International Journal of Forecasting 41 (2025) 1666–1695

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Deep learning and NLP in cryptocurrency forecasting: Integrating financial, blockchain, and social media data

Vincent Gurgul [a,*], Stefan Lessmann [a,b], Wolfgang Karl Härdle [a]

[a] Humboldt-Universität zu Berlin, Unter den Linden 6, 10117, Berlin, Germany
[b] Bucharest University of Economic Studies, Piata Romană 8, 010374, Bucharest, Romania

ARTICLE INFO

Keywords:
Cryptocurrency price forecasting
Machine learning
Deep learning
Natural language processing
Market sentiment analysis
Social media analysis

ABSTRACT

We introduce novel approaches to cryptocurrency price forecasting, leveraging Machine Learning (ML) and Natural Language Processing (NLP) techniques, with a focus on Bitcoin and Ethereum. By analysing news and social media content, primarily from Twitter and Reddit, we assess the impact of public sentiment on cryptocurrency markets. A distinctive feature of our methodology is the application of the BART MNLI zero-shot classification model to detect bullish and bearish trends, significantly advancing beyond traditional sentiment analysis. Additionally, we systematically compare a range of pre-trained and fine-tuned deep learning NLP models against conventional dictionary-based sentiment analysis methods. Another key contribution of our work is the adoption of local extrema alongside daily price movements as predictive targets, reducing trading frequency and portfolio volatility. Our findings demonstrate that integrating textual data into cryptocurrency price forecasting not only improves forecasting accuracy but also consistently enhances the profitability and Sharpe ratio across various validation scenarios, particularly when applying deep learning NLP techniques. The entire codebase of our experiments is available via an online repository: https://anonymous.4open.science/r/crypto-forecasting-public.

1. Introduction

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

The research presented in this work examines the impact of news from various cryptocurrency-related outlets and social media posts from Twitter and Reddit on the valuations of BTC and ETH, the two largest cryptocurrencies by market capitalisation. Traditionally, studies in the cryptocurrency domain have leaned on dictionary-based methods to analyse the influence of news and social media. However, with the advancements in linguistic AI models, there is an opportunity to explore new avenues for sentiment analysis. Our research extends beyond traditional techniques by incorporating deep learning NLP methods to gauge market sentiment. While the application of deep learning in sentiment analysis is established, our work distinguishes itself by adopting a zero-shot classification language model specifically to differentiate between bullish and bearish market perspectives, a nuanced approach that moves beyond general sentiment classification to provide more targeted insights into market dynamics.

[*] Corresponding author.
E-mail address: vincent.gurgul@hu-berlin.de (V. Gurgul).

https://doi.org/10.1016/j.ijforecast.2025.02.007

We also advance beyond established practices in our price forecasting methodology. In the realm of cryptocurrency analysis, this is usually regression onto price changes or the binary classification of daily price movement (upward or downward). Local extrema remain unexplored as a target variable, even though their inherent lower noise levels offer a significant advantage in the highly volatile cryptocurrency market. We conjecture that by classifying local minima and maxima, our ML models obtain enhanced prediction performance, with respect to both classification metrics and profitability. In addition to the forecasting of daily price movements, we therefore aim to establish whether the textual data can aid in the prediction of local extreme points with various observational time frames.

We compare the predictive performance of multiple 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 and includes a trading simulation.

By exploring the evolution of market efficiency, our research traces the adaptation of markets to the increasing prominence of social media discourse over time, dating back to 2012. Our comprehensive approach, which also includes on-chain data, GitHub, and Google Trends, enables us to reveal the nuanced ways in which information dissemination affects cryptocurrency market behaviour.

2. Literature review

Since 2015, there has been sustained interest in leveraging information from social media for forecasting cryptocurrency prices. This growing body of research has even led to the publication of review papers on the subject. To provide a comprehensive overview of the literature, we summarise the key findings from these reviews and conduct an analysis of significant individual studies. In the latter, we include papers that predict cryptocurrency prices or associated target variables using NLP data. Studies that do not make use of textual data in their modelling pipeline, or that solely focus on forecasting volatility or examining statistical properties of the time series such as change points, are not taken into account.

In the following subsections, we dissect the forecasting methods, the NLP approaches, the target variable selection, and the variety of explanatory variables considered—aiming to establish a thorough understanding of how the field has developed, and to identify prevailing trends and avenues for further exploration. The results are summarised in Table 1.

2.1. Evolution of forecasting techniques

The forecasting methods are categorised into linear, non-linear, and sequential models. Linear models, such as exponential smoothing, autoregressive moving averages, Ordinary Least Squares (OLS), and support vector machines are valued for their simplicity and interpretability. According to the review by Fang et al. (2022), OLS regression not only formed the backbone of cryptocurrency forecasting in its beginnings, but is still the most used forecasting method in the literature.

However, the landscape of financial analytics evolved, gradually shifting towards more sophisticated methods (Zhang et al., 2024). Non-linear machine learning approaches are capable of capturing more intricate interactions between the variables, yet they are less interpretable due to their complex internal structures. Multiple studies systematically evaluate various non-linear model types, such as random forests, AdaBoost, gradient boosting, and MLPs in the context of cryptocurrency forecasting. These studies frequently highlight the superior performance of gradient boosting and MLP neural networks (Valencia et al., 2019; Wołk, 2019).

Sequential models, such as RNNs and transformers, excel at handling ordered time-series data prevalent in financial contexts. The review by Zhang et al. (2024) reveals that many state-of-the-art neural network architectures are applied to cryptocurrencies, ranging from LSTM and Gated Recurrent Unit (GRU) networks to CNNs. Yet only 14% of cryptocurrency forecasting papers make use of neural networks at all, with sequential models accounting for only 4% (Fang et al., 2022). Pant et al. (2018), Mittal et al. (2019), Raju and Tarif (2020), and Kim et al. (2023) achieve very good results with standard RNNs and LSTM networks. Ortu et al. (2022) additionally implement CNNs and Multivariate Attention Long Short Term Memory Fully Convolutional Networks (MALSTM-FCNs). Parekh et al. (2022) build an RNN-based model that integrates sequential and non-sequential data side by side.

Even less widespread than standard RNNs is the application of transformer-based time series models like the TFT, with (Murray et al., 2023) representing the only exploration to date. Our study aims to bridge this gap by further investigating the TFT alongside OLS regression, gradient boosting, MLPs, and LSTM networks.

2.2. Application of NLP methods

NLP methods are crucial for understanding market sentiment, a significant driver of cryptocurrency price fluctuations. The review by Fang et al. (2022) determines two main approaches to utilising textual data in this context: 1) assigning a sentiment score to unlabelled text data using an unsupervised method, such as a sentiment dictionary or a pre-trained transformer, and 2) labelling the text data (either manually or using the price movements, assuming that a price increase implies a positive market sentiment) and subsequently taking a supervised learning approach to training a machine learning model, such as an RNN. However, there are also intermediate approaches that do not fit neatly into either category, for example, using pre-trained (unsupervised) embeddings and feeding the results into a supervised numerical model, or fine-tuning a pre-trained transformer. Instead of the training approach, therefore, we divide the literature based on the language model architectures, which can be classified into dictionary-based approaches, embeddings, RNNs, and transformers.

Dictionary-based approaches assign sentiment scores to words based on pre-defined, often manually labelled

Table 1
Overview of cryptocurrency price forecasting approaches that utilise NLP.

Paper	Forecasting Methods			NLP Methods				Targets			Features							Cryptocurrencies[a]
	Linear Models	Non-linear Models	Sequential Models	Dictionary	Embeddings	RNN	Transformers	Regression	Classification	Local extrema	Financial	Blockchain	News	Twitter	Reddit	GitHub	Google Trends	
Colianni et al. (2015)	✓	✓		✓					✓					✓				b
Abraham et al. (2018)	✓			✓				✓						✓			✓	be
Jain et al. (2018)	✓			✓				✓						✓				bl
Karalevicius et al. (2018)				✓					✓				✓					b
Pant et al. (2018)			✓		✓			✓						✓				b
Chen et al. (2019)	✓			✓				✓						✓[b]	✓			i
Hao et al. (2019)		✓		✓				✓						✓			✓	b
Inamdar et al. (2019)		✓				✓		✓					✓	✓				b
Li et al. (2019)		✓		✓				✓			✓			✓				z
Mittal et al. (2019)	✓	✓	✓	✓				✓				✓		✓			✓	b
Valencia et al. (2019)	✓	✓		✓					✓		✓			✓				berl
Wołk (2019)	✓	✓		✓				✓			✓	✓		✓			✓	bermnc
Raju and Tarif (2020)	✓		✓	✓				✓			✓			✓				b
Ider and Lessmann (2022)	✓	✓					✓	✓	✓		✓	✓	✓	✓	✓			be
Ortu et al. (2022)		✓	✓				✓		✓		✓				✓	✓		be
Parekh et al. (2022)			✓	✓				✓						✓				bld
Kim et al. (2023)		✓	✓				✓		✓		✓		✓					b
Subramanian et al. (2024)	✓		✓	✓				✓					✓	✓				b
The proposed approach	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	be

[a] b = Bitcoin, e = Ethereum, l = Litecoin, m = Monero, z = ZClassic, n = Electroneum, c = ZCash, d = Dash, i = CRIX index.
[b] (Chen et al., 2019) leverage the investment-specific platform StockTwits instead of Twitter.

dictionaries. While computationally efficient, they fail to capture long-term dependencies and contextual nuances (Liu, 2012). According to Fang et al. (2022), dictionary-based sentiment analysis is highly prevalent in cryptocurrency price forecasting with the most commonly employed dictionaries being the general-purpose emotional valence libraries VADER and Textblob. In addition to sentiment scores for each word, those two consider a list of negators and intensifiers (words that reverse or amplify/weaken sentiment). Valencia et al. (2019) report significant improvements in Bitcoin and Litecoin price forecasting accuracy when including VADER sentiment scores from Twitter data. Some studies make use of domain-specific lexicons, such as (Karalevicius et al., 2018), who compare the Harvard Psychosocial and Loughran–McDonald finance-specific dictionaries, and Chen et al. (2019), who develop a cryptocurrency-specific lexicon using posts from the StockTwits platform, where users themselves label their posts as bullish or bearish.

Embeddings offer a more sophisticated mechanism for capturing semantic meanings by encoding words as high-dimensional vectors (Mikolov et al., 2013). Those vectors can then be fed as a numeric input into a standard machine learning model. Pant et al. (2018) leverage the pre-trained Gensim Word2Vec embeddings alongside a bag-of-words approach to include textual data from Twitter in their cryptocurrency forecasting pipeline.

RNNs have the capability to model contextual relationships over sequences, making them suitable for capturing dependencies in text (Graves, 2012). Inamdar et al. (2019) train an LSTM network on tweets and news articles labelled with historical Bitcoin price data and subsequently feed those features into a random forests model for future price prediction.

Transformer architectures represent the cutting edge of NLP by excelling at capturing complex long-range dependencies in language, which enables them to model intricate grammatical and logical patterns (Vaswani et al., 2017). Ortu et al. (2022) employ a pre-trained BERT model to categorise emotions and sentiments in cryptocurrency-related comments on GitHub and Reddit, analysing both sentiment polarity and specific emotional reactions, such as love, joy, anger, or sadness. They report a significant improvement in forecasting accuracy on hourly data when including the sentiment scores. Kim et al. (2023) fine-tune BERT, DeBERTa, and RoBERTa models on a manually labelled cryptocurrency-specific corpus of news articles and compare their performance to pre-trained mBERT and XLM-RoBERTa models. They show that the fine-tuned models slightly outperform the pre-trained models, with the fine-tuned RoBERTa taking the lead. Furthermore, they state that including NLP data resulted in a 3% increase in accuracy and 20% increase in profits with no change in volatility.

Outside of price forecasting, there exists a broader corpus of literature on advanced sentiment analysis methods in the realm of cryptocurrencies (Dwivedi, 2023; Widianto & Cornelius, 2023), but their potential for improving predictive modelling remains underexplored. We aim to address this gap by further exploring Transformer models, fine-tuning them, and introducing a novel approach for converting textual data into numerical representations of market sentiment. The VADER dictionary is included as a benchmark due to its proven effectiveness in previous research.

2.3. Target and feature selection

Literature predominantly treats price forecasting as a regression problem, predicting the next period's price or its relative change (Fang et al., 2022). Classification approaches, which predict price increases or decreases, are less common but have demonstrated superior performance in terms of both classification metrics and trading profit (Leung et al., 2000; Mudassir et al., 2020). The prediction of local extreme points, however, remains unexplored in cryptocurrencies. Recognising the potential of discrete targets, our research expands beyond price regression, incorporating classification and, distinctively, the forecasting of local extrema.

Fang et al. (2022) observe a diverse range of predictors, ranging from financial data (indicators and other assets) to blockchain data (information contained on the cryptocurrencies' ledgers, e.g. transaction and balance records), textual data (news outlets, Twitter, Reddit), GitHub data, and Google Trends. Initially, research focused on autoregressive analyses and sentiment data from Twitter and news outlets. Early on, Google Trends, which refers to the volume of searches for specific related keywords on Google, were integrated as a feature in various approaches. Fang et al. (2022) highlight that many papers reported a significant relationship between Google Trends and cryptocurrency price movements. More recently, however, there was a trend towards using blockchain metrics, transaction data, and Reddit discussions. Our analysis employs a comprehensive set of explanatory variables to evaluate the relative merit of these diverse data sources.

3. Methodology

Building on the comprehensive review of prior literature, this section delves into the methodological framework of our study. We begin by exploring the realm of time series modelling through the lens of neural networks, shedding light on their respective utilities and the advancements they have introduced in modelling financial time series data. Subsequently, we present an extensive overview of neural network-based NLP methods, beginning with embeddings and advancing to more sophisticated techniques such as RNNs and transformer models. In this context, we introduce and explain the three deep learning models at the forefront of our textual analysis: (i) Twitter-RoBERTa, a sentiment analysis model trained specifically on social media data; (ii) BART MNLI, a zero-shot classification model that we tailor for gauging bullishness in financial narratives; and (iii) a vanilla RoBERTa model, fine-tuned on our targets. Finally, our discourse shifts to the rationale behind our target variable selection and our trading strategy. Here, we elaborate on our choice of targets, detail the process of creating local extreme point targets, and articulate the principles that shape our approach to market entry and exit.

3.1. Time series modelling with neural networks

Traditional statistical methods, such as Autoregressive Integrated Moving Average (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 et al., 1998). This has led research in the realm of financial analysis to turn increasingly to deep learning methodologies for price forecasting (Ozbayoglu et al., 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 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 encapsulates historical lags and their intricate relationships (see Fig. 1). 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.

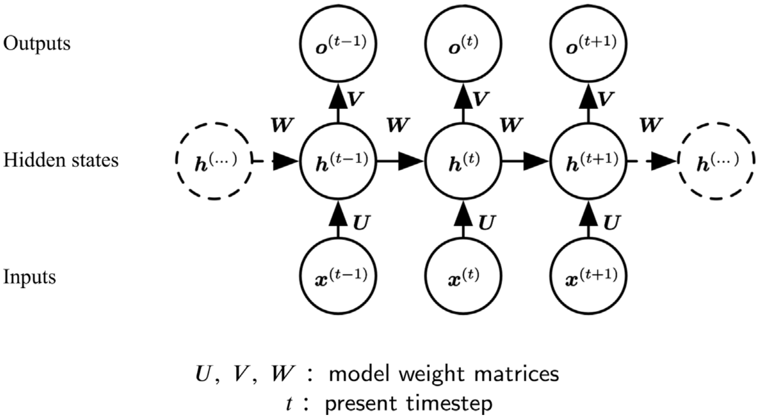

Fig. 1. The standard RNN model architecture.
Source: Adapted from Goodfellow et al. (2016)

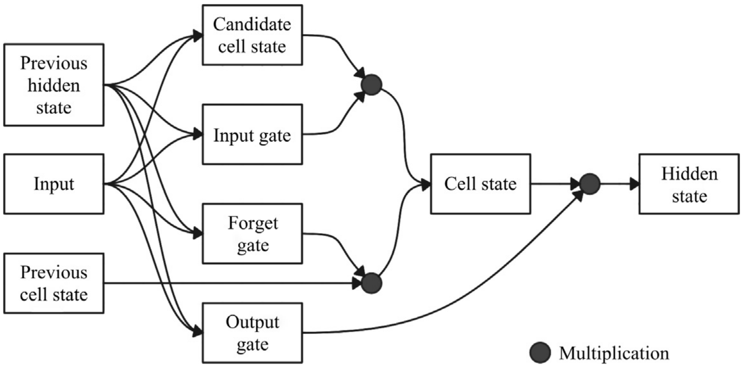

Fig. 2. LSTM cell architecture.

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).

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 et al. (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 (see Fig. 2).

Fig. 3 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.

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).

Another neural network architecture that is applicable in time series analysis is the CNN (see Fig. 4). 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 et al., 2019).

The TFT, introduced by Lim et al. (2020), is a notable evolution in time series modelling, blending aspects

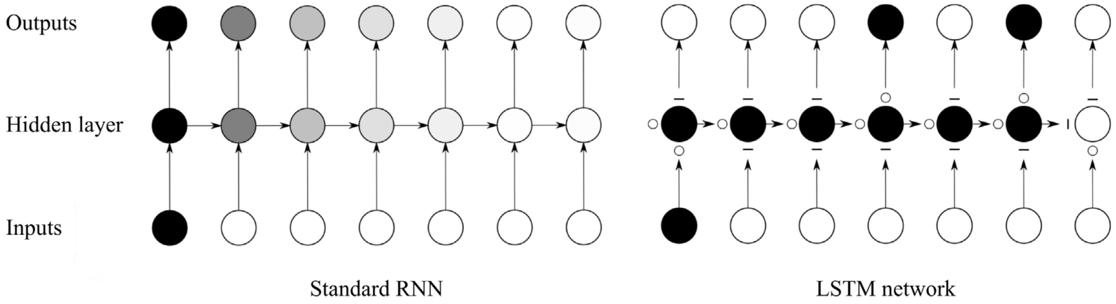

Fig. 3. Comparison of information decay in RNNs and LSTM networks.
Source: Adapted from Graves (2012)

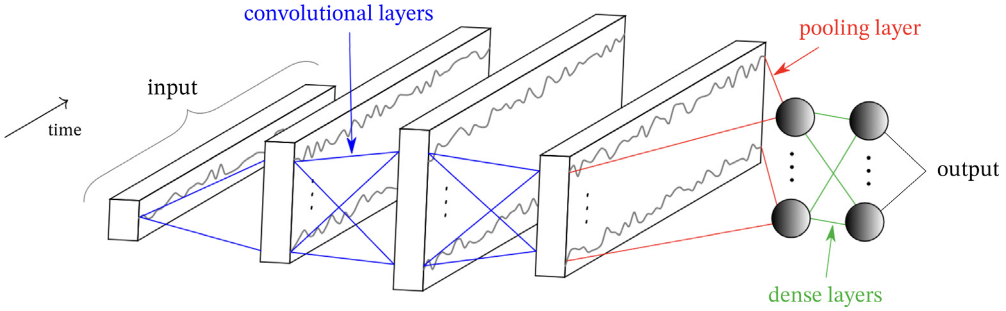

Fig. 4. CNN time series model architecture.
Source: Adapted from Ismail Fawaz et al. (2019)

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 (see Fig. 5). 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 above, are adept at capturing sequential patterns and ensuring that 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 alternate dynamically between the original input and the transformed input. This adaptability helps in selectively emphasising certain features, proving useful for variable selection and facilitating the extraction of feature importances.

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. 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 forecasting 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 demonstrated consistently superior performance over other RNNs on a variety of different datasets (Greff et al., 2017). A detailed overview of all employed ML models is found in Section 4.2.

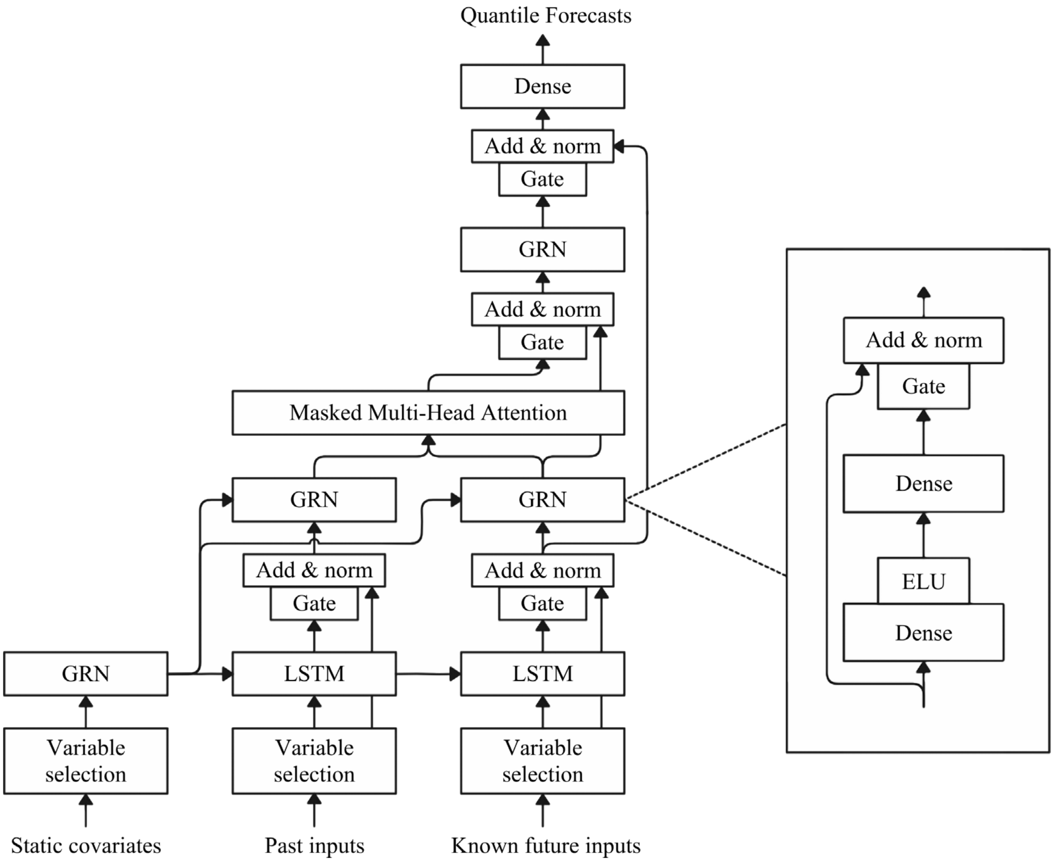

Fig. 5. TFT model architecture.

3.2. Deep learning approaches for NLP

Deep learning methods have revolutionised the field of NLP, offering context understanding, handling of linguistic ambiguities, reduced need 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 et al., 2019; Young et al., 2017).

In deep learning, words are represented as high-dimensional vectors called embeddings, which are capable of capturing semantic and syntactic similarities (Mikolov et al., 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 like the transformer (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 & 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. 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 et al., 2014; LeCun et al., 2015).

Furthermore, deep learning models have demonstrated significant efficacy in transfer learning applications within the realm of NLP. Like BERT and GPT, which are pre-trained on vast corpora, Large Language Models (LLMs) 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 unlabelled 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 & Ruder, 2018).

In this work, we apply three different deep-learning-based LLMs. The first is the fine-tuned 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 pre-trained 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 et al., 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 fine-tuned 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 433,000 sentence pairs annotated with textual entailment information. Each data point consists of (i) a premise, i.e. 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. As 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. This application of a zero-shot classification language model goes 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 fine-tune a pre-trained RoBERTa-Base model (Liu et al., 2019) directly on the cryptocurrency price. This model, in its raw form, is not yet trained to perform any specific task, and needs to be fine-tuned. As the target for the training, we opt for daily price movements represented as a binary variable.

All three models handle emoticons (e.g. ':)') and unicode (e.g. emoji) appropriately and no additional vocabulary need to be added given that they already contain all relevant cryptocurrency-related vernacular in their pre-trained vocabulary. Cleaning the textual data therefore only entails removing HTML elements and hyperlinks.

In Table 2, we present the top-scoring r/Bitcoin posts for the sentiment dictionary VADER and each of the three deep learning NLP models. The symbol '|' is used to separate the title of the Reddit post from the main body.

A post consisting merely of the word 'Good' achieved the highest rating according to the VADER sentiment dictionary. This is unsurprising, as 'Good' is one of the highest-ranking words, and any additional words would only lower the average score, thereby highlighting the inherent limitations of dictionary-based approaches in sentiment analysis.

The Twitter-RoBERTa sentiment model effectively selected a notably positive post, while the BART MNLI bullishness classifier chose a post that conveys optimism with regard to the future price of Bitcoin. It is noteworthy that the latter does not explicitly include terms like 'bullish' or 'bull', demonstrating the model's ability to infer higher-level semantics from the presented hypothesis.

When evaluating the third deep learning 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.

3.3. 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. 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 it.

For the local extrema analysis, we delve into an approach centred on local extrema, spanning observational intervals of +/− seven days, +/− 14 days, or +/− 21 days. We construct two binary variables that indicate whether a given timepoint is a local minimum or maximum within the set time interval (see Fig. 6). 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. In all three cases, our trading simulation starts by purchasing the asset at the first timestep and ends with the liquidation of all held assets at the last timestep.

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 than daily price changes, an argument can be made that they present a more stable and distinct pattern for machine learning approaches to model. Thus, even though the 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

Table 2
Highest-scoring r/Bitcoin subreddit posts by NLP model.

Model	Highest-scoring r/Bitcoin post
VADER (sentiment dictionary)	Good | Good
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.
Fine-tuned RoBERTa (trained directly on the cryptocurrency price movements)	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?

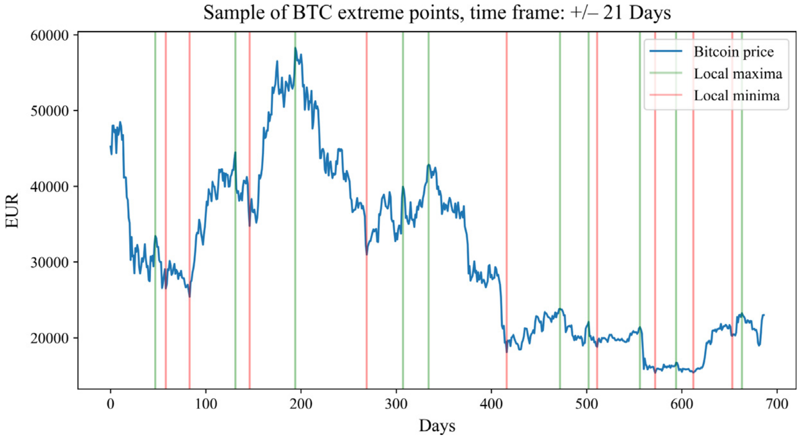

Fig. 6. Sample of local extreme points of the BTC price.

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.

For the trading simulation, we buy if the predicted probability of a local minimum occurring the next day exceeds a set threshold, provided the predicted 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 classification forecasts, we determine classification thresholds through a comprehensive grid search with the accuracy metric as the objective. This metric proves to correlate well with maximum profit, while offering a significant computational advantage over the extensive resources required for running trading simulations at each threshold.

4. Experimental design

4.1. Data collection and preprocessing

We utilise a diverse range of data sources, with the time frame of our dataset ranging from August 2011 for BTC and from August 2015 for ETH until March 2023. Fig. 7 outlines our data sources and experimental design.

We collect text data from social media platforms and news outlets, focusing on English-language content. From Google News, we extract approximately 55,000 news headlines, encompassing all articles from CoinDesk, Cointelegraph, and Decrypt that mention the keywords 'Bitcoin' or 'BTC' (and 'Ethereum' or 'ETH'). On Reddit, we gather all posts from the r/Bitcoin and r/ethereum subreddits, totalling around 338,000 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 hashtag #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

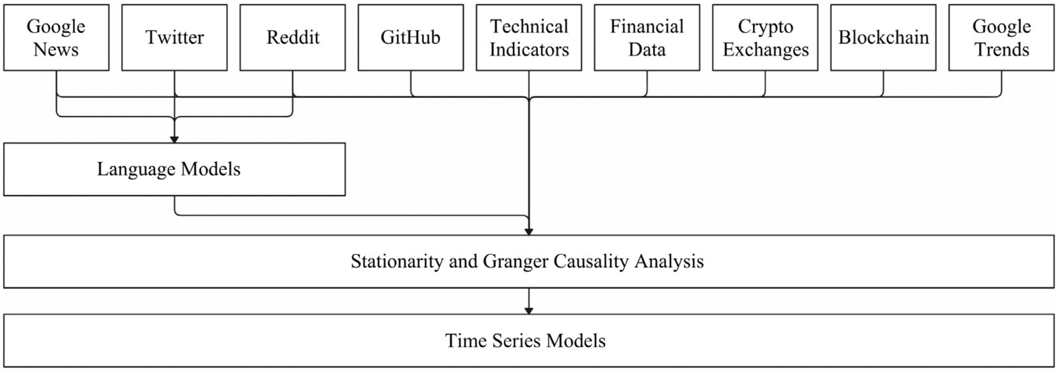

Fig. 7. Overview of the experimental design.

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 are processed using the sentiment dictionary VADER and the three LLMs detailed in Section 3.2. In addition to the textual information, we consider 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 the code hosting platform GitHub, specifically the repositories of the two cryptocurrencies we analyse (‘bitcoin/bitcoin’ and ‘ethereum/go-ethereum’), offering a perspective on development activity. We consider the commit count, number of additions/deletions, forks, stars, and subscribers.

Considering financial data, we incorporate 48 different technical indicators based on past price and volume, including trend, momentum, volatility, and volume indicators. Additionally, 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 the cryptocurrencies of interest, are also factored into the 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 leverage Google Trends to gauge public interest. We include 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 of Appendix B.

Values identified as erroneous due to technical anomalies, such as periods of social media platform downtime, are manually excluded. Apart from these specific instances, outliers are not removed from the dataset. The target variables and blockchain data are complete, containing no missing values. However, missing values are present in the financial data, notably prices on weekends, and are sporadically found in the social media and cryptocurrency exchange data due to server errors. These gaps are addressed by imputing the value from the previous day.

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.

We consider lagged values of up to 14 days of all features, including financial indicators, and apply Granger causality for feature selection. Our decision to use 14 lags is underpinned by the fact that the presence of causal lags diminished considerably beyond the 14th day. To select features for our non-sequential models, we test for a Granger causal relationship between each lag of each feature with the respective target variable. The sequential models require all lags to be passed. Thus, in this case, we test for Granger causality between all lags of each feature and the respective target.

Our Granger causality analysis reveals that the prices of both BTC and ETH are significantly influenced by their respective trading volumes, technical indicators, public sentiment, and broader economic trends. While both share common predictors, the BTC price shows a more pronounced response to its network metrics and large transactions, whereas ETH is affected more by developer activity. 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 have 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 et al., 2021) and Facebook's Prophet model (Taylor & Letham, 2017). Both approaches reveal no significant daily, weekly, or monthly seasonality. Hence, we refrain from including day-of-the-week or the month-of-the-year dummies 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. Model development and optimisation

As the target for the fine-tuning of RoBERTa-Base, we opt for daily price movements represented as a binary variable. We strategically utilise the available textual data: half of the data from each day are allocated to the training process, while the remaining half are 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 et al., 2019). The hyperparameter search entails 240 iterations with the Area Under the Receiver Operating Characteristic Curve (AUC ROC) as the objective function. The search ranges of the hyperparameter tuning are outlined in Table B.1 of Appendix B.

For the time series analysis, we employ a range of sequential and non-sequential 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'), as well as subsampling.

We construct the MLP with a maximum of four layers and apply a parameter penalty using the L2 norm to mitigate overfitting. Unlike most approaches in the existing body of literature, 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 additional flexibility, optimising its adaptability to different data patterns. Aside from the neuron count, we tune as hyperparameters the activation function, the batch size, the learning rate, the optimiser, and the type of scaling.

Next, we construct an LSTM architecture consisting 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 are the LSTM layer sizes 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, a model which is particularly challenging, due to its extensive training time. This is a result of it being fed all variables and conducting variable selection using GRNs, a notably less efficient method 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. Aside from that, 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 when training 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. 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 with Optuna, we use the trading profit as the objective function. The search ranges for this tuning are detailed in in Appendix B. For each target and model, the optimisation process is terminated either 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. Additionally, we measure the accuracy of the model by quantifying 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 and Sharpe ratio 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. To calculate the Sharpe ratio we imply a risk-free rate of 0% and apply annualisation as described by Sharpe (1994) for the sake

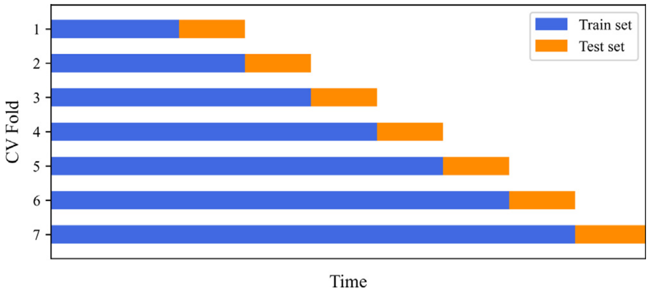

Fig. 8. The applied time series cross-validation approach.

of interpretability. We use 365 days for the annualisation, since cryptocurrencies are also traded on weekends.

$$\textit{Annualised Sharpe ratio} = \frac{365\,\bar{r}}{\sqrt{365}\,\sigma} \quad \text{where} \quad \bar{r} = \frac{1}{n}\sum_{t=1}^{n} r_t \,,$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}(r_t - \bar{r})^2}$$

$r_t :=$ asset return on day t

$n :=$ total number of days in the given time window

For the profit calculation, 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 ensure the 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 et al., 2020).

All the metrics are computed as averages of a seven-fold rolling-window cross-validation with incrementally increasing training window sizes (see Fig. 8). 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, to five different target variables, as explained in Section 3.3. 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 start by considering a few reference points. Table 3 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, with each fold spanning a time frame of approximately 1.5 years, and aggregated across both cryptocurrencies.

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 distil 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.

By integrating the NLP 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.

Fig. 9 offers a comparative analysis of the profit, Sharpe ratio, and 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 integration of the scores from both pre-trained LLMs: Twitter-RoBERTa and BART MNLI. The profit reported is

Table 3
Reference points for the analysis of trading profit.

Trading strategy	Buy-and-hold		Perfect knowledge of target	
	Profit	# of trades	Profit	# of trades
Price movement (binary)	182.78%	2.0	99,092.42%	205.9
Extrema +/− 7 days	"	"	759.40%	28.9
Extrema +/− 14 days	"	"	761.64%	14.2
Extrema +/− 21 days	"	"	702.27%	9.3

All metrics are averages of seven-fold cross-validation and aggregated across both coins.

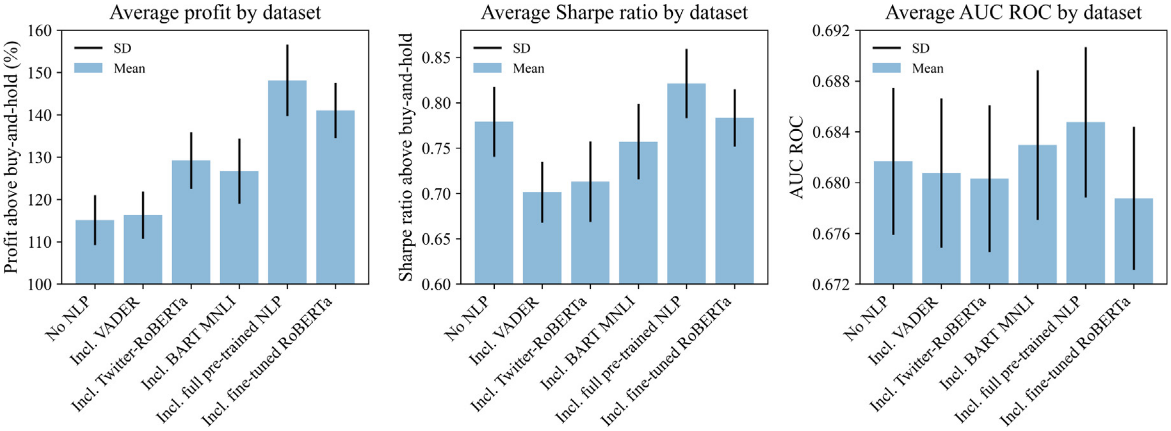

Fig. 9. Comparison of MLP profit, Sharpe ratio, and AUC ROC by dataset.

the amount of percentage points above the profit resulting from a buy-and-hold strategy. For a more detailed examination of the significance of NLP data across the various cross-validation splits, see Section 5.3.

The comparison vividly illustrates that deep learning NLP models surpass the sentiment dictionary VADER, highlighting the advanced capabilities of these models. Despite its simplicity, VADER contributes positively to forecasting profit. The Sharpe ratio, however, significantly diminishes when compared to using the no-NLP models, indicating a high level of noise in the VADER scores.

An intriguing observation is the interplay between the NLP models used. While the Twitter-RoBERTa sentiment model and the BART MNLI bullishness classifier perform on par individually, integrating both models yields the highest performance in terms of all metrics. This confirms that our NLP models indeed extract different signals and indicates that previous studies have not fully exploited the informative value of text data for forecasting.

Another observation that deserves 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. In particular, the comparatively low Sharpe ratio and AUC ROC indicate that the fine-tuned RoBERTa model introduced as much noise as it introduced information. This shows how complex the relationship between social media data and price fluctuations is and underscores the potential of transfer learning in the domain of financial forecasting.

Next, we examine the comparative performance of forecasting different targets. Fig. 10 provides a visual representation, illustrating the average performance of an MLP model broken down by target variables. A fundamental observation from our analysis is that all the chosen target types consistently outperform the buy-and-hold profit and Sharpe ratio accompanied by a decent AUC ROC, thus underscoring their viability for financial forecasting.

However, we discover certain differences. The binary representation of the daily price change, which simplifies the price movements into two categories of increase or decrease, consistently surpasses its continuous counterpart. This result might be a consequence of the binary representation being less affected by market noise. We also observe that, under our assumption of zero frictions, the daily models demonstrate superior profitability compared to the extrema models, despite exhibiting higher levels of noise. This can be attributed to the large potential profit associated with them, as showcased in Table 3. Therefore, even though we are able to harness only a small fraction of the daily fluctuations, this fraction still contains more information than what we are able to extract from the local extreme points. It is important to note, however, that the inclusion of transaction costs of 0.5% per transaction positions the extrema models as equally competitive in terms of profitability, due to their significantly lower trading frequency. In circumstances of low trading volume and high transaction costs, the extrema approach might therefore be not only competitive but even superior to forecasting daily price movements.

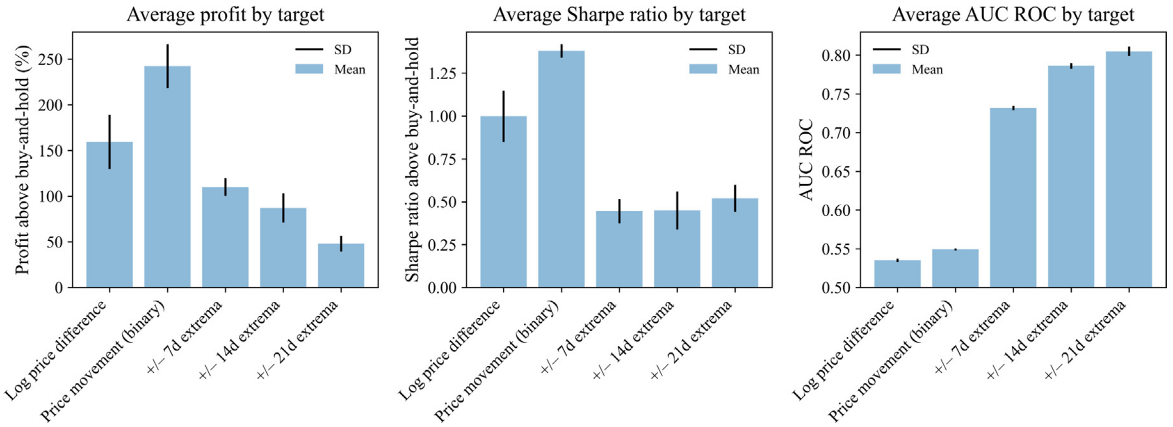

Fig. 10. Comparison of MLP profit, Sharpe ratio, and AUC ROC by target variable.

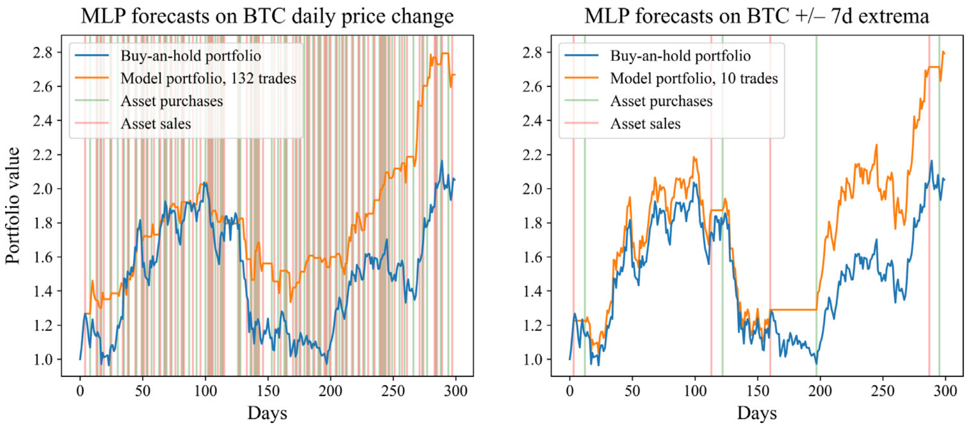

Fig. 11. Comparison of MLPs trained on daily price change and +/− seven-day extreme points.

Shifting our focus to the AUC ROC, the extreme point models show greater values than the daily price change models indicating 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 3, it becomes apparent that they capture a greater proportion of the information that the targets contain.

It is also worth highlighting that the Sharpe ratio looks more favourable for the extrema models than the profit, particularly for the 21-day window. The fact that the Sharpe ratio increases with the increasing observation time frames, while the profit decreases, proves that their more stable signal and the lower trading frequency do reduce the portfolio variance. This positions extreme points an efficient alternative to targets with higher granularity—not only in the face of high transaction costs but also for volatile assets or turbulent market phases.

In Fig. 11, 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 +/− seven-day extreme points. The blue line is the value of the asset, and the orange line is the value of the trading portfolio, while the vertical lines indicate points of entry and exit. Evidently, the extreme point models generate commendable results with significantly fewer trades than the model based on daily signals.

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 those with fewer positive predictions (indicating extremes) and subsequently fewer trades are the most profitable. For a detailed look at our findings, a thorough breakdown of the results can be found in Tables A.1 to A.10 of Appendix A.

Navigating through the spectrum of models (see Table 4), we find that the OLS-based approaches already generate substantial and consistent profits, as well as a decent AUC ROC, possibly due to the complex non-linearities already captured by the financial indicators. Nevertheless, all models except for the TFT consistently surpass the OLS benchmark, with the MLP taking the lead.

Table 4
Average performance by time series model.

Model	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
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

[a] Profit exceeding buy-and-hold strategy.

[b] All metrics are averages of seven-fold cross-validation and were aggregated across all target variables and coins.

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 data points 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. An emerging stream of literature, such as (Chen et al., 2023), has begun to critically evaluate the adoption of highly complex transformer models for time series forecasting. This literature suggests that while transformer architectures are powerful, simpler models, with better efficiency and interpretability, can often achieve comparable or even superior performance in time series forecasting, challenging the need for overly complex architectures.

5.2. Analysis of feature importance

To understand the significance of individual variables, we employ an XGBoost model trained on the entire set of available features. As prediction target, we utilise daily price movements encoded as a binary variable. This approach is justified by the performance of this model configuration, which achieved the highest AUC ROC for both BTC and ETH, and was also ranked second and third in terms of profitability for each cryptocurrency, respectively.

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 Tables 5 and 6, the feature importances are aggregated for all lags of each feature. Moreover, given the many variables, we present the importance scores consolidated by feature category. Interested readers can find disaggregated importance scores for individual features in of the Online Resource.

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 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.

Examining the ETH feature set, technical indicators persist in their dominance. Moreover, the significance of

Table 5
Feature categories ranked by their importance for predicting BTC price movements.

Feature category[a]	Normalised total gain	Normalised average gain
Technical indicators	0.3722	0.4998
Transaction/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

[a] Categories are sorted by total gain.

Table 6
Feature categories ranked by their importance for predicting ETH price movements.

Feature category[a]	Normalised total gain	Normalised average gain
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/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

[a] Categories are sorted by total gain.

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 longer 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, manifest as some of the most impactful explanatory variables for both BTC and ETH. This reaffirms our conclusions above 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 (Fama, 1970). In a market environment that is efficient, particularly in the semi-strong or strong form,

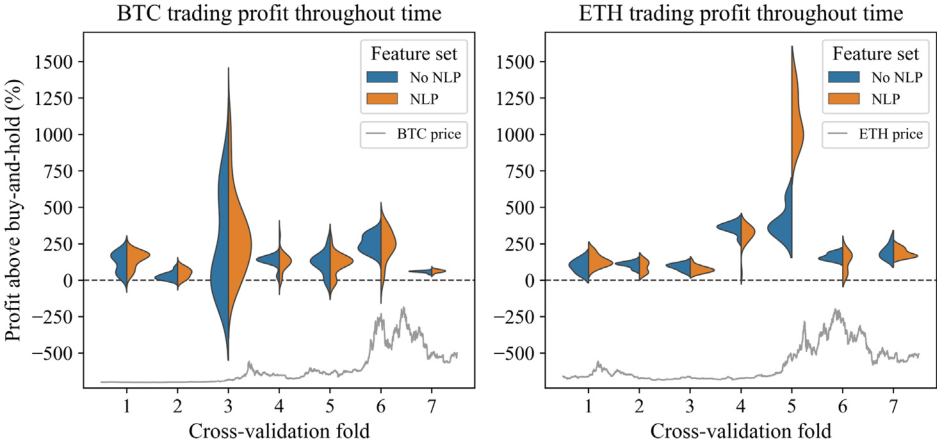

Fig. 12. Distribution of trading profits of the 10 most profitable MLP models throughout time.

consistently outperforming the market becomes challenging. 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 of whether the cryptocurrency market is efficient or not.

The time series models are evaluated by computing metrics on a seven-fold increasing window time-series cross-validation. By evaluating the model profits over the course of these seven cross-validation splits, we 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 with the underlying cryptocurrency. Additionally, by observing whether profits exhibit a trend over time, we can gauge whether the market is increasingly integrating NLP into their trading strategies, which might consequently shrink the potential future gains from text analysis.

Fig. 12 displays 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 largely 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.

We cannot observe a clear 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. These findings indicate that textual data, as analysed through our methods, may not have significantly impacted market efficiency during the study period.

However, 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 introduce 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 seem 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.

6. Conclusion

6.1. Summary of findings

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

Our research revealed that including NLP data improved the performance of our ML models with respect to all evaluated metrics. Furthermore, deep learning NLP models demonstrated superior performance compared with dictionary-based sentiment analysis. We found that pre-trained LLMs, namely Twitter-RoBERTa and BART MNLI, showed promising capabilities at capturing market sentiment, performing on par with language models that are fine-tuned directly on the target at hand.

Additionally, our results indicated that text features lagged by up to one week were Granger-causal and that incorporating NLP data in the time series models resulted in enhancements in forecasting 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 found that non-linear models outperformed those based on OLS, demonstrating the existence of relevant non-linear relationships in the time series. We further identified that using the daily price change as a binary target variable consistently resulted in superior profitability compared to other targets, at least under the assumption of no transaction costs. Nevertheless, our models more reliably predicted local extrema than daily price fluctuations, and the extreme point models yielded decent profits and Sharpe ratios with significantly fewer trades.

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

6.2. Implications

Incorporating NLP data into our models significantly improved price forecasting performance, demonstrating the value of considering such data in predictive efforts. Our study also underscored the efficacy of deep learning-based language models in this context. Particularly, when using several approaches in tandem, these models demonstrated significantly better performance than dictionary-based sentiment analysis.

Our results indicated that pre-trained models delivered comparable, if not superior, results to fine-tuned models, even when tackling abstract tasks in a specialised domain like finance. In particular, we found 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 and substantially enhances predictive accuracy. These findings suggest promising prospects for the use of transfer learning in NLP and not only highlight the versatility and robustness of pre-trained language models but also point towards a cost- and time-effective route for future endeavours in financial forecasting.

When turning our attention to the target variables, daily price movements encoded as a binary target consistently yielded the highest profit. However, our models were 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.

Data and code availability

The entire codebase and documentation of our experiments necessary to reproduce the results and analyses presented in our paper is available via an online repository: https://anonymous.4open.science/r/crypto-forecasting-public. Please note, that the repository only includes the codes to acquire the data, not the datasets themselves, as publishing them openly would infringe on the terms of service of some providers.

CRediT authorship contribution statement

Vincent Gurgul: Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation. **Stefan Lessmann:** Writing – review & editing, Validation, Supervision, Resources, Conceptualization. **Wolfgang Karl Härdle:** Writing – review & editing.

Declaration of competing interest

All authors certify that they have no affiliations with or involvement in any organisation or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Appendix A. Disaggregated results

See Tables A.1–A.11.

Appendix B. Documentation

See Tables B.1–B.3.

Table A.1
The 10 most profitable Bitcoin price movement regression models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
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	VADER	139.93%	8.6	0.5278	0.5357
LSTM	Twitter-RoBERTa	126.24%	61.1	0.5267	0.5375
LSTM	Twitter-RoBERTa + BART MNLI	107.76%	12.6	0.5213	0.5403

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.2
The 10 most profitable Ethereum price movement regression models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
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
MLP (FNN)	VADER	208.85%	80.9	0.5444	0.5481
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

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.3
The 10 most profitable Bitcoin price movement classification models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
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

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.4
The 10 most profitable Ethereum price movement classification models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
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

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.5
The 10 most profitable Bitcoin +/−seven-day extrema classification models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
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)	VADER	134.45%	2.6	0.7459	0.9577
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
LSTM	VADER	128.22%	2.3	0.6926	0.9568
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

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.6
The 10 most profitable Ethereum +/−seven-day extrema classification models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
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

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.7
The 10 most profitable Bitcoin +/−14-day extrema classification models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
MLP (FNN)	Fine-tuned RoBERTa	123.68%	2.9	0.7581	0.9795
MLP (FNN)	VADER	101.60%	3.1	0.7632	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
LSTM	VADER	21.00%	2.6	0.7159	0.9788

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.8
The 10 most profitable Ethereum +/−14-day extrema classification models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
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
LSTM	VADER	62.03%	2.0	0.7581	0.9812
XGBoost	Twitter-RoBERTa + BART MNLI	53.28%	2.6	0.8334	0.9821

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.9
The 10 most profitable Bitcoin +/−21-day extrema classification models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
LSTM	Twitter-RoBERTa + BART MNLI	81.72%	4.0	0.7565	0.9872
MLP (FNN)	VADER	68.95%	2.9	0.7939	0.9873
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
LSTM	VADER	26.53%	7.5	0.7572	0.9867
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

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.10
The 10 most profitable Ethereum +/−21-day extrema classification models.

Model	NLP features	Excess profit[a]	Trades[b]	AUC ROC[b]	Accuracy[b]
LSTM	Twitter-RoBERTa	121.54%	2.3	0.8260	0.9857
LSTM	VADER	84.99%	2.6	0.7730	0.9855
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

[a] Profit exceeding buy-and-hold strategy.
[b] All metrics are averages of seven-fold cross-validation.

Table A.11
The 50 most influential features for predicting BTC price movements using XGBoost.

Feature name[a]	Category	Normalised total gain	Normalised average gain
tweets_roberta_finetuned_score	NLP data	0.0754	0.0728
balance_distribution_from_0.01_addressesCount	Transaction/account balance data	0.0430	0.0285
reddit_count	Numerical social media data	0.0346	0.0223
tweets_twitter_roberta_pretrained_score	NLP data	0.0336	0.0343
indicator_UI	Technical indicators	0.0331	0.0278
average_transaction_value	Transaction/account balance data	0.0308	0.0156
indicator_AO	Technical indicators	0.0291	0.0434
indicator_Ichimoku_A	Technical indicators	0.0272	0.0294
balance_distribution_from_0.01_totalVolume	Transaction/account balance data	0.0260	0.0181
price_close	Past price data	0.0235	0.0187
current_supply	Technical blockchain metrics	0.0204	0.0126
indicator_NVI	Technical indicators	0.0200	0.0338
total_volume	Exchange volume data	0.0198	0.0167
indicator_Ichimoku_Conversion	Technical indicators	0.0197	0.0241
gtrends_blockchain_relative_change	Google Trends	0.0191	0.0134
indicator_EMA	Technical indicators	0.0189	0.0222
EUR_volumefrom	Exchange volume data	0.0181	0.0120
zero_balance_addresses_all_time	Transaction/account balance data	0.0174	0.0099
indicator_CR	Technical indicators	0.0172	0.0148
balance_distribution_from_100.0_addressesCount	Transaction/account balance data	0.0170	0.0101
indicator_MACD	Technical indicators	0.0165	0.0344
indicator_DCM	Technical indicators	0.0164	0.0217
indicator_PPO	Technical indicators	0.0160	0.0376
balance_distribution_from_10.0_addressesCount	Transaction/account balance data	0.0154	0.0095
reddit_bart_mnli_bullish_score	NLP data	0.0149	0.0093
indicator_Vortex_down	Technical indicators	0.0148	0.0096
indicator_KCW	Technical indicators	0.0145	0.0088
unique_addresses_all_time	Transaction/account balance data	0.0135	0.0093
large_transaction_count	Transaction/account balance data	0.0134	0.0092

(continued on next page)

Table A.11 (*continued*).

Feature name[a]	Category	Normalised total gain	Normalised average gain
indicator_Stoch_RSI	Technical indicators	0.0131	0.0096
block_time	Technical blockchain metrics	0.0128	0.0098
indicator_KCM	Technical indicators	0.0126	0.0186
tweet_count	Numerical social media data	0.0120	0.0097
balance_distribution_from_0.1_totalVolume	Transaction/account balance data	0.0119	0.0088
indicator_BBM	Technical indicators	0.0110	0.0167
hashrate	Technical blockchain metrics	0.0110	0.0097
indicator_KAMA	Technical indicators	0.0108	0.0104
gold_usd_price	Financial data	0.0093	0.0083
indicator_BBW	Technical indicators	0.0092	0.0049
indicator_Ichimoku_Base	Technical indicators	0.0090	0.0171
block_size	Technical blockchain metrics	0.0089	0.0047
balance_distribution_from_0.0_addressesCount	Transaction/account balance data	0.0087	0.0046
EUR_volumeto	Exchange volume data	0.0086	0.0091
indicator_TRIX	Technical indicators	0.0085	0.0178
balance_distribution_from_100.0_totalVolume	Transaction/account balance data	0.0076	0.0043
indicator_WMA	Technical indicators	0.0076	0.0090
balance_distribution_from_1000.0_totalVolume	Transaction/account balance data	0.0076	0.0051
indicator_DPO	Technical indicators	0.0075	0.0047
balance_distribution_from_10.0_totalVolume	Transaction/account balance data	0.0072	0.0044
balance_distribution_from_1000.0_addressesCount	Transaction/account balance data	0.0071	0.0048

[a] Features are sorted by total gain; see for variable definitions.

Table A.12
The 50 most influential features for predicting ETH price movements using XGBoost.

Feature name[a]	Category	Normalised total gain	Normalised average gain
tweets_twitter_roberta_pretrained_score	NLP data	0.0578	0.0440
tweets_roberta_finetuned_score	NLP data	0.0540	0.0372
news_roberta_finetuned_score	NLP data	0.0472	0.0476
indicator_CR	Technical indicators	0.0397	0.0220
reddit_roberta_finetuned_score	NLP data	0.0397	0.0247
indicator_EMA	Technical indicators	0.0359	0.0519
indicator_Ichimoku_A	Technical indicators	0.0302	0.0308
indicator_VWAP	Technical indicators	0.0301	0.0355
indicator_DCM	Technical indicators	0.0300	0.0345
indicator_BBM	Technical indicators	0.0290	0.0492
indicator_MI	Technical indicators	0.0278	0.0140
total_issues	GitHub metrics	0.0264	0.0240
indicator_WilliamsR	Technical indicators	0.0252	0.0145
tweets_bart_mnli_bullish_score	NLP data	0.0233	0.0122
indicator_KAMA	Technical indicators	0.0225	0.0279
indicator_Ichimoku_Conversion	Technical indicators	0.0223	0.0253
indicator_WMA	Technical indicators	0.0222	0.0285
indicator_CMF	Technical indicators	0.0211	0.0133
indicator_TRIX	Technical indicators	0.0209	0.0480
indicator_Stoch_RSI	Technical indicators	0.0193	0.0121
indicator_ROC	Technical indicators	0.0188	0.0199
indicator_KCM	Technical indicators	0.0179	0.0555
indicator_FI	Technical indicators	0.0166	0.0144
closed_issues	GitHub metrics	0.0135	0.0133
price_close	Past price data	0.0129	0.0097
average_transaction_value	Transaction/account balance data	0.0128	0.0076
reddit_accounts_active_48h	Numerical social media data	0.0123	0.0046
indicator_ultimate	Technical indicators	0.0107	0.0076
indicator_Ichimoku_Base	Technical indicators	0.0107	0.0165
exchange_Coinbase_volumefrom	Exchange volume data	0.0097	0.0057
twitter_followers	Numerical social media data	0.0097	0.0039
indicator_Ichimoku_B	Technical indicators	0.0092	0.0227
unique_addresses_all_time	Transaction/account balance data	0.0091	0.0060
indicator_DCW	Technical indicators	0.0090	0.0059
staking_rate	Technical blockchain metrics	0.0089	0.0099
indicator_KST	Technical indicators	0.0084	0.0204

(*continued on next page*)

Table A.12 (*continued*).

Feature name[a]	Category	Normalised total gain	Normalised average gain
exchange_Kraken_volumeto	Exchange volume data	0.0083	0.0042
zero_balance_addresses_all_time	Transaction/account balance data	0.0082	0.0070
indicator_AO	Technical indicators	0.0076	0.0092
indicator_EMV	Technical indicators	0.0075	0.0048
indicator_VPT	Technical indicators	0.0074	0.0066
indicator_DPO	Technical indicators	0.0071	0.0051
indicator_Aroon_down	Technical indicators	0.0069	0.0071
total_volume	Exchange volume data	0.0067	0.0038
indicator_CCI	Technical indicators	0.0067	0.0070
indicator_Vortex_down	Technical indicators	0.0065	0.0045
indicator_MACD	Technical indicators	0.0064	0.0128
indicator_Stoch	Technical indicators	0.0064	0.0039
USD_volumeto	Exchange volume data	0.0062	0.0029
exchange_Kraken_volumefrom	Exchange volume data	0.0062	0.0039

[a] Features are sorted by total gain; see for variable definitions.

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

Hyperparameter	Search range
learning rate	5×10^{-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	0.001 to 100
	solver	SVD, Cholesky, LSQR, Sparse CG, SAG, SAGA
Logistic Regression	L2 regularisation parameter	0.0005 to 1000
	solver	L-BFGS, Liblinear, 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[a]	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[a]	50 to 300
	number of dense layers	0 to 3
	size of each dense layer[a]	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

(*continued on next page*)

Table B.2 (*continued*).

Model	Hyperparameter	Search range
	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[b]	16, 32, 64, 128
	dropout	0.1 to 0.5
	learning rate	5×10^{-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

[a] The number of neurons was tuned individually for each layer, not set uniformly for all.

[b] 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[a]	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	24 h	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	24 h	0	Average BART MNLI bullish score of news
tweets_bart_mnli_bullish_score	Twitter / Own calculation	24 h	0	Average BART MNLI bullish score of Twitter posts
reddit_bart_mnli_bullish_score	Reddit / Own calculation	24 h	0	Average BART MNLI bullish score of Reddit posts
news_twitter_roberta_pretrained_score	Google News / Own calculation	24 h	0	Average Twitter-RoBERTa sentiment score of news
tweets_twitter_roberta_pretrained_score	Twitter / Own calculation	24 h	0	Average Twitter-RoBERTa sentiment score of Twitter posts
reddit_twitter_roberta_pretrained_score	Reddit / Own calculation	24 h	0	Average Twitter-RoBERTa sentiment score of Reddit posts
news_roberta_finetuned_score	Google News / Own calculation	24 h	0	Average Finetuned RoBERTa score of news
tweets_roberta_finetuned_score	Twitter / Own calculation	24 h	0	Average Finetuned RoBERTa score of Twitter posts
reddit_roberta_finetuned_score	Reddit / Own calculation	24 h	0	Average Finetuned RoBERTa score of Reddit posts
news_count	Google News	24 h	1	Number of news articles from CoinDesk, Cointelegraph or Decrypt for the keywords Bitcoin, BTC (Ethereum, ETH)
tweet_count	Twitter	24 h	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	24 h	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

(*continued on next page*)

Table B.3 (*continued*).

Variable name	Source	Interval	I[a]	Description
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	24 h	1	Number of additions on the GitHub repository
deletions	GitHub	24 h	1	Number of deletions on the GitHub repository
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	24 h	1	Volume of transactions from ETH to BTC (BTC to ETH) across 301 exchanges
ETH_volumeto (BTC_volumeto)	CryptoCompare	24 h	1	Volume of transactions from BTC to ETH (ETH to BTC) across 301 exchanges
USD_volumefrom	CryptoCompare	24 h	1	Volume of transactions from USD to BTC (ETH) across 301 exchanges
USD_volumeto	CryptoCompare	24 h	1	Volume of transactions from BTC (ETH) to USD across 301 exchanges
EUR_volumefrom	CryptoCompare	24 h	1	Volume of transactions from EUR to BTC (ETH) across 301 exchanges
EUR_volumeto	CryptoCompare	24 h	1	Volume of transactions from BTC (ETH) to EUR across 301 exchanges
exchange_Bitfinex_volumeto	CryptoCompare	24 h	1	Inflow of BTC (ETH) on Bitfinex exchange
exchange_Bitfinex_volumefrom	CryptoCompare	24 h	1	Outflow of BTC (ETH) on Bitfinex exchange
exchange_Bitfinex_volumetotal	CryptoCompare	24 h	1	Total BTC (ETH) cashflows on Bitfinex exchange
exchange_Kraken_volumeto	CryptoCompare	24 h	1	Inflow of BTC (ETH) on Kraken exchange
exchange_Kraken_volumefrom	CryptoCompare	24 h	1	Outflow of BTC (ETH) on Kraken exchange
exchange_Kraken_volumetotal	CryptoCompare	24 h	1	Total BTC (ETH) cashflows on Kraken exchange
exchange_Coinbase_volumeto	CryptoCompare	24 h	1	Inflow of BTC (ETH) on Coinbase exchange
exchange_Coinbase_volumefrom	CryptoCompare	24 h	1	Outflow of BTC (ETH) on Coinbase exchange
exchange_Coinbase_volumetotal	CryptoCompare	24 h	1	Total BTC (ETH) cashflows on Coinbase exchange
exchange_BTSE_volumeto	CryptoCompare	24 h	1	Inflow of BTC (ETH) on BTSE exchange
exchange_BTSE_volumefrom	CryptoCompare	24 h	1	Outflow of BTC (ETH) on BTSE exchange
exchange_BTSE_volumetotal	CryptoCompare	24 h	1	Total BTC (ETH) cashflows on BTSE exchange
exchange_Binance_volumeto	CryptoCompare	24 h	1	Inflow of BTC (ETH) on Binance exchange
exchange_Binance_volumefrom	CryptoCompare	24 h	1	Outflow of BTC (ETH) on Binance exchange
exchange_Binance_volumetotal	CryptoCompare	24 h	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	24 h	1	Amount of new BTC (ETH) addresses created
active_addresses	IntoTheBlock	24 h	1	Amount of BTC (ETH) addresses that executed at least one transaction
transaction_count	IntoTheBlock	24 h	1	Number of valid transactions on the BTC (ETH) blockchain
large_transaction_count	IntoTheBlock	24 h	1	Number of valid transactions greater than 100,000 USD on the BTC (ETH) blockchain
average_transaction_value	IntoTheBlock	24 h	1	Average transaction value on the BTC (ETH) blockchain in BTC (ETH)

(*continued on next page*)

Table B.3 (*continued*).

Variable name	Source	Interval	I[a]	Description
hashrate (only Bitcoin)	IntoTheBlock	24 h	1	Number of terahashes per second the BTC network is performing
difficulty (only Bitcoin)	IntoTheBlock	24 h	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 min)
block_time (only Bitcoin)	IntoTheBlock	24 h	1	Average time in seconds it takes miners to verify transactions within one block on the BTC network
block_size	IntoTheBlock	24 h	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	24 h	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	24 h	1	Percentage change in amount of Google searches of the keyword ‘bitcoin’

(*continued on next page*)

Table B.3 (*continued*).

Variable name	Source	Interval	I[a]	Description
gtrends_ethereum_relative_change	Google Trends	24 h	1	Percentage change in amount of Google searches of the keyword 'ethereum'
gtrends_cryptocurrency_relative_change	Google Trends	24 h	1	Percentage change in amount of Google searches of the keyword 'cryptocurrency'
gtrends_blockchain_relative_change	Google Trends	24 h	1	Percentage change in amount of Google searches of the keyword 'blockchain'
gtrends_investing_relative_change	Google Trends	24 h	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	24 h	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

(*continued on next page*)

Table B.3 (*continued*).

Variable name	Source	Interval	I[a]	Description
indicator_VPT	Own calculation	timepoint	1	Volume Price Trend (VPT) - Combines price and volume to show the direction of price trend
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, indicator_PSAR_up	Own calculation	timepoint	0	Parabolic stop and reverse - 'down' (providing exit points) and 'up' (providing entry points)

[a] Order of integration (number of times the time series had to be differenced to become stationary).

References

Abraham, J., Higdon, D., Nelson, J., Ibarra, J., & Nelson, J. (2018). Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, *1*, URL: https://scholar.smu.edu/cgi/viewcontent.cgi?article=1039&context=datasciencereview.

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. ArXiv:1907.10902 [cs, stat] URL: https://arxiv.org/abs/1907.10902.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. URL: https://arxiv.org/abs/1409.0473.

Bandara, K., Hyndman, R. J., & Bergmeir, C. (2021). MSTL: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns. ArXiv:2107.13462 [stat] URL: https://arxiv.org/abs/2107.13462.

Chen, C. Y.-H., Despres, R., Guo, L., & Renault, T. (2019). What makes cryptocurrencies special? Investor sentiment and return predictability during the bubble. *SSRN Electronic Journal*, http://dx.doi.org/10.2139/ssrn.3398423.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., & Pfister, T. (2023). TSMixer: An all-MLP architecture for time series forecasting. http://dx.doi.org/10.48550/arXiv.2303.06053, URL: https://arxiv.org/abs/2303.06053.

Colianni, S., Rosales, S., & Signorotti, M. (2015). Algorithmic trading of cryptocurrency based on Twitter sentiment analysis. URL: http://cs229.stanford.edu/proj2015/029_report.pdf.

Dwivedi, H. (2023). Cryptocurrency sentiment analysis using bidirectional transformation. (pp. 140–142). http://dx.doi.org/10.1109/ICSMDI57622.2023.00032, URL: https://ieeexplore.ieee.org/abstract/document/10127932.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*, 383–417. http://dx.doi.org/10.2307/2325486, URL: https://www.jstor.org/stable/2325486.

Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., & Li, L. (2022). Cryptocurrency trading: A comprehensive survey. *Financial Innovation*, *8*, 1–59. http://dx.doi.org/10.1186/s40854-021-00321-6, URL: https://jfin-swufe.springeropen.com/articles/10.1186/s40854-021-00321-6.

Gers, F., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*, 2451–2471. http://dx.doi.org/10.1162/089976600300015015.

Goldberg, Y. (2017). *Neural network methods in natural language processing*. Morgan & Claypool.

Goodfellow, I., Bengio, Y., & 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., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*, 2222–2232. http://dx.doi.org/10.1109/tnnls.2016.2582924.

Hafid, A., Hafid, A. S., & Samih, M. (2020). Scaling blockchains: A comprehensive survey. *IEEE Access*, *8*, 125244–125262. http://dx.doi.org/10.1109/access.2020.3007251.

Hao, V. M., Huy, N. H., Dao, B., Mai, T.-T., & Nguyen-An, K. (2019). Predicting cryptocurrency price movements based on social media. In *2019 international conference on advanced computing and applications*. http://dx.doi.org/10.1109/acomp.2019.00016.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1–42. http://dx.doi.org/10.1162/neco.1997.9.1.1, URL: https://www.bioinf.jku.at/publications/older/2604.pdf.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. URL: https://arxiv.org/abs/1801.06146.

Ider, D., & Lessmann, S. (2022). Cryptocurrency return prediction using investor sentiment extracted by BERT-based classifiers from news articles, Reddit posts and tweets. ArXiv:2204.05781 [cs, q-fin] URL: https://arxiv.org/abs/2204.05781.

Inamdar, A., Bhagtani, A., Bhatt, S., & Shetty, P. M. (2019). Predicting cryptocurrency value using sentiment analysis. (pp. 932–934). http://dx.doi.org/10.1109/ICCS45141.2019.9065838, URL: https://ieeexplore.ieee.org/abstract/document/9065838.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: A review. *Data Mining and Knowledge Discovery*, http://dx.doi.org/10.1007/s10618-019-00619-1.

Jain, A., Tripathi, S., Dwivedi, H. D., & Saxena, P. (2018). Forecasting price of cryptocurrencies using tweets sentiment analysis. In *2018 Eleventh International Conference on Contemporary Computing*. http://dx.doi.org/10.1109/ic3.2018.8530659.

Karalevicius, V., Degrande, N., & De Weerdt, J. (2018). Using sentiment analysis to predict interday Bitcoin price movements. *The Journal of Risk Finance*, *19*, 56–75. http://dx.doi.org/10.1108/jrf-06-2017-0092.

Kim, G., Kim, M., Kim, B., & Lim, H. (2023). CBITS: Crypto BERT incorporated trading system. *IEEE Access*, *11*, 6912–6921. http://dx.doi.org/10.1109/ACCESS.2023.3236032, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10014986.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. http://dx.doi.org/10.1038/nature14539, URL: https://www.nature.com/articles/nature14539.

Leung, M. T., Daouk, H., & Chen, A.-S. (2000). Forecasting stock indices: A comparison of classification and level estimation models. *International Journal of Forecasting*, *16*, 173–190. http://dx.doi.org/10.1016/s0169-2070(99)00048-5.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ArXiv:1910.13461 [cs, stat] URL: https://arxiv.org/abs/1910.13461.

Li, T. R., Chamrajnagar, A. S., Fong, X. R., Rizik, N. R., & Fu, F. (2019). Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model. *Frontiers in Physics*, *7*, http://dx.doi.org/10.3389/fphy.2019.00098.

Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2020). Temporal fusion transformers for interpretable multi-horizon time series forecasting. ArXiv:1912.09363 [cs, stat] URL: https://arxiv.org/abs/1912.09363.

Liu, B. (2012). *Sentiment analysis and opinion mining*. Morgan & Claypool.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. URL: https://arxiv.org/abs/1907.11692.

Loureiro, D., Barbieri, F., Neves, L., Espinosa, L., & Camacho-Collados, J. (2022). TimeLMs: Diachronic language models from Twitter. URL: https://arxiv.org/pdf/2202.03829.pdf.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. URL: https://arxiv.org/pdf/1310.4546.pdf.

Mittal, A., Dhiman, V., Singh, A., & Prakash, C. (2019). Short-term Bitcoin price fluctuation prediction using social media and web search data. http://dx.doi.org/10.1109/ic3.2019.8844899,

Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*, http://dx.doi.org/10.1007/s00521-020-05129-6.

Murray, K., Rossi, A., Carraro, D., & Visentin, A. (2023). On forecasting cryptocurrency prices: A comparison of machine learning, deep learning, and ensembles. *Forecasting*, *5*, 196–209. http://dx.doi.org/10.3390/forecast5010010.

Ortu, M., Uras, N., Conversano, C., Bartolucci, S., & Destefanis, G. (2022). On technical trading and social media indicators for cryptocurrency price classification through deep learning. *Expert Systems with Applications*, *198*, Article 116804. http://dx.doi.org/10.1016/j.eswa.2022.116804.

Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, *93*, Article 106384. http://dx.doi.org/10.1016/j.asoc.2020.106384.

Pant, D. R., Neupane, P., Poudel, A., Pokhrel, A. K., & Lama, B. K. (2018). Recurrent neural network based Bitcoin price prediction by Twitter sentiment analysis. http://dx.doi.org/10.1109/cccs.2018.8586824,

Parekh, R., Patel, N. P., Thakkar, N., Gupta, R., Tanwar, S., Sharma, G., DAVIDSON, I. E., & Sharma, R. (2022). DL-GuesS: Deep learning and sentiment analysis-based cryptocurrency price prediction. *IEEE Access*, 1. http://dx.doi.org/10.1109/access.2022.3163305.

Raju, S. M., & Tarif, A. M. (2020). Real-time prediction of BITCOIN price using machine learning techniques and public sentiment analysis. ArXiv:2006.14473 URL: https://arxiv.org/abs/2006.14473.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. URL: https://arxiv.org/pdf/1908.10084.pdf.

Ruder, S., Vulić, I., & Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, *65*, 569–631. http://dx.doi.org/10.1613/jair.1.11640, URL: https://arxiv.org/pdf/1706.04902.pdf.

Sharpe, W. F. (1994). The Sharpe ratio. URL: https://web.stanford.edu/~wfsharpe/art/sr/sr.htm.

Shleifer, A. (2009). *Inefficient markets: An introduction to behavioral finance*. Oxford Univ. Press.

Subramanian, H., Angle, P., Rouxelin, F., & Zhang, Z. (2024). A decision support system using signals from social media and news to predict cryptocurrency prices. *Decision Support Systems*, *178*, Article 114129. http://dx.doi.org/10.1016/j.dss.2023.114129, URL: https://www.sciencedirect.com/science/article/abs/pii/S016792362300204X.

Taylor, S. J., & Letham, B. (2017). Forecasting at scale. *The American Statistician*, *72*, 37–45. http://dx.doi.org/10.1080/00031305.2017.1380080, URL: http://lethalletham.com/ForecastingAtScale.pdf.

Valencia, F., Gómez-Espinosa, A., & Valdés-Aguirre, B. (2019). Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. *Entropy*, *21*, 589. http://dx.doi.org/10.3390/e21060589.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. URL: https://arxiv.org/abs/1706.03762.

Widianto, M. H., & Cornelius, Y. (2023). 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, URL: https://www.ijisae.org/index.php/IJISAE/article/view/2539/1122.

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

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2017). Recent trends in deep learning based natural language processing. URL: https://arxiv.org/abs/1708.02709.

Zhang, J., Cai, K., & Wen, J. (2024). A survey of deep learning applications in cryptocurrency. *IScience*, *27*, Article 108509. http://dx.doi.org/10.1016/j.isci.2023.108509.

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