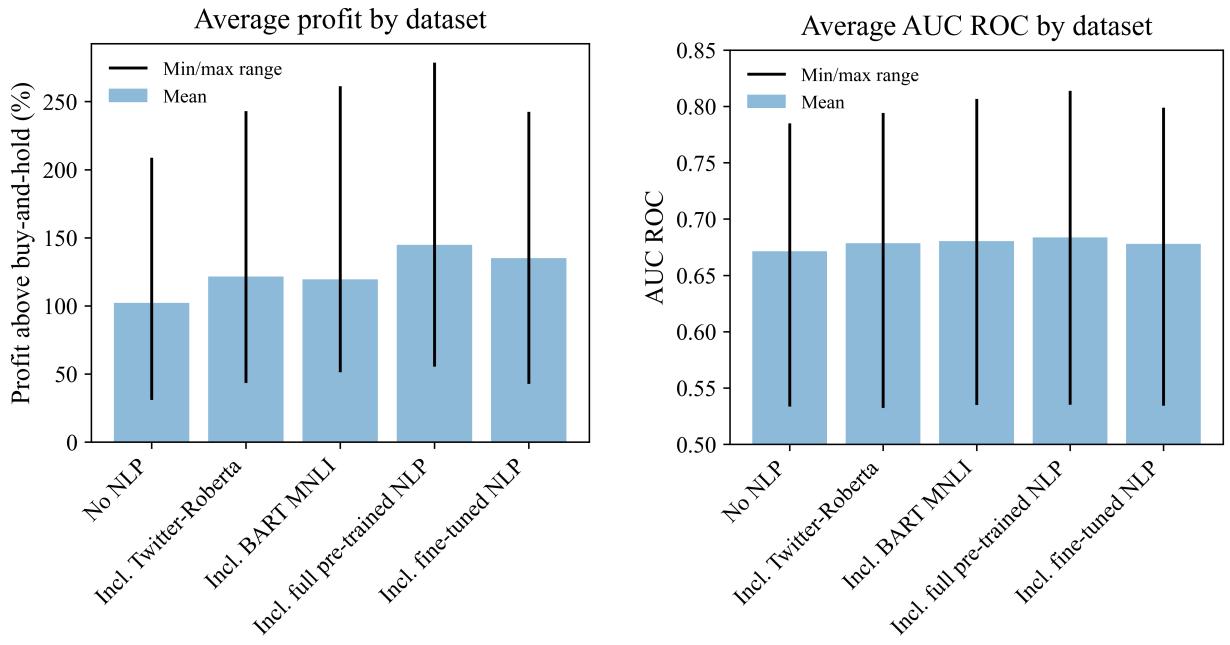


Figure 5.1: Comparison of MLP profit and AUC ROC by dataset  
(Author’s illustration)

Figure 5.1 offers a comparative analysis of the profit and the AUC ROC of an MLP model across the various sets of NLP features, aggregated over both cryptocurrencies. In this context, “full pre-trained NLP” denotes the combination of the scores from both pre-trained LLMs: Twitter-RoBERTa and BART MNLI. The profit reported is the amount of percentage points above the profit resulting from a buy-and-hold strategy, where the asset is bought at the beginning of the time period and sold at the end. Note that the whiskers indicate the variability of profits across different targets, not across cross-validation splits. Their primary function is to depict the impact of NLP

data on forecasting the individual targets, rather than to convey the statistical significance of the observed effects. To this end, Section 5.3 “Market Efficiency Throughout Time” provides a more detailed examination of the significance of NLP data across the various cross-validation splits.

An intriguing observation is the interplay between the NLP models used. While the Twitter-RoBERTa sentiment model displays higher profitability than the BART MNLI bullishness classifier, the latter demonstrates a superior AUC ROC on average. Interestingly, integrating both models yields the highest performance with regards to both metrics, underlining the potential of combining different NLP methodologies for financial forecasting.

Our main contribution in the realm of NLP is the application of the BART MNLI zero-shot classification model used to discern bullish sentiments from social media. This approach proves its viability in quantifying market sentiment. It not only identifies posts that reflect optimism with regards to future price movements with precision but also enhances the forecasting performance.

One observation that deserves a special mention relates to the performance comparison between pre-trained and fine-tuned models. The pre-trained NLP models, despite not being tailored to our dataset, yield greater benefits than those of the fine-tuned LLM. This insight underscores the potential of transfer learning in the domain of financial forecasting.

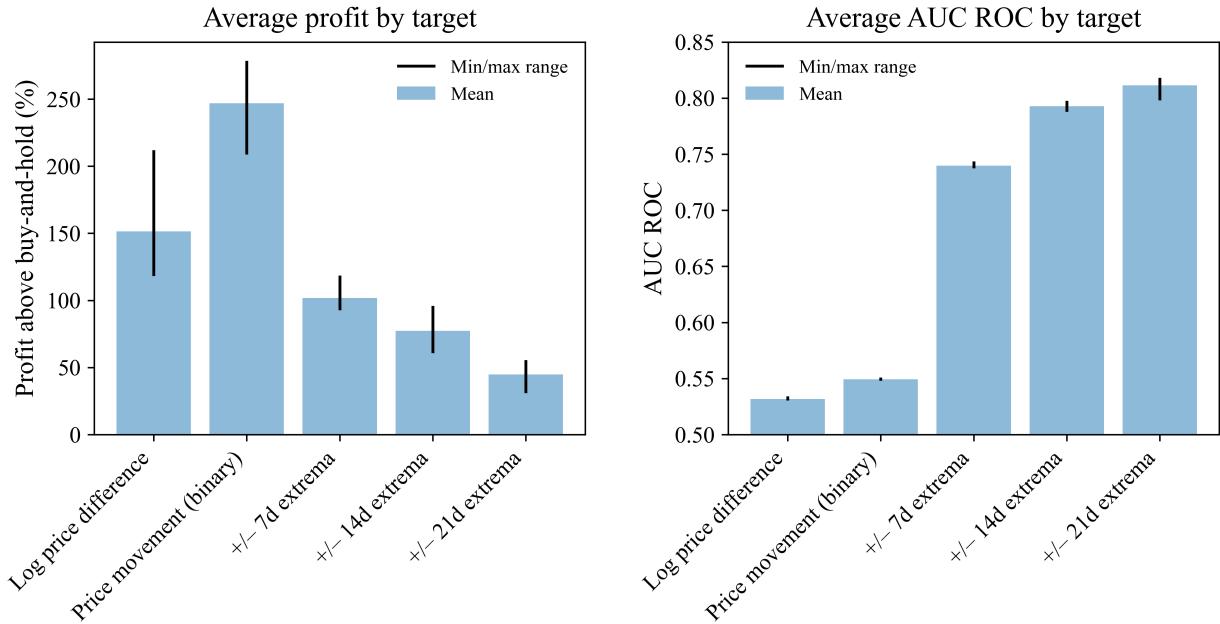


Figure 5.2: Comparison of MLP profit and AUC ROC by target variable  
(Author’s illustration)

Our second line of analysis delves into the comparative evaluation of various target variables. To this end, we employ the following targets: daily log price change treated as a continuous variable, its binary equivalent, and local extreme points (comprising local minima and maxima encoded as binary variables) across various observational time frames. Figure 5.2 provides a visual

representation, illustrating the average performance of an MLP model in terms of profit and AUC ROC broken down by target variables. Here, the whiskers represent the variability across the various sets of NLP features. A fundamental observation from our analysis is that all the chosen target types demonstrate consistent outperformance of the buy-and-hold profit accompanied by a decent AUC ROC, even without the incorporation of NLP data, thus underscoring their viability for financial forecasting.

However, we discover certain differences in performance. The binary representation of the daily price change, which simplifies the price movements into two categories of increase or decrease, consistently surpasses its counterparts in terms of profitability. This result might be a consequence of this target being less complex than local extrema, and at the same time less affected by market noise than the continuous representation of daily price movements. Conversely, when we shift our focus to the AUC ROC, the extreme point models showcase superior precision compared to the daily price change models. The class imbalance might be a concern, as it can sometimes result in misleadingly high AUC ROC values. However, even when accounting for that, they appear to have significantly better discriminatory power. Additionally, when one puts the profits generated by the extreme point models into the context of the potential profits detailed in Table 5.1, it becomes apparent that they capture a greater proportion of the information that the targets contain. In the figure below, we illustrate the contrast between the two approaches by presenting a sample of the trading performances of MLP classifiers, specifically one trained on daily price change in comparison to a prediction ensemble for  $\pm 7$  day extreme points.

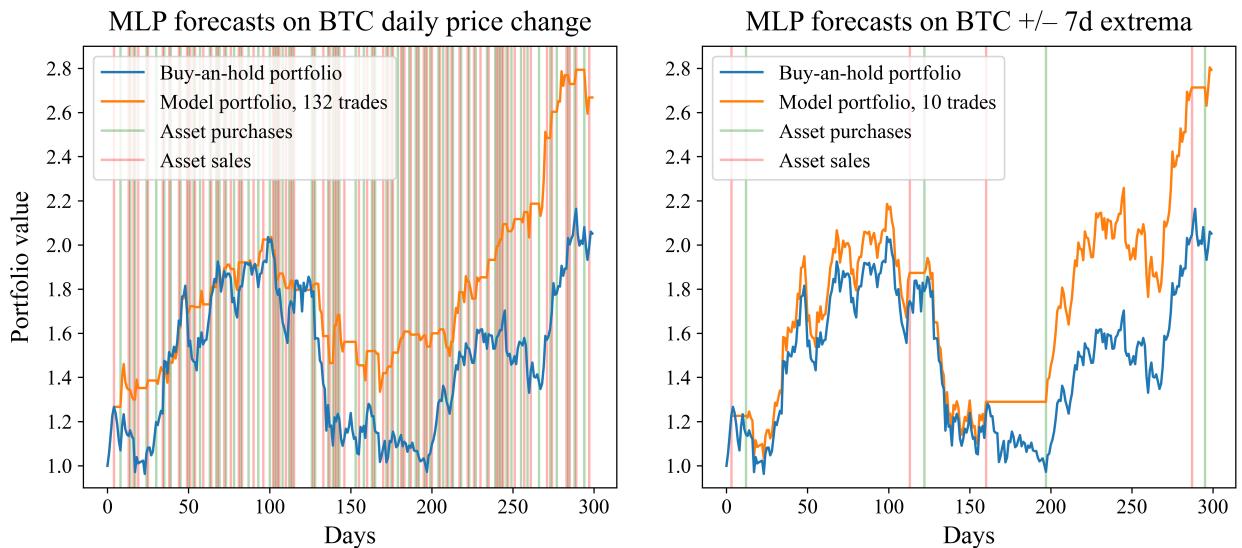


Figure 5.3: Comparison of MLPs trained on daily price change and  $\pm 7$  day extreme points  
(Author's illustration)

One particular observation worth highlighting is that the extreme point models generate commendable profits with significantly fewer trades than the models based on daily signals. An

extrema-based trading strategy could therefore be particularly useful in scenarios characterised by high transaction costs. While our simulations operate under the assumption of zero frictions, the inclusion of transaction costs might position extrema models not just as competitive, but potentially superior in terms of profitability compared to the daily models.

Expanding on our observations related to extrema prediction models, it becomes evident that not all market extremes carry equal weight. A few key turning points can have a bigger impact on profits than many smaller ones. If a model can accurately pinpoint these critical moments and skip the minor fluctuations, it stands to capture larger price movements, translating to more substantial profits. Our extreme point models seem to be adopting a “quality over quantity” approach, since it is those with fewer positive predictions (indicating extremes) and subsequently fewer trades that are the most profitable. For a detailed look at our findings, a thorough breakdown of the results can be found in tables A.2 to A.11 in Appendix A.

Navigating through the spectrum of models, we find that the OLS-based approaches already generate substantial and consistent profits as well as a decent AUC ROC, supporting the notion that financial time series exhibit primarily linear relationships. Nevertheless, all models except for the TFT consistently surpass the OLS benchmark, with the MLP taking the lead. Regardless of the target variable in play, the MLP consistently emerges as the most profitable, simultaneously clocking the highest Accuracy. The XGBoost model on the other hand produces the highest AUC ROC. Furthermore, it is the most profitable model for daily price movement prediction. A caveat worth noting here is that our hyperparameter tuning is aligned with profit optimisation. Thus, while the XGBoost model shines in terms of AUC ROC, the MLP might have outperformed XGBoost in that regard too, had it been specifically tuned with an emphasis on this metric.

Although the LSTM’s overall performance trails slightly behind the MLP, it achieves its results with significantly fewer trades, adding a layer of efficiency. In particular, the LSTM displays good capability at forecasting extreme points. Given the inherent sequential characteristics of the LSTM, it requires the input of all lags for each selected variable though. Instead of performing Granger causality analysis for every individual lag, the procedure has to be streamlined by executing it for all lags of each variable simultaneously. Consequently, this approach supplies the LSTM with a higher volume of non-causal datapoints compared to the non-sequential models, which might be the cause for the LSTM’s dampened performance.

In contrast, the weak performance of the TFT can likely be attributed to the fact that it utilises all lags of all variables and performs variable selection itself. This is significantly less efficient and, apparently, also less effective than the Granger causality approach employed for the other models, particularly given the large number of explanatory variables at hand. In addition, the reduced efficiency results in much greater training time and therefore unfortunately a less extensive hyperparameter tuning – again impacting performance. A breakdown of the average model results can be found in Table A.1 in Appendix A.

In summary, NLP proves to be notably impactful for cryptocurrency price forecasting. We observe that the efficacy of pre-trained NLP models is impressive, aligning closely with their fine-tuned counterparts. This finding underscores the robustness and adaptability of LLMs in financial forecasting scenarios. Turning our attention to the target variables, daily price movements emerge as the most profitable target. However, it is essential to highlight that predictions focusing on extrema are not just mere alternatives but stand as strong contenders, especially in market environments characterised by elevated transaction costs.

We determine that our linear models, while delivering good performance, find themselves overshadowed by their non-linear counterparts. This distinction becomes especially pronounced when forecasting local extreme points. Lastly, an unexpected observation emerges concerning the performance of sequential models. Contrary to initial expectations, these models do not outperform their non-sequential counterparts. We hypothesise that this outcome stems from the inherent challenges the sequential models encounter in variable selection.

## 5.2 Analysis of Feature Importance

In our effort to understand the significance of the individual variables, we utilise an XGBoost model trained on the entire set of available features. As the target for this training we utilise daily price movements encoded as a binary variable. The rationale for this approach is twofold. Firstly, this model configuration yielded the highest AUC ROC for both BTC and ETH, while claiming the top 2 and top 3 positions in terms of profitability, respectively. Secondly, extracting feature importances from XGBoost has the advantage of reflecting the actual dynamics of the model, rather than being a mere approximation.

XGBoost provides multiple metrics to determine the significance of each feature: “gain”, “weight”, and “cover”. Gain represents the enhancement in the optimisation objective brought about by a split on a particular feature. In our case of binary classification, this objective is the BCE loss. Weight gives insight into the frequency with which a feature appears across all trees in the model. A feature with a higher weight value has been selected as a splitting criterion more often when constructing the model. However, a feature may be frequently used without having a substantial impact on the outcome. Lastly, the cover metric measures the relative quantity of observations affected by a feature. A high cover value indicates that the feature is responsible for splitting a considerable number of samples.

We report average and total gain since they quantify the contribution a feature brings to the model’s predictive capability. The average gain represents how beneficial, on average, splits on a specific feature are when they are made. Total gain, on the other hand, aggregates these benefits across all trees, representing a features’ cumulative contribution to the model performance. For the sake of clarity and interpretability, we present the normalised values of these metrics. Portraying them as fractions of the total makes them interpretable as percentages of overall importance.

In the subsequent tables, the feature importances are aggregated for all lags of each feature. Moreover, given the extensive list of variables, we present the importance scores consolidated by feature category. An overview of the individual features used for the analysis can be found in Section 4.1 “Data Collection and Preprocessing”. For those inclined to delve into the granular details of the feature importances, the disaggregated results are annexed in tables A.12 and A.13 of Appendix A.

Table 5.2: Feature categories ranked by their importance for predicting BTC price movements

<b>Feature category*</b>	<b>Normalised total gain</b>	<b>Normalised average gain</b>
Technical indicators	0.3722	0.4998
Transaction and account balance data	0.2462	0.1558
NLP data	0.1273	0.1224
Exchange volume data	0.0753	0.0694
Technical blockchain metrics	0.0556	0.0545
Numerical social media data	0.0521	0.0366
Google Trends	0.0262	0.0185
Past price data	0.0235	0.0187
Financial data	0.0168	0.0198
GitHub metrics	0.0049	0.0046

\* Categories are sorted by total gain

Upon evaluating the BTC feature importances, it is evident that technical indicators hold a preeminent position. The prominence of transaction and account balance data, ranked as the second most relevant category, highlights the valuable insights drawn from the transparent nature of individual wallet holdings. Additionally, the NLP scores and post counts of Reddit and Twitter are noteworthy, emphasising the importance of textual data in financial forecasting. In particular, our fine-tuned RoBERTa model that was trained on the corpus of Tweets stands out by claiming the top position (as seen in Table A.12 in Appendix A).

Technical metrics related to the blockchain, such as the hashrate or block size, also prove influential. Interestingly, of all the Google search query trends, “blockchain” ranks notably higher than the rest. This may suggest that a curiosity about blockchain’s mechanics indicates a more profound interest in Bitcoin than just googling its name and, therefore, a higher likelihood of future purchase.

Other relevant features include past lags of price and volume as well as the circulating amount of the currency. Lastly, GitHub metrics appear to have the least impact on the model, suggesting that while the health of the developmental community in open-source projects is crucial, its bearing on the BTC price might be minimal.

Table 5.3: Feature categories ranked by their importance for predicting ETH price movements

<b>Feature category*</b>	<b>Normalised total gain</b>	<b>Normalised average gain</b>
Technical indicators	0.5691	0.6714
NLP data	0.2220	0.1657
GitHub metrics	0.0541	0.0526
Exchange volume data	0.0477	0.0334
Numerical social media data	0.0341	0.0176
Transaction and account balance data	0.0301	0.0207
Past price data	0.0129	0.0097
Financial data	0.0120	0.0121
Google Trends	0.0091	0.0070
Technical blockchain metrics	0.0089	0.0099

\* Categories are sorted by total gain

Examining the ETH feature set, technical indicators persist in their dominance. Moreover, the significance of NLP models, particularly Twitter-RoBERTa and our fine-tuned RoBERTa model, is even more accentuated, reaffirming the overarching influence of social media on Ethereum’s price dynamics. Other notable variables include the active user count on Reddit, trading data from various exchanges, and intriguingly, several metrics from GitHub, specifically the number of created and resolved issues, and the commit count. Being an indicator of upcoming technical changes, developmental activity may be of particular importance in the case of ETH, considering its transition from a proof-of-work to a proof-of-stake consensus mechanism in 2022. Further variables of interest encompass transaction and account balance data as well as numerical social media data, such as the subscriber counts of the ETH Twitter account or subreddit.

The prominence of technical indicators for both cryptocurrencies can be attributed to a number of factors. Firstly, while we provide the model with lags up to 14 of price and volume, some indicators can access a more extensive lookback period, thus encompassing more long-term information. Secondly, these indicators simplify intricate relationships into more digestible signals, making it easier for the model to discern patterns and trends that might otherwise be obscured in the raw data, especially considering our dataset’s relatively limited size of a couple thousand observations. Thirdly, while the model only has access to the lags selected as relevant by Granger causality analysis, indicators like moving averages can combine the information of several consecutive lags, that may be missing in the feature set, in condensed form. Another dimension worth considering is the historical reliance of human traders on these indicators. If a significant section of market participants leans on these tools to make decisions, then the price movement will inherently reflect the signals from these indicators. Finally, the intrinsic smoothing within some of the used indicators can combat the noise in the raw data, acting as a form of implicit regularisation.

While our feature importance analysis underscores the significance of technical indicators, the outputs of our NLP models, especially those representing Twitter and Reddit content, manifested as some of the most impactful explanatory variables for both BTC and ETH. This reaffirms our earlier conclusions that social media plays a pivotal role in influencing cryptocurrency price dynamics. Additionally, various data from the blockchain, exchange trading volumes, and metrics representing developer activity on GitHub have emerged as relevant.

### 5.3 Market Efficiency Throughout Time

Market efficiency refers to the idea that asset prices in financial markets reflect all available information at any given time. The concept is rooted in the Efficient Market Hypothesis, which asserts that it is impossible to consistently achieve returns in the market that outperform average market returns on a risk-adjusted basis.

In a market environment that is efficient, particularly in the semi-strong or strong form, consistently outperforming the market becomes challenging. The reason is that any new piece of information that could influence an asset's price is almost instantaneously incorporated into its price. This rapid incorporation offers minimal opportunities for traders to exploit the information for their advantage (for an extensive review of the theoretical and empirical backgrounds of market efficiency see Shleifer, 2009).

If a trader consistently achieves above-market profits, it could signify one of several situations: (i) the market is not efficient, (ii) the trader possesses a unique skill or system that the broader market has not adopted yet, or (iii) the trader is taking on higher risks to achieve those returns. Given that the profit of our trading portfolio is likely influenced by the latter two factors, we direct our attention to the trajectory of the profit over time, sidestepping the question whether the cryptocurrency market is efficient or not. However, the pronounced feature importance of technical indicators and volume data discussed in the previous section hints at the possibility that crypto markets might not even achieve efficiency in the weak form.

We employ a range of ML models to forecast the price movements of BTC and ETH. Some of these models are trained only using financial, blockchain, GitHub, Google Trends, and numerical social media data, while some additionally incorporate NLP features. The models are evaluated by computing metrics on a 7-fold increasing window time-series cross-validation. By evaluating the model profits over the course of these seven cross-validation splits, we intend to provide insights relating to the development of market efficiency throughout time.

Our evaluation will illuminate the consistency of the profits, hence providing an insight into the risk profile of the models' trading portfolio compared to the underlying cryptocurrency. Additionally, by observing if profits exhibit a trend over time, we can gauge if the market is increasingly integrating NLP into their trading strategies, which might consequently shrink the potential gains from text analysis.

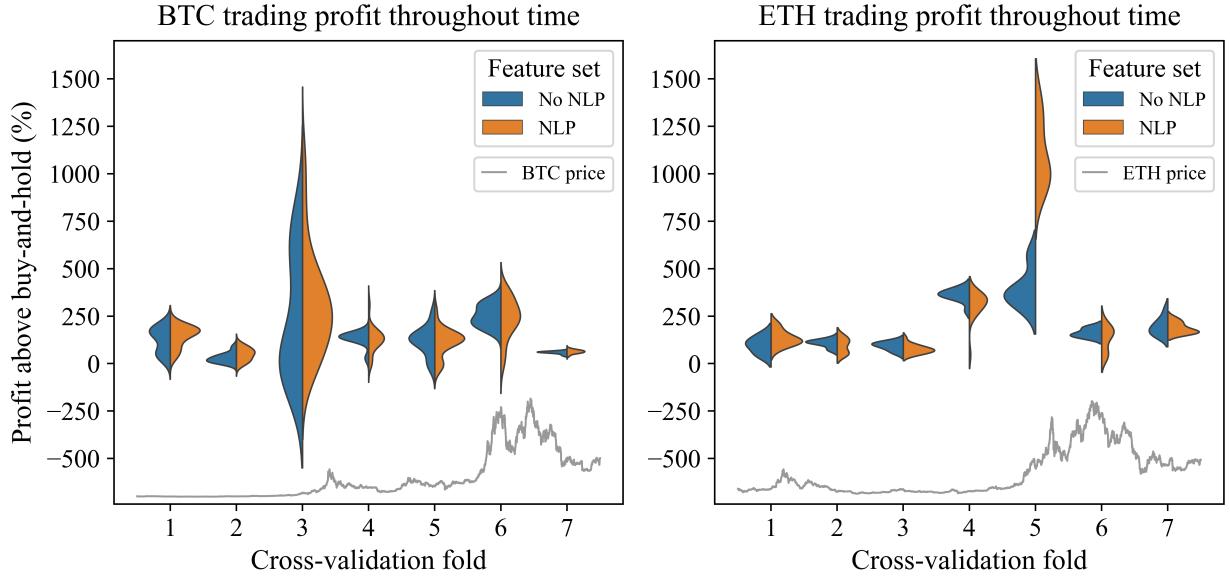


Figure 5.4: Distribution of trading profits of the 10 most profitable MLP models throughout time  
(Author's illustration)

The plots above display the kernel density estimations for the trading profits of the 10 most profitable MLP models across the cross-validation splits. The choice of focusing on the MLP model was motivated by it being the most profitable among the ML models.

We observe that our models consistently deliver trading profits that surpass the buy-and-hold benchmark. However, it is essential to highlight that while the profits remain above the benchmark for every cross-validation fold, the magnitude of profit does experience substantial fluctuations across the time splits. It is evident that during phases marked by heightened volatility of the underlying cryptocurrency, our models display a pronounced outperformance of the buy-and-hold approach.

Intriguingly, during the seventh split of the BTC data, the MLPs barely managed to outdo the benchmark, even though they managed to do so effectively on the first and fourth split, which were also characterised by a price decline. This may suggest that the models used for the predictions, which have been exposed to the particularly bullish periods in the fifth and sixth split during training, might struggle to capitalise on volatile bearish periods effectively.

Nevertheless, as we span across the various splits, there is no observable upwards or downwards trend in profits, whether in terms of NLP effects or overall excess profit. This suggests that over the periods examined, the market's efficiency, or lack thereof, appears to remain largely unchanged. This reinforces the potential viability of our forecasting methods in future scenarios.

Furthermore, the violin plot sheds light on the significance of the impact of NLP data on our models' forecasting performance. For BTC, introducing NLP data into our models slightly nudges the profit distributions upwards for most splits, suggesting that the numerical

representations derived from the textual data consistently provide information that is useful for predicting price movements. A second aspect is the potential reduction in volatility, most evident during the third split. This might indicate that linguistic data introduces more nuanced information, especially beneficial during turbulent market phases.

In the case of ETH, the benefits of NLP appear more period-specific, with notable advantages emerging in the fifth split, which was also characterised by the highest excess profit of the models trained without NLP data. Yet, in other splits, the NLP data seems less consequential, either offering limited enhancement or even marginally impeding the forecast. It is possible that during this specific period there was social media activity that was especially indicative of ETH’s price movement. However, it seems more plausible to attribute the selective impact of the NLP data to the fact, that the fifth split is notably volatile and bullish. Given that these conditions present an elevated opportunity for ML models to leverage price fluctuations, they might have naturally become the primary target for our models, which were tuned with the objective of maximising trading profit.

In summary, our models consistently outperform the buy-and-hold benchmark across multiple splits, suggesting a robustness in their forecasting capability. This outperformance becomes particularly pronounced during volatile market conditions, though there are indications that training predominantly on bullish data may limit the models’ effectiveness during bearish cycles. Additionally, while the integration of NLP data offers consistent benefits in predicting BTC’s price movements, its impact on ETH predictions is more sporadic and largely confined to a few specific time frames. Importantly, throughout our examination, there is no discernible trend indicating diminishing or increasing profits over time, indicating that the market’s efficiency remains relatively constant during the observed periods. This suggests that our methodological framework, particularly when complemented by textual data, holds significant promise for future cryptocurrency forecasting endeavours.

## 6 Conclusion

### 6.1 Summary of Findings

In this study, we aim to explore the viability of news and social media data for cryptocurrency price forecasting. We are particularly interested in the time frame of the impact of textual data and the differences between various types of target variables. With regards to the targets for training, we utilise changepoints (local minima and maxima) with varying observational time frames in addition to daily price movements. In the context of NLP, we focus on investigating the application of multiple deep learning techniques. Moreover, we seek to assess the evolution of the market efficiency over time.

Our findings demonstrate that integrating NLP data improves the forecasting performance of our ML models with respect to all metrics. We find that pre-trained models, namely Twitter-RoBERTa and BART MNLI, show promising capabilities in effectively capturing market sentiment, performing on par with LLMs that are fine-tuned directly on the target at hand. In this scope, it is worth noting the considerable improvements in forecasting accuracy brought about by using the BART MNLI zero-shot classification model to extract bullish and bearish signals from social media.

Additionally, our results indicate that text features lagged by up to one week are Granger causal and that incorporating NLP data in the time series models results in enhancements in the forecasting of 21-day extrema. These findings suggest that news and social media can have a more long-term impact on price movements.

In terms of model performance, we find that non-linear models outperform those based on OLS, demonstrating the existence of relevant non-linear relationships in the time series. However, linear models already produce commendable results, reaffirming the widely held notion that financial time series predominantly exhibit linear relationships. We further identify that using the daily price change as a binary target variable consistently results in superior profitability compared to other targets, at least under the assumption of no transaction costs. Nevertheless, our models are able to more reliably predict local extrema than daily price fluctuations, and the extreme point models yield decent profits with significantly fewer trades.

All models consistently generate profits throughout all cross-validation splits and we do not observe a decrease in overall profits or a reduction in the impact of NLP data across time. This suggests potential for the continued use of text analysis to enhance financial forecasts in the future.

## 6.2 Implications

Our study has several implications for the field of price forecasting. The incorporation of NLP data into our models significantly improves forecasting performance, demonstrating the value of considering such data in predictive efforts. When turning our attention to the target variables, daily price movements encoded as a binary target consistently yield the highest profit. However, our models are also able to capture valuable information when employing local extrema as target variables. This suggests that although daily price movement may maximise profit, the use of extrema as target variables potentially offers deeper insights into the underlying market dynamics and proves useful in circumstances where one aims to reduce the number of trades, for example, in the face of high transaction costs.

Regarding the realm of NLP, our study provides important insights into the effectiveness of certain models in capturing market sentiment. In particular, we find BART MNLI to be highly proficient as a zero-shot classifier. It effectively interprets market sentiment expressed through text that extends

beyond mere positivity or negativity. Furthermore, fine-tuning an LLM on the target variable results in considerable improvement in forecasting performance compared to models without NLP. This is the case even though the task of predicting price movements is far more abstract than traditional sentiment analysis.

Nevertheless, our results indicate that pre-trained models deliver comparable, if not superior, results to fine-tuned models, even when tackling abstract tasks in a specialised domain like finance. This finding suggests promising prospects for the use of transfer learning in NLP and not only highlights the versatility and robustness of these models but also points towards a cost and time-effective route for future endeavours in financial forecasting.

# References

- Abraham, J., Higdon, D., Nelson, J., Ibarra, J., and Nelson, J. (2018). Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1. Retrieved from <https://scholar.smu.edu/cgi/viewcontent.cgi?article=1039&context=datasciencereview>
- Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019, 07). Optuna: A next-generation hyperparameter optimization framework. *arXiv:1907.10902 [cs, stat]*. Retrieved from <https://arxiv.org/abs/1907.10902>
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate*. Retrieved from <https://arxiv.org/abs/1409.0473>
- Bakar, N. A., Rosbi, S., and Uzaki, K. (2020). Weighted moving average method for forecasting of cryptocurrency price: A data analytical study on xrp ripple cryptocurrency. *International Journal of Advances in Scientific Research and Engineering*, 06, 127-133. doi: 10.31695/ijasre.2020.33691
- Bandara, K., Hyndman, R. J., and Bergmeir, C. (2021, 07). Mstl: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns. *arXiv:2107.13462 [stat]*. Retrieved from <https://arxiv.org/abs/2107.13462>
- Colianni, S., Rosales, S., and Signorotti, M. (2015). *Algorithmic trading of cryptocurrency based on twitter sentiment analysis*. Retrieved from [http://cs229.stanford.edu/proj2015/029\\_report.pdf](http://cs229.stanford.edu/proj2015/029_report.pdf)
- Derbentsev, V., Matviychuk, A., and Soloviev, V. N. (2020). Forecasting of cryptocurrency prices using machine learning. *Advanced Studies of Financial Technologies and Cryptocurrency Markets*, 211-231. doi: 10.1007/978-981-15-4498-9\_12
- Dwivedi, H. (2023, 03). *Cryptocurrency sentiment analysis using bidirectional transformation*. Retrieved from <https://ieeexplore.ieee.org/abstract/document/10127932> doi: 10.1109/ICSMID57622.2023.00032
- Fama, E. F. (1970, 05). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25, 383-417. Retrieved from <https://www.jstor.org/stable/2325486> doi: <https://doi.org/10.2307/2325486>
- Ferraz, F., and Moura, V. (2017). A bayesian model for multiple change point to extremes, with application to environmental and financial data. *Journal of Applied Statistics*, 44. doi: 10.1080/02664763.2016.1254733
- Gers, F., Schmidhuber, J., and Cummins, F. (2000, 10). Learning to forget: Continual prediction with lstm. *Neural Computation*, 12, 2451-2471. doi: 10.1162/089976600300015015
- Goldberg, Y. (2017). *Neural network methods in natural language processing*. Morgan & Claypool Publishers.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

- Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*. Springer.
- Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., and Schmidhuber, J. (2017, 10). Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28, 2222-2232. doi: 10.1109/tnnls.2016.2582924
- Hafid, A., Hafid, A. S., and Samih, M. (2020). Scaling blockchains: A comprehensive survey. *IEEE Access*, 8, 125244-125262. doi: 10.1109/access.2020.3007251
- Hochreiter, S., and Schmidhuber, J. (1997, 01). Long short-term memory. *Neural Computation*, 9, 1-42. Retrieved from <https://www.bioinf.jku.at/publications/older/2604.pdf> doi: 10.1162/neco.1997.9.1.1
- Howard, J., and Ruder, S. (2018). *Universal language model fine-tuning for text classification*. Retrieved from <https://arxiv.org/abs/1801.06146>
- Ider, D., and Lessmann, S. (2022, 04). Cryptocurrency return prediction using investor sentiment extracted by bert-based classifiers from news articles, reddit posts and tweets. *arXiv:2204.05781 [cs, q-fin]*. Retrieved from <https://arxiv.org/abs/2204.05781>
- Inamdar, A., Bhagtni, A., Bhatt, S., and Shetty, P. M. (2019, 05). *Predicting cryptocurrency value using sentiment analysis*. Retrieved from <https://ieeexplore.ieee.org/abstract/document/9065838> doi: 10.1109/ICCS45141.2019.9065838
- Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019, 03). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*. doi: 10.1007/s10618-019-00619-1
- Jain, A., Tripathi, S., Dwivedi, H. D., and Saxena, P. (2018, 08). Forecasting price of cryptocurrencies using tweets sentiment analysis. *2018 Eleventh International Conference on Contemporary Computing (IC3)*. doi: 10.1109/ic3.2018.8530659
- Jay, P., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N., and Alazab, M. (2020). Stochastic neural networks for cryptocurrency price prediction. *IEEE Access*, 8, 82804-82818. doi: 10.1109/access.2020.2990659
- Karalevicius, V., Degrande, N., and De Weerdt, J. (2018, 01). Using sentiment analysis to predict interday bitcoin price movements. *The Journal of Risk Finance*, 19, 56-75. doi: 10.1108/jrf-06-2017-0092
- Kim, G., Kim, M., Kim, B., and Lim, H. (2023). Cbits: Crypto bert incorporated trading system. *IEEE Access*, 11, 6912–6921. Retrieved 2023-08-09, from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10014986> doi: 10.1109/ACCESS.2023.3236032
- Kim, G., Shin, D.-H., Choi, J. G., and Lim, S. (2022). A deep learning-based cryptocurrency price prediction model that uses on-chain data. *IEEE Access*, 10, 56232-56248. doi: 10.1109/access.2022.3177888
- LeCun, Y., Bengio, Y., and Hinton, G. (2015, 05). Deep learning. *Nature*, 521, 436-444. Retrieved from <https://www.nature.com/articles/nature14539> doi: 10.1038/nature14539
- Leung, M. T., Daouk, H., and Chen, A.-S. (2000, 04). Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting*, 16, 173-

190. doi: 10.1016/s0169-2070(99)00048-5
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2019, 10). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv:1910.13461 [cs, stat]*. Retrieved from <https://arxiv.org/abs/1910.13461>
- Li, T. R., Chamrajanagar, A. S., Fong, X. R., Rizik, N. R., and Fu, F. (2019, 07). Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model. *Frontiers in Physics*, 7. doi: 10.3389/fphy.2019.00098
- Liang, M., Wang, X., and Wu, S. (2022, 11). Improving stock trend prediction through financial time series classification and temporal correlation analysis based on aligning change point. *Soft Computing*. doi: 10.1007/s00500-022-07630-7
- Lim, B., Arik, S. O., Loeff, N., and Pfister, T. (2020, 09). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv:1912.09363 [cs, stat]*. Retrieved from <https://arxiv.org/abs/1912.09363>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. Retrieved from <https://arxiv.org/abs/1907.11692>
- Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., and Pintelas, P. (2021, 01). An advanced cnn-lstm model for cryptocurrency forecasting. *Electronics*, 10, 287. doi: 10.3390/electronics10030287
- Loureiro, D., Barbieri, F., Neves, L., Espinosa, L., and Camacho-Collados, J. (2022, 04). Timelms: Diachronic language models from twitter. Retrieved from <https://arxiv.org/pdf/2202.03829.pdf>
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). *Distributed representations of words and phrases and their compositionality*. Retrieved from <https://arxiv.org/pdf/1310.4546.pdf>
- Mittal, A., Dhiman, V., Singh, A., and Prakash, C. (2019, 08). Short-term bitcoin price fluctuation prediction using social media and web search data. *2019 Twelfth International Conference on Contemporary Computing (IC3)*. doi: 10.1109/ic3.2019.8844899
- Mudassir, M., Bennbaia, S., Unal, D., and Hammoudeh, M. (2020, 07). Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*. doi: 10.1007/s00521-020-05129-6
- Murray, K., Rossi, A., Carraro, D., and Visentin, A. (2023, 01). On forecasting cryptocurrency prices: A comparison of machine learning, deep learning, and ensembles. *Forecasting*, 5, 196-209. doi: 10.3390/forecast5010010
- Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W., and Wills, G. (2020, 05). *A survey on machine learning for stock price prediction: Algorithms and techniques*. Retrieved from <https://eprints.soton.ac.uk/437785/> doi: 10.5220/0009340700630071
- Ortu, M., Uras, N., Conversano, C., Bartolucci, S., and Destefanis, G. (2022, 07). On technical trading and social media indicators for cryptocurrency price classification through deep

- learning. *Expert Systems with Applications*, 198, 116804. doi: 10.1016/j.eswa.2022.116804
- Ozbayoglu, A. M., Gudelek, M. U., and Sezer, O. B. (2020, 08). Deep learning for financial applications: A survey. *Applied Soft Computing*, 93, 106384. doi: 10.1016/j.asoc.2020.106384
- Pant, D. R., Neupane, P., Poudel, A., Pokhrel, A. K., and Lama, B. K. (2018, 10). Recurrent neural network based bitcoin price prediction by twitter sentiment analysis. *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. doi: 10.1109/cccs.2018.8586824
- Parekh, R., Patel, N. P., Thakkar, N., Gupta, R., Tanwar, S., Sharma, G., . . . Sharma, R. (2022). Dl-guess: Deep learning and sentiment analysis-based cryptocurrency price prediction. *IEEE Access*, 1-1. doi: 10.1109/access.2022.3163305
- Pintelas, E., Livieris, I. E., Stavroyiannis, S., Kotsilieris, T., and Pintelas, P. (2020, 05). Investigating the problem of cryptocurrency price prediction: A deep learning approach. *Artificial Intelligence Applications and Innovations*, 584, 99–110. Retrieved from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7256561/> doi: 10.1007/978-3-030-49186-4\_9
- Rafi, M., Mirza, Q. A. K., Sohail, M. I., Aliasghar, M., Aziz, A., and Hameed, S. (2023). Enhancing cryptocurrency price forecasting accuracy: A feature selection and weighting approach with bi-directional lstm and trend-preserving model bias correction. *IEEE Access*, 11, 65700–65710. Retrieved 2023-08-08, from <https://ieeexplore.ieee.org/document/10156850> doi: 10.1109/ACCESS.2023.3287888
- Raheman, A., Kolonin, A., Fridkins, I., Ansari, I., and Vishwas, M. (2022, 04). Social media sentiment analysis for cryptocurrency market prediction. *arXiv:2204.10185 [cs]*. Retrieved from <https://arxiv.org/abs/2204.10185>
- Raju, S. M., and Tarif, A. M. (2020, 06). Real-time prediction of bitcoin price using machine learning techniques and public sentiment analysis. *arXiv:2006.14473*. Retrieved from <https://arxiv.org/abs/2006.14473>
- Reimers, N., and Gurevych, I. (2019). *Sentence-bert: Sentence embeddings using siamese bert-networks*. Retrieved from <https://arxiv.org/pdf/1908.10084.pdf>
- Ruder, S., Vulić, I., and Søgaard, A. (2019, 08). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65, 569-631. Retrieved from <https://arxiv.org/pdf/1706.04902.pdf> doi: 10.1613/jair.1.11640
- Shleifer, A. (2009). *Inefficient markets: An introduction to behavioral finance*. Oxford Univ. Press.
- Taylor, S. J., and Letham, B. (2017, 09). Forecasting at scale. *The American Statistician*, 72, 37-45. Retrieved from <http://lethalletham.com/ForecastingAtScale.pdf> doi: 10.1080/00031305.2017.1380080
- Thies, S., and Molnár, P. (2018, 12). Bayesian change point analysis of bitcoin returns. *Finance Research Letters*, 27, 223-227. doi: 10.1016/j.frl.2018.03.018
- Valencia, F., Gómez-Espinosa, A., and Valdés-Aguirre, B. (2019, 06). Price movement prediction

of cryptocurrencies using sentiment analysis and machine learning. *Entropy*, 21, 589. doi: 10.3390/e21060589

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). *Attention is all you need*. Retrieved from <https://arxiv.org/abs/1706.03762>

Widianto, M. H., and Cornelius, Y. (2023, 02). Sentiment analysis towards cryptocurrency and nft in bahasa indonesia for twitter large amount data using bert. *International Journal of Intelligent Systems and Applications in Engineering*, 11, 303–309. Retrieved from <https://www.ijisae.org/index.php/IJISAE/article/view/2539/1122>

Wołk, K. (2019, 11). Advanced social media sentiment analysis for short-term cryptocurrency price prediction. *Expert Systems*, 37. doi: 10.1111/exsy.12493

Yoo, S., Jeon, S., Jeong, S., Lee, H., Ryou, H., Park, T., ... Oh, K. (2021, 10). Prediction of the change points in stock markets using dae-lstm. *Sustainability*, 13, 11822. doi: 10.3390/su132111822

Young, T., Hazarika, D., Poria, S., and Cambria, E. (2017). *Recent trends in deep learning based natural language processing*. Retrieved from <https://arxiv.org/abs/1708.02709>

Zhang, G., Patuwo, E., and Hu, M. (1998, 03). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14, 35-62. doi: 10.1016/s0169-2070(97)00044-7

# Appendix A: Disaggregated Results

Table A.1: Average performance by time series model using all NLP features

Model	Excess profit*	Trades**	AUC ROC**	Accuracy**
OLS/Logit	67.48 %	52.7	0.6782	0.8025
XGBoost	126.12 %	44.0	0.6998	0.8057
MLP (FNN)	138.61 %	47.4	0.6797	0.8065
LSTM	83.88 %	12.0	0.6526	0.8028
TFT	11.13 %	4.2	0.5653	0.7971

\* Profit exceeding buy-and-hold strategy

\*\* All metrics are averages of 7-fold cross-validation and were aggregated across target variables

Table A.2: The ten most profitable Bitcoin price movement regression models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
XGBoost	Fine-tuned RoBERTa	351.34 %	70.9	0.5262	0.5458
XGBoost	Twitter-RoBERTa	317.41 %	134.4	0.5202	0.5409
XGBoost	Twitter-RoBERTa + BART MNLI	300.96 %	94.0	0.5285	0.5394
XGBoost	None	216.35 %	186.9	0.5242	0.5360
XGBoost	BART MNLI	201.68 %	90.9	0.5198	0.5357
MLP (FNN)	BART MNLI	172.08 %	142.3	0.5213	0.5379
LSTM	Fine-tuned RoBERTa	159.98 %	22.9	0.5202	0.5400
LSTM	Twitter-RoBERTa	126.24 %	61.1	0.5267	0.5375
LSTM	Twitter-RoBERTa + BART MNLI	107.76 %	12.6	0.5213	0.5403
MLP (FNN)	Fine-tuned RoBERTa	87.10 %	172.0	0.5242	0.5391

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.3: The ten most profitable Ethereum price movement regression models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
MLP (FNN)	Twitter-RoBERTa + BART MNLI	251.98 %	127.4	0.5492	0.5529
MLP (FNN)	Fine-tuned RoBERTa	230.31 %	104.9	0.5445	0.5481
MLP (FNN)	Twitter-RoBERTa	224.38 %	109.4	0.5439	0.5495
MLP (FNN)	None	210.75 %	128.3	0.5445	0.5462
XGBoost	Twitter-RoBERTa + BART MNLI	210.53 %	108.3	0.5333	0.5333
MLP (FNN)	BART MNLI	206.17 %	121.7	0.5466	0.5505
XGBoost	Twitter-RoBERTa	200.77 %	112.9	0.5293	0.5300
XGBoost	Fine-tuned RoBERTa	195.57 %	93.4	0.5254	0.5238
XGBoost	BART MNLI	164.15 %	84.9	0.5314	0.5315
XGBoost	None	163.06 %	93.7	0.5291	0.5314

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.4: The ten most profitable Bitcoin price movement classification models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
LSTM	Twitter-RoBERTa	234.35 %	13.7	0.5253	0.5525
XGBoost	BART MNLI	217.23 %	116.0	0.5438	0.5656
MLP (FNN)	Twitter-RoBERTa	214.29 %	92.3	0.5378	0.5620
MLP (FNN)	Twitter-RoBERTa + BART MNLI	208.54 %	90.6	0.5339	0.5565
XGBoost	Twitter-RoBERTa	203.69 %	143.7	0.5458	0.5696
XGBoost	Twitter-RoBERTa + BART MNLI	202.01 %	157.7	0.5488	0.5659
LSTM	Fine-tuned RoBERTa	193.66 %	17.1	0.5277	0.5519
XGBoost	Fine-tuned RoBERTa	184.13 %	132.6	0.5408	0.5623
MLP (FNN)	None	183.08 %	99.1	0.5354	0.5583
XGBoost	None	177.05 %	61.7	0.5418	0.5653

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.5: The ten most profitable Ethereum price movement classification models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
XGBoost	Fine-tuned RoBERTa	370.77 %	94.9	0.5607	0.5729
MLP (FNN)	BART MNLI	354.81 %	82.9	0.5630	0.5705
MLP (FNN)	Twitter-RoBERTa + BART MNLI	348.69 %	85.7	0.5612	0.5695
MLP (FNN)	Fine-tuned RoBERTa	316.07 %	82.6	0.5660	0.5695
XGBoost	None	311.69 %	103.4	0.5614	0.5705
Logit	BART MNLI	304.98 %	77.1	0.5470	0.5676
Logit	Twitter-RoBERTa + BART MNLI	302.06 %	80.0	0.5586	0.5710
Logit	Twitter-RoBERTa	300.73 %	80.0	0.5650	0.5752
Logit	Fine-tuned RoBERTa	295.88 %	67.7	0.5574	0.5690
XGBoost	Twitter-RoBERTa + BART MNLI	291.43 %	98.6	0.5616	0.5714

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.6: The ten most profitable Bitcoin  $\pm 7$  day extrema classification models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
MLP (FNN)	Fine-tuned RoBERTa	169.12 %	2.3	0.7334	0.9574
MLP (FNN)	BART MNLI	140.25 %	2.6	0.7414	0.9577
MLP (FNN)	None	135.93 %	2.6	0.7409	0.9576
MLP (FNN)	Twitter-RoBERTa + BART MNLI	133.59 %	2.3	0.7396	0.9576
LSTM	Twitter-RoBERTa + BART MNLI	129.85 %	2.3	0.6923	0.9570
MLP (FNN)	Twitter-RoBERTa	127.43 %	2.3	0.7338	0.9574
LSTM	None	126.26 %	2.0	0.6917	0.9570
LSTM	BART MNLI	126.26 %	2.0	0.6851	0.9570
LSTM	Twitter-RoBERTa	126.26 %	2.0	0.6774	0.9586
XGBoost	Twitter-RoBERTa + BART MNLI	64.77 %	2.9	0.7686	0.9574

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.7: The ten most profitable Ethereum  $\pm$  7 day extrema classification models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
LSTM	BART MNLI	82.67 %	2.6	0.6890	0.9562
LSTM	None	74.77 %	2.3	0.6814	0.9564
MLP (FNN)	Twitter-RoBERTa	73.56 %	2.0	0.7221	0.9567
LSTM	Twitter-RoBERTa + BART MNLI	73.52 %	2.0	0.6900	0.9564
MLP (FNN)	Twitter-RoBERTa + BART MNLI	70.50 %	2.0	0.7270	0.9569
MLP (FNN)	Fine-tuned RoBERTa	68.14 %	2.0	0.7230	0.9567
LSTM	Twitter-RoBERTa	68.14 %	2.0	0.6779	0.9562
LSTM	Fine-tuned RoBERTa	67.31 %	2.6	0.6737	0.9564
MLP (FNN)	BART MNLI	49.65 %	2.0	0.7128	0.9564
MLP (FNN)	None	49.41 %	2.0	0.7270	0.9567

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.8: The ten most profitable Bitcoin  $\pm$  14 day extrema classification models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
MLP (FNN)	Fine-tuned RoBERTa	123.68 %	2.9	0.7581	0.9795
MLP (FNN)	Twitter-RoBERTa	71.56 %	2.3	0.7628	0.9794
MLP (FNN)	None	52.34 %	2.6	0.7616	0.9795
MLP (FNN)	Twitter-RoBERTa + BART MNLI	46.92 %	2.6	0.7617	0.9795
XGBoost	Twitter-RoBERTa	31.76 %	2.3	0.8069	0.9795
XGBoost	BART MNLI	31.76 %	2.3	0.7986	0.9799
XGBoost	Fine-tuned RoBERTa	27.28 %	2.3	0.8060	0.9795
XGBoost	None	26.91 %	2.3	0.8031	0.9794
XGBoost	Twitter-RoBERTa + BART MNLI	19.50 %	2.3	0.8000	0.9797
LSTM	BART MNLI	14.02 %	2.3	0.7244	0.9791

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.9: The ten most profitable Ethereum  $\pm$  14 day extrema classification models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
LSTM	BART MNLI	105.21 %	2.3	0.7735	0.9810
MLP (FNN)	Twitter-RoBERTa + BART MNLI	104.91 %	2.9	0.8144	0.9819
MLP (FNN)	Fine-tuned RoBERTa	102.19 %	2.9	0.7998	0.9821
MLP (FNN)	BART MNLI	101.97 %	2.3	0.7957	0.9819
LSTM	Twitter-RoBERTa	95.87 %	2.3	0.7527	0.9814
MLP (FNN)	Twitter-RoBERTa	95.55 %	4.6	0.8159	0.9821
MLP (FNN)	None	69.31 %	2.6	0.8085	0.9826
XGBoost	Twitter-RoBERTa	64.22 %	2.3	0.8325	0.9821
XGBoost	Twitter-RoBERTa + BART MNLI	53.28 %	2.6	0.8334	0.9821
XGBoost	None	48.95 %	2.6	0.8187	0.9824

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.10: The ten most profitable Bitcoin  $\pm$  21 day extrema classification models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
LSTM	Twitter-RoBERTa + BART MNLI	81.72 %	4.0	0.7565	0.9872
MLP (FNN)	BART MNLI	47.08 %	2.3	0.8207	0.9875
MLP (FNN)	Fine-tuned RoBERTa	36.24 %	2.3	0.7944	0.9875
MLP (FNN)	Twitter-RoBERTa + BART MNLI	32.09 %	2.3	0.8012	0.9875
XGBoost	Twitter-RoBERTa + BART MNLI	20.38 %	2.3	0.8487	0.9875
XGBoost	Fine-tuned RoBERTa	19.14 %	2.3	0.8487	0.9875
XGBoost	Twitter-RoBERTa	16.97 %	2.3	0.8517	0.9873
XGBoost	BART MNLI	13.73 %	2.3	0.8567	0.9873
LSTM	BART MNLI	12.57 %	2.3	0.7742	0.9869
MLP (FNN)	Twitter-RoBERTa	11.35 %	2.3	0.7796	0.9873

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.11: The ten most profitable Ethereum  $\pm$  21 day extrema classification models

Model	NLP features	Excess profit*	Trades**	AUC ROC**	Accuracy**
LSTM	Twitter-RoBERTa	121.54 %	2.3	0.8260	0.9857
LSTM	Twitter-RoBERTa + BART MNLI	79.57 %	2.3	0.8014	0.9860
LSTM	BART MNLI	78.61 %	2.0	0.7971	0.9862
MLP (FNN)	Twitter-RoBERTa	75.58 %	3.1	0.8088	0.9855
LSTM	Fine-tuned RoBERTa	75.02 %	2.0	0.8162	0.9862
LSTM	None	68.51 %	2.0	0.8123	0.9862
MLP (FNN)	BART MNLI	55.70 %	2.6	0.7927	0.9862
MLP (FNN)	Twitter-RoBERTa + BART MNLI	51.67 %	2.3	0.8057	0.9862
MLP (FNN)	None	50.79 %	3.4	0.8133	0.9864

\* Profit exceeding buy-and-hold strategy; \*\* All metrics are averages of 7-fold cross-validation

Table A.12: The 50 most influential features for predicting BTC price movements using XGBoost  
 (see Table B.3 for variable definitions)

Feature name*	Normalised total gain	Normalised average gain
tweets_roberta_finetuned_score	0.0754	0.0728
balance_distribution_from_0.01_addressesCount	0.0430	0.0285
reddit_count	0.0346	0.0223
tweets_twitter_roberta_pretrained_score	0.0336	0.0343
indicator_UI	0.0331	0.0278
average_transaction_value	0.0308	0.0156
indicator_AO	0.0291	0.0434
indicator_Ichimoku_A	0.0272	0.0294
balance_distribution_from_0.01_totalVolume	0.0260	0.0181
price_close	0.0235	0.0187
current_supply	0.0204	0.0126
indicator_NVI	0.0200	0.0338
total_volume	0.0198	0.0167
indicator_Ichimoku_Conversion	0.0197	0.0241
gtrends_blockchain_relative_change	0.0191	0.0134
indicator_EMA	0.0189	0.0222
EUR_volumefrom	0.0181	0.0120
zero_balance_addresses_all_time	0.0174	0.0099
indicator_CR	0.0172	0.0148
balance_distribution_from_100.0_addressesCount	0.0170	0.0101
indicator_MACD	0.0165	0.0344
indicator_DCM	0.0164	0.0217
indicator_PPO	0.0160	0.0376
balance_distribution_from_10.0_addressesCount	0.0154	0.0095
reddit_bart_mnli_bullish_score	0.0149	0.0093
indicator_Vortex_down	0.0148	0.0096
indicator_KCW	0.0145	0.0088
unique_addresses_all_time	0.0135	0.0093
large_transaction_count	0.0134	0.0092
indicator_Stoch_RSI	0.0131	0.0096
block_time	0.0128	0.0098
indicator_KCM	0.0126	0.0186
tweet_count	0.0120	0.0097
balance_distribution_from_0.1_totalVolume	0.0119	0.0088
indicator_BBM	0.0110	0.0167
hashrate	0.0110	0.0097
indicator_KAMA	0.0108	0.0104
gold_usd_price	0.0093	0.0083
indicator_BBW	0.0092	0.0049
indicator_Ichimoku_Base	0.0090	0.0171
block_size	0.0089	0.0047
balance_distribution_from_0.0_addressesCount	0.0087	0.0046
EUR_volumeto	0.0086	0.0091
indicator_TRIX	0.0085	0.0178
balance_distribution_from_100.0_totalVolume	0.0076	0.0043
indicator_WMA	0.0076	0.0090
balance_distribution_from_1000.0_totalVolume	0.0076	0.0051
indicator_DPO	0.0075	0.0047
balance_distribution_from_10.0_totalVolume	0.0072	0.0044
balance_distribution_from_1000.0_addressesCount	0.0071	0.0048

\* Features are sorted by total gain

Table A.13: The 50 most influential features for predicting ETH price movements using XGBoost  
 (see Table B.3 for variable definitions)

Feature name*	Normalised total gain	Normalised average gain
tweets_twitter_roberta_pretrained_score	0.0578	0.0440
tweets_roberta_finetuned_score	0.0540	0.0372
news_roberta_finetuned_score	0.0472	0.0476
indicator_CR	0.0397	0.0220
reddit_roberta_finetuned_score	0.0397	0.0247
indicator_EMA	0.0359	0.0519
indicator_Ichimoku_A	0.0302	0.0308
indicator_VWAP	0.0301	0.0355
indicator_DCM	0.0300	0.0345
indicator_BBM	0.0290	0.0492
indicator_MI	0.0278	0.0140
total_issues	0.0264	0.0240
indicator_WilliamsR	0.0252	0.0145
tweets_bart_mnli_bullish_score	0.0233	0.0122
indicator_KAMA	0.0225	0.0279
indicator_Ichimoku_Conversion	0.0223	0.0253
indicator_WMA	0.0222	0.0285
indicator_CMF	0.0211	0.0133
indicator_TRIX	0.0209	0.0480
indicator_Stoch_RSI	0.0193	0.0121
indicator_ROC	0.0188	0.0199
indicator_KCM	0.0179	0.0555
indicator_FI	0.0166	0.0144
closed_issues	0.0135	0.0133
price_close	0.0129	0.0097
average_transaction_value	0.0128	0.0076
reddit_accounts_active_48h	0.0123	0.0046
indicator_ultimate	0.0107	0.0076
indicator_Ichimoku_Base	0.0107	0.0165
exchange_Coinbase_volumefrom	0.0097	0.0057
twitter_followers	0.0097	0.0039
indicator_Ichimoku_B	0.0092	0.0227
unique_addresses_all_time	0.0091	0.0060
indicator_DCW	0.0090	0.0059
staking_rate	0.0089	0.0099
indicator_KST	0.0084	0.0204
exchange_Kraken_volumeto	0.0083	0.0042
zero_balance_addresses_all_time	0.0082	0.0070
indicator_AO	0.0076	0.0092
indicator_EMV	0.0075	0.0048
indicator_VPT	0.0074	0.0066
indicator_DPO	0.0071	0.0051
indicator_Aroon_down	0.0069	0.0071
total_volume	0.0067	0.0038
indicator_CCI	0.0067	0.0070
indicator_Vortex_down	0.0065	0.0045
indicator_MACD	0.0064	0.0128
indicator_Stoch	0.0064	0.0039
USD_volumeto	0.0062	0.0029
exchange_Kraken_volumefrom	0.0062	0.0039

\* Features are sorted by total gain

## Appendix B: Documentation

Table B.1: Search ranges of the hyperparameter optimisation for the RoBERTa fine-tuning

Hyperparameter	Search range
learning rate	$5 \times 10^{-6}$ to 0.05
epochs	2 to 9
batch size	8, 16, 32, 64
warmup steps	0 to 20
L2 regularisation parameter	0.001 to 0.2

Table B.2: Search ranges of the hyperparameter optimisation for the time series models

Model	Hyperparameter	Search range
Ridge Regression	L2 regularisation parameter solver	0.001 to 100 SVD, Cholesky, LSQR, Sparse CG, SAG, SAGA
Logistic Regression	L2 regularisation parameter solver	0.0005 to 1000 L-BFGS, Lblinear, Newton-CG, Newton-Cholesky, SAG, SAGA
XGBoost	number of estimators	100 to 1400
	max depth	1 to 20
	learning rate	0.01 to 0.3
	subsampling ratio of instances	0.5 to 1
	subsampling ratio of features	0.5 to 1
	L1 regularisation parameter	0.001 to 1
	L2 regularisation parameter	0.001 to 1
	partitioning threshold (gamma)	0 to 1
MLP (FNN)	number of layers	1 to 4
	size of each layer*	10 to 200
	activation function	identity (linear), logistic, hyperbolic tangent, ReLU
	optimiser	L-BFGS, SGD, Adam
	L2 regularisation parameter	0.0001 to 0.1
	learning rate	0.001 to 0.1
	scaling	none, standardisation, min-max scaling
	epochs	10 to 1000
	batch size	16, 32, 64, 128
LSTM	number of LSTM layers	1 to 3
	size of each LSTM layer*	50 to 300
	number of dense layers	0 to 3
	size of each dense layer*	10 to 150
	activation function	hyperbolic tangent, ReLU
	dropout	0.1 to 0.5
	optimiser	Adam, RMSprop, SGD
	learning rate	0.0001 to 0.1
	scaling	none, standardisation, min-max scaling
	epochs	10 to 200
	batch size	32, 64, 128, 256
TFT	number of LSTM layers	1 to 3
	number of attention heads	4, 8, 16
	size of variable selection GRNs	16, 32, 64, 128
	size of remaining layers**	16, 32, 64, 128
	dropout	0.1 to 0.5
	learning rate	$5 \times 10^{-5}$ to 0.01
	optimiser	Adam, RMSprop, SGD, Adagrad, Ranger
	gradient clipping value	0.1 to 1.0
	limit_train_batches	0.8 to 1.0
	reduce_on_plateau_patience	5, 10, 15
	epochs	1 to 200
	batch size	16, 32, 64, 128, 256

\* The number of neurons was tuned individually for each layer, not set uniformly for all

\*\* The number of neurons was set uniformly for all layers

Table B.3: Overview and description of the Bitcoin/Ethereum features

Variable name	Source	Interval	I*	Description
price_close	CryptoCompare	timepoint	1	BTC (ETH) market value in EUR calculated with the CCCAGG method (weighted average of EUR prices of 301 exchanges – weighted by exchange volume and time since last trade)
total_volume	CoinGecko	24h	1	Total value in EUR of BTC (ETH) that has been bought and sold on the spot market on 639 exchanges
news_bart_mnli_bullish_score	Google News / Own calculation	24h	0	Average BART MNLI bullish score of news
tweets_bart_mnli_bullish_score	Twitter / Own calculation	24h	0	Average BART MNLI bullish score of Twitter posts
reddit_bart_mnli_bullish_score	Reddit / Own calculation	24h	0	Average BART MNLI bullish score of Reddit posts
news_twitter_roberta_pretrained_score	Google News / Own calculation	24h	0	Average Twitter-RoBERTa sentiment score of news
tweets_twitter_roberta_pretrained_score	Twitter / Own calculation	24h	0	Average Twitter-RoBERTa sentiment score of Twitter posts
reddit_twitter_roberta_pretrained_score	Reddit / Own calculation	24h	0	Average Twitter-RoBERTa sentiment score of Reddit posts
news_roberta_finetuned_score	Google News / Own calculation	24h	0	Average Finetuned RoBERTa score of news
tweets_roberta_finetuned_score	Twitter / Own calculation	24h	0	Average Finetuned RoBERTa score of Twitter posts
reddit_roberta_finetuned_score	Reddit / Own calculation	24h	0	Average Finetuned RoBERTa score of Reddit posts
news_count	Google News	24h	1	Number of news articles from CoinDesk, Cointelegraph or Decrypt for the keywords Bitcoin, BTC (Ethereum, ETH)
tweet_count	Twitter	24h	1	Number of tweets containing hashtags #bitcoin or #btc (#ethereum or #eth)
twitter_followers	Twitter	timepoint	2	Count of followers of the twitter account @Bitcoin (@ethereum)
reddit_count	Reddit	24h	1	Number of Reddit posts on r/Bitcoin (r/ethereum)
reddit_subscribers	Reddit	timepoint	2	Count of subscribers to the subreddit r/Bitcoin (r/ethereum)
reddit_accounts_active_48h	Reddit	48h	1	Count of reddit accounts active on the subreddit r/Bitcoin (r/ethereum)
forks	GitHub	timepoint	2	Number of forks on the bitcoin/bitcoin (ethereum/go-ethereum) GitHub repository
stars	GitHub	timepoint	2	Number of stars on the GitHub repository
subscribers	GitHub	timepoint	2	Number of watchers of the GitHub repository
total_issues	GitHub	timepoint	2	Number of open and closed issues of the GitHub repository
closed_issues	GitHub	timepoint	2	Number of closed issues of the GitHub repository
pull_requests_merged	GitHub	timepoint	2	Number of merged pull requests of the GitHub repository
pull_request_contributors	GitHub	timepoint	2	Number of pull request contributors of the GitHub repository
additions	GitHub	24h	1	Number of additions on the GitHub repository
deletions	GitHub	24h	1	Number of deletions on the GitHub repository

\* Order of integration (number of times the time series had to be differenced to become stationary)

Table B.3 continued

Variable name	Source	Interval	I*	Description
commit_count_4_weeks	GitHub	4 weeks	1	Number of commits in the past 4 weeks on the GitHub repository
ETH_volumefrom (BTC_volumefrom)	CryptoCompare	24h	1	Volume of transactions from ETH to BTC (BTC to ETH) across 301 exchanges
ETH_volumeto (BTC_volumeto)	CryptoCompare	24h	1	Volume of transactions from BTC to ETH (ETH to BTC) across 301 exchanges
USD_volumefrom	CryptoCompare	24h	1	Volume of transactions from USD to BTC (ETH) across 301 exchanges
USD_volumeto	CryptoCompare	24h	1	Volume of transactions from BTC (ETH) to USD across 301 exchanges
EUR_volumefrom	CryptoCompare	24h	1	Volume of transactions from EUR to BTC (ETH) across 301 exchanges
EUR_volumeto	CryptoCompare	24h	1	Volume of transactions from BTC (ETH) to EUR across 301 exchanges
exchange_Bitfinex_volumeto	CryptoCompare	24h	1	Inflow of BTC (ETH) on Bitfinex exchange
exchange_Bitfinex_volumefrom	CryptoCompare	24h	1	Outflow of BTC (ETH) on Bitfinex exchange
exchange_Bitfinex_volumetotal	CryptoCompare	24h	1	Total BTC (ETH) cashflows on Bitfinex exchange
exchange_Kraken_volumeto	CryptoCompare	24h	1	Inflow of BTC (ETH) on Kraken exchange
exchange_Kraken_volumefrom	CryptoCompare	24h	1	Outflow of BTC (ETH) on Kraken exchange
exchange_Kraken_volumetotal	CryptoCompare	24h	1	Total BTC (ETH) cashflows on Kraken exchange
exchange_Coinbase_volumeto	CryptoCompare	24h	1	Inflow of BTC (ETH) on Coinbase exchange
exchange_Coinbase_volumefrom	CryptoCompare	24h	1	Outflow of BTC (ETH) on Coinbase exchange
exchange_Coinbase_volumetotal	CryptoCompare	24h	1	Total BTC (ETH) cashflows on Coinbase exchange
exchange_BTSE_volumeto	CryptoCompare	24h	1	Inflow of BTC (ETH) on BTSE exchange
exchange_BTSE_volumefrom	CryptoCompare	24h	1	Outflow of BTC (ETH) on BTSE exchange
exchange_BTSE_volumetotal	CryptoCompare	24h	1	Total BTC (ETH) cashflows on BTSE exchange
exchange_Binance_volumeto	CryptoCompare	24h	1	Inflow of BTC (ETH) on Binance exchange
exchange_Binance_volumefrom	CryptoCompare	24h	1	Outflow of BTC (ETH) on Binance exchange
exchange_Binance_volumetotal	CryptoCompare	24h	1	Total BTC (ETH) cashflows on Binance exchange
zero_balance_addresses_all_time	IntoTheBlock	timepoint	2	Amount of BTC (ETH) addresses that have always had zero balance since inception
unique_addresses_all_time	IntoTheBlock	timepoint	2	Amount of BTC (ETH) addresses that executed at least one transaction since inception
new_addresses	IntoTheBlock	24h	1	Amount of new BTC (ETH) addresses created
active_addresses	IntoTheBlock	24h	1	Amount of BTC (ETH) addresses that executed at least one transaction
transaction_count	IntoTheBlock	24h	1	Number of valid transactions on the BTC (ETH) blockchain
large_transaction_count	IntoTheBlock	24h	1	Number of valid transactions greater than 100,000 USD on the BTC (ETH) blockchain
average_transaction_value	IntoTheBlock	24h	1	Average transaction value on the BTC (ETH) blockchain in BTC (ETH)
hashrate (only Bitcoin)	IntoTheBlock	24h	1	Number of terahashes per second the BTC network is performing
difficulty (only Bitcoin)	IntoTheBlock	24h	1	Mean difficulty of finding a hash that meets the protocol-designated requirement (difficulty is adjusted every 2016 blocks so that the average time between each block remains ~10 minutes)
block_time (only Bitcoin)	IntoTheBlock	24h	1	Average time in seconds it takes miners to verify transactions within one block on the BTC network

Table B.3 continued

Variable name	Source	Interval	I*	Description
block_size	IntoTheBlock	24h	1	Average block size in bytes on the BTC (ETH) blockchain
current_supply	IntoTheBlock	timepoint	2	Sum of all BTC (ETH) issued on the BTC (ETH) ledger
staking_rate (only Ethereum)	Attestant	24h	2	ETH staking yield (1 year ROI of staking ETH) offered by Attestant
balance_distribution_from_0.0_totalVolume	IntoTheBlock	timepoint	1	Total amount of BTC (ETH) held by addresses with a balance between 0 and 0.001 BTC (ETH)
balance_distribution_from_0.001_totalVolume	IntoTheBlock	timepoint	1	Total amount of BTC (ETH) held by addresses with a balance between 0.001 and 0.01 BTC (ETH)
balance_distribution_from_0.01_totalVolume	IntoTheBlock	timepoint	1	Total amount of BTC (ETH) held by addresses with a balance between 0.01 and 0.1 BTC (ETH)
balance_distribution_from_0.1_totalVolume	IntoTheBlock	timepoint	1	Total amount of BTC (ETH) held by addresses with a balance between 0.1 and 1 BTC (ETH)
balance_distribution_from_1.0_totalVolume	IntoTheBlock	timepoint	1	Total amount of BTC (ETH) held by addresses with a balance between 1 and 10 BTC (ETH)
balance_distribution_from_10.0_totalVolume	IntoTheBlock	timepoint	1	Total amount of BTC held by addresses with a balance between 10 and 100 BTC
balance_distribution_from_100.0_totalVolume	IntoTheBlock	timepoint	1	Total amount of BTC (ETH) held by addresses with a balance between 100 and 1000 BTC (ETH)
balance_distribution_from_1000.0_totalVolume	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 1000 and 10000 BTC (ETH)
balance_distribution_from_10000.0_totalVolume	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 10000 and 100000 BTC (ETH)
balance_distribution_from_100000.0_totalVolume	IntoTheBlock	timepoint	1	Total amount of addresses with a balance above 100000 BTC (ETH)
balance_distribution_from_0.0_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 0 and 0.001 BTC (ETH)
balance_distribution_from_0.001_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 0.001 and 0.01 BTC (ETH)
balance_distribution_from_0.01_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 0.01 and 0.1 BTC (ETH)
balance_distribution_from_0.1_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 0.1 and 1 BTC (ETH)
balance_distribution_from_1.0_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 1 and 10 BTC (ETH)
balance_distribution_from_10.0_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 10 and 100 BTC (ETH)
balance_distribution_from_100.0_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 100 and 1000 BTC (ETH)
balance_distribution_from_1000.0_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 1000 and 10000 BTC (ETH)
balance_distribution_from_10000.0_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance between 10000 and 100000 BTC (ETH)
balance_distribution_from_100000.0_addressesCount	IntoTheBlock	timepoint	1	Total amount of addresses with a balance above 100000 BTC (ETH)
index_MVDA_close	CryptoCompare	timepoint	1	Close value of the MarketVector Digital Assets 100 (MVDA) index tracking the 100 largest digital assets
index_BVIN_close	CryptoCompare	timepoint	1	Close value of the CryptoCompare Bitcoin Volatility (BVIN) index tracking BTC implied volatility using options data from Deribit
gtrends_bitcoin_relative_change	Google Trends	24h	1	Percentage change in amount of Google searches of the keyword “bitcoin”
gtrends_ethereum_relative_change	Google Trends	24h	1	Percentage change in amount of Google searches of the keyword “ethereum”

Table B.3 continued

Variable name	Source	Interval	I*	Description
gtrends_cryptocurrency_relative_change	Google Trends	24h	1	Percentage change in amount of Google searches of the keyword “cryptocurrency”
gtrends_blockchain_relative_change	Google Trends	24h	1	Percentage change in amount of Google searches of the keyword “blockchain”
gtrends_investing_relative_change	Google Trends	24h	1	Percentage change in amount of Google searches of the keyword “investing”
sp500_price	Yahoo Finance	timepoint	1	Close value of the S&P 500 index
sp500_volume	Yahoo Finance	24h	1	Volume of the S&P 500 index
vix	Yahoo Finance	timepoint	1	Close value of the CBOE Volatility Index (VIX) tracking the S&P 500 implied volatility
gold_usd_price	Yahoo Finance	timepoint	1	Close value of the COMEX gold future (GC)
indicator_AO	Own calculation	timepoint	1	Awesome Oscillator (AO) - Measures market momentum to capture the potential change in trend
indicator_KAMA	Own calculation	timepoint	1	Kaufman's Adaptive Moving Average (KAMA) - A moving average that adjusts its length based on market volatility
indicator_PPO	Own calculation	timepoint	1	Percentage Price Oscillator (PPO) - Measures the difference between two moving averages as a percentage of the larger moving average
indicator_PVO	Own calculation	timepoint	1	Percentage Volume Oscillator (PVO) - Like PPO but for volume, it measures the difference between two volume moving averages
indicator_ROC	Own calculation	timepoint	1	Rate of Change (ROC) - Measures the percentage change in price from one period to the next
indicator_RSI	Own calculation	timepoint	1	Relative Strength Index (RSI) - Measures the speed and change of price movements and indicates overbought or oversold conditions
indicator_Stoch_RSI	Own calculation	timepoint	1	Stochastic RSI - Combines stochastic oscillator and RSI to measure the RSI relative to its high-low range
indicator_Stoch	Own calculation	timepoint	1	Stochastic Oscillator - Compares a closing price to its price range over a specific time period
indicator_TSI	Own calculation	timepoint	1	True Strength Index (TSI) - Measures the momentum of price movements
indicator_ultimate	Own calculation	timepoint	1	Ultimate Oscillator - Combines short, medium, and long-term price action into one oscillator to avoid false divergences
indicator_WilliamsR	Own calculation	timepoint	1	Williams %R - A momentum indicator that measures overbought/oversold levels
indicator_ADI	Own calculation	timepoint	1	Accumulation/Distribution Index (ADI) - Measures the cumulative flow of money into and out of a security
indicator_CMF	Own calculation	timepoint	1	Chaikin Money Flow (CMF) - Measures the amount of Money Flow Volume over a specific period
indicator_EMV	Own calculation	timepoint	1	Ease of Movement (EMV) - Relates volume and price change to show how much volume is needed to move prices
indicator_FI	Own calculation	timepoint	1	Force Index (FI) - Measures the buying or selling pressure over a specific period
indicator_MFI	Own calculation	timepoint	1	Money Flow Index (MFI) - A volume-weighted version of RSI that shows price strength
indicator_NVI	Own calculation	timepoint	1	Negative Volume Index (NVI) - Focuses on days where the volume decreases from the previous day
indicator_OBV	Own calculation	timepoint	1	On-Balance Volume (OBV) - Relates volume to price change
indicator_VPT	Own calculation	timepoint	1	Volume Price Trend (VPT) - Combines price and volume to show the direction of price trend

Table B.3 continued

Variable name	Source	Interval	I*	Description
indicator_VWAP	Own calculation	timepoint	1	Volume Weighted Average Price (VWAP) - The average price weighted by volume
indicator_BBM	Own calculation	timepoint	1	Bollinger Middle Band - The middle band in the Bollinger Bands, which is a simple moving average
indicator_BBW	Own calculation	timepoint	1	Bollinger Bandwidth - The width of the Bollinger Bands
indicator_DCM	Own calculation	timepoint	1	Donchian Channel Middle Band - The average of the Donchian high and low bands
indicator_DCW	Own calculation	timepoint	1	Donchian Channel Width - The width of the Donchian Bands
indicator_KCM	Own calculation	timepoint	1	Keltner Channel Middle Band - The average of the Keltner high and low bands
indicator_KCW	Own calculation	timepoint	1	Keltner Channel Width - The width of the Keltner Bands
indicator_UI	Own calculation	timepoint	1	Ulcer Index (UI) - Measures downside risk in terms of price declines
indicator_Aroon_down	Own calculation	timepoint	1	Aroon Down - Identifies the number of days since a 25-day low
indicator_Aroon_up	Own calculation	timepoint	1	Aroon Up - Identifies the number of days since a 25-day high
indicator_CCI	Own calculation	timepoint	1	Commodity Channel Index (CCI) - Measures the difference between a security's price change and its average price change.
indicator_DPO	Own calculation	timepoint	1	Detrended Price Oscillator (DPO) - Removes trend from price.
indicator_EMA	Own calculation	timepoint	1	Exponential Moving Average (EMA) - A moving average that gives more weight to recent prices.
indicator_Ichimoku_A, indicator_Ichimoku_B, indicator_Ichimoku_Base, indicator_Ichimoku_Conversion	Own calculation	timepoint	1	Ichimoku Cloud - A collection of technical indicators that show support and resistance levels, as well as momentum and trend direction
indicator_KST	Own calculation	timepoint	1	Know Sure Thing (KST) - A momentum oscillator based on the smoothed rate-of-change for four different time frames
indicator_MACD	Own calculation	timepoint	1	Moving Average Convergence Divergence (MACD) - Shows the relationship between two moving averages of a security's price
indicator_MACD_Signal	Own calculation	timepoint	1	MACD Signal - A signal line for the MACD
indicator_MI	Own calculation	timepoint	1	Mass Index (MI) - Measures the volatility of price changes
indicator_TRIX	Own calculation	timepoint	1	TRIX - Shows the percent rate of change of a triple exponentially smoothed moving average
indicator_Vortex_down	Own calculation	timepoint	1	Vortex Indicator - Identifies the start of a new trend or the continuation of a current trend
indicator_Vortex_up	Own calculation	timepoint	1	Vortex Indicator - Identifies the start of a new trend or the continuation of a current trend
indicator_WMA	Own calculation	timepoint	1	Weighted Moving Average (WMA) - A moving average where more recent prices are given more weight
indicator_CR	Own calculation	timepoint	1	Cumulative Return (CR) - Measures the total return of a stock over a set period
indicator_PSAR_down	Own calculation	timepoint	0	Parabolic SAR Down - Provides exit points
indicator_PSAR_up	Own calculation	timepoint	0	Parabolic SAR Up - Provides entry points

# Declaration of Academic Honesty

I, Vincent B. Gurgul, hereby declare that I have not previously submitted the present work for other examinations. I wrote this work independently. All sources, including sources from the Internet, that I have reproduced in either an unaltered or modified form (particularly sources for texts, graphs, tables, and images), have been acknowledged by me as such. I understand that violations of these principles will result in proceedings regarding deception or attempted deception.

---

Vincent B. Gurgul  
Berlin, September 14, 2023