Kaunas University of Technology
Faculty of Mathematics and Natural Sciences

Cryptocurrency Steep Price Movement Prediction Using Reddit Messages and Technical Indicators

Master's Final Degree Project

Dominykas Švedas
Project Author

Assoc. Prof. Dr. Kristina Šutienė
Supervisor

Assoc. Prof. Dr. Aušrinė Lakštutienė
Supervisor

Kaunas, 2023

Kaunas University of Technology
Faculty of Mathematics and Natural Sciences

Cryptocurrency Steep Price Movement Prediction Using Reddit Messages and Technical Indicators

Master's Final Degree Project
Business Big Data Analytics (6213AX001)

Dominykas Švedas
Project Author

Assoc. Prof. Dr. Kristina Šutienė
Supervisor

Assoc. Prof. Dr. Aušrinė Lakštutienė
Supervisor

Assoc. Prof. Dr. Mindaugas Kavaliauskas
Reviewer

Dr. Arvydas Jadevičius
Reviewer

Kaunas, 2023

Kaunas University of Technology
Faculty of Mathematics and Natural Sciences
Dominykas Švedas

Cryptocurrency Steep Price Movement Prediction Using Reddit Messages and Technical Indicators

Declaration of Academic Integrity

I confirm the following:

1. I have prepared the final degree project independently and honestly without any violations of the copyrights or other rights of others, following the provisions of the Law on Copyrights and Related Rights of the Republic of Lithuania, the Regulations on the Management and Transfer of Intellectual Property of Kaunas University of Technology (hereinafter – University) and the ethical requirements stipulated by the Code of Academic Ethics of the University;
2. All the data and research results provided in the final degree project are correct and obtained legally; none of the parts of this project are plagiarised from any printed or electronic sources; all the quotations and references provided in the text of the final degree project are indicated in the list of references;
3. I have not paid anyone any monetary funds for the final degree project or the parts thereof unless required by the law;
4. I understand that in the case of any discovery of the fact of dishonesty or violation of any rights of others, the academic penalties will be imposed on me under the procedure applied at the University; I will be expelled from the University and my final degree project can be submitted to the Office of the Ombudsperson for Academic Ethics and Procedures in the examination of a possible violation of academic ethics.

Dominykas Švedas
Confirmed electronically

Dominykas Švedas. Cryptocurrency steep price movement prediction using Reddit messages and technical indicators. Master's Final Degree Project/ supervisors Assoc. Prof. Dr. Kristina Šutienėand Assoc. Prof. Dr. Aušrinė Lakštutienė; Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Study field and area (study field group): Applied Mathematics (Mathematical Sciences).

Keywords: Crypto, Cryptocurrency, Price movement prediction, LSTM, Transformers, GPT-3.5.

Kaunas, 2023. 68.

Summary

This thesis explores the prediction of steep price movements in cryptocurrencies by employing Reddit messages and technical indicators. 'Steep' is defined as daily price changes that exceed 10 percent in either direction. The analysis involves a wide variety of cryptocurrencies traded on Binance, one of the leading cryptocurrency exchanges, covering data from 2020 and 2021.

The study explores the impact of sentiments extracted from Reddit messages on predicting these significant price movements. Sentiment analysis was conducted using advanced Natural Language Processing models, including Bidirectional Encoder Representation from Transformers (BERT), XL-Net, BigBird-RoBERTa, and OpenAI's Generative Pertained Transformer 3.5 (GPT-3.5).

The effectiveness of Reddit-derived sentiments in predicting price movements were conducted using traditional machine learning algorithms, namely Random Forest and Gradient Boosting. This was contrasted with a recurrent neural network model, the Bidirectional Long Short-Term Memory (Bi-LSTM). Results revealed that the Bi-LSTM model outperformed traditional machine learning models. Furthermore, the study demonstrated that sentiments extracted from Reddit messages significantly improved the performance of price movement prediction models, thereby highlighting the potential of integrating social media sentiment analysis into cryptocurrency price forecasting.

However, the implementation of a simple trading strategy using these predictions did not yield profitable results when backtested.

Dominykas Švedas. Staigių kriptovaliutų kainų pokyčių prognozavimas naudojant „Redit“ žinutes ir techninius rodiklius. Magistro studijų baigiamasis projektas/ vadovės doc. dr. Kristina Šutienėir doc. dr. Aušrinė Lakštutienė; ; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Taikomoji matematika (Matematikos mokslai).

Reikšminiai žodžiai: Kriptovaliutos, kainos pokyčio prognozė.

Kaunas, 2023, 68 p.

Santrauka

Darbe nagrinėjami staigūs kriptovaliutų kainos pokyčiai, įtraukiant „Redit“ žinutes ir techninės analizės kintamuosius. „Staigus“ pokytis apibrėžtas kaip dienos kainos pasikeitimas, viršijantis 10 procentų į kurią nors pusę. Analizėje įtraukta daugybė skirtingų kriptovaliutų, prekiaujamų „Binance“, viename pirmaujančių kriptovaliutų keityklų, periodu nuo 2020 iki 2021.

Tyrime nagrinėjama nuotaikų (sentimentų), gautų išanalizavus „Redit“ žinutes, įtaka bandant atspėti kainos pokyčio kryptį. Nuotaikų analizė atlikta naudojant pažangius natūralios kalbos apdorojimo modelius, tokius kaip „Bidirectional Encoder Representations from Transformers (BERT)‘, „XLNet“, „BigBird-RoBERTa“ ir „OpenAI’s Generative Pertained Transformer 3.5 (GPT-3.5)“.

„Reddit“ gautų nuotaikų veiksmingumas spėjant kainų pokyčių kryptį buvo atliktas naudojant tradicinius mašininio mokymosi algoritmus - atsitiktinį mišką ir gradiento didinimą. Rezultatai buvo palyginti su pasikartojančio neuroninio tinklo modeliu, dvikrypte ilgalaike trumpalaike atmintimi („Bi-LSTM”). Pastarasis pranoko tradicinius mašininio mokymosi modelius. Be to, tyrimas atskleidė, kad nuotaikų, gautų iš „Redit“ pranešimų, įtraukimas į modelį pagerino modelio rezultatus, taip nustatydamas socialinių tinklų nuotaikų svarbą kritovaliutų kainos prognozėje.

Kita vertus, pritaikius paprastą prekybos strategiją naudojant praeities duomenis, paaiškėjo, kad pasirinktas metodas neteikia pelningų rezultatų investuojant.

Contents

List of Figures **9**

List of Tables **10**

1 Literature Review **13**
1.1 Introduction . . . 13
1.2 Short history of cryptocurrency . . . 13
1.3 Technical indicators in predicting price movement . . . 14
1.4 Machine Learning and Statistical Methods to predict price . . . 16
1.5 Sentiment analysis methods . . . 19
1.6 Sentiment Analysis and its Role in Financial Markets . . . 21
1.7 Social Media Sentiment and Financial Markets . . . 22
1.7.1 Early Studies on the Role of Internet Discussions and Stock Market. . . . 22
1.7.2 Twitter Sentiment and Stock Market . . . 22
1.7.3 Reddit Sentiment and Financial Market . . . 22
1.8 Other related work . . . 23
1.9 Summary . . . 25

2 Methodology **26**
2.1 Data collection and short overview . . . 26
2.1.1 Cryptocurrency price data . . . 26
2.1.2 Reddit data . . . 26
2.2 Data labeling . . . 28
2.2.1 Cryptocurrency price change labeling . . . 28
2.2.2 Reddit posts preprocessing and labeling . . . 30
2.3 Test data holdout . . . 30
2.4 Classification of textual data using BERT, BigBird-RoBERTa, and XLNet . . . 30
2.5 Classification of textual data using GPT- 3.5 . . . 31
2.6 Feature engineering . . . 32
2.6.1 Computation of technical indicators . . . 32
2.6.2 Features from Reddit sentiment analysis . . . 32
2.7 Data manipulation before price prediction . . . 32
2.7.1 Data split for model training and best model selection . . . 32
2.7.2 Reddit messages cleaning and a subset of data selection . . . 33
2.8 Hyperparameters search for Random Forest and Gradient Boost . . . 33
2.9 Features combination . . . 36
2.10 Bidirectional Long Short-Term Memory (BI-LSTM) . . . 36
2.11 Data Scaling for BI-LSTM . . . 37
2.12 Testing and training strategy on the held-out data . . . 38

3 Research results **40**
3.1 Reddit sentiment analysis . . . 40
3.1.1 Hyperparameter search and method selection . . . 40

3.1.2 Fine tuning BigBird-RoBERTa . . . 40
3.1.3 Overview of sentiment predictions and the labeling problem . . . 40
3.2 Sentiment classification using OpenAI's Generative Pre-Trained Transformer-3.5 (GPT-3.5) . . . 41
3.3 Price movement classification models . . . 43
3.3.1 Model features selection and Hyperparameters search . . . 43
3.4 Bidirectional Long Short-Term Memory (Bi-LSTM) . . . 44
3.4.1 Results of the hyperparameter search . . . 45
3.4.2 Performance of Bi-LSTM and comparison with other methods on test data . . . 45
3.5 Bi-LSTM performance on holdout data . . . 46
3.6 Trading strategy and backtesting trading results . . . 46

Conclutions **52**

References **53**

Appendices **60**

A Data overview and labels distribution **61**
A.1 Data Overview . . . 61
A.2 Labels distribution among cryptocurrencies with the different threshold for the price change . . . 62

B Technical indicators list **67**

List of Figures

Fig. 1.1. Major Cryptoassets By Percentage of Total Market Capitalization (Bitcoin Dominance Chart), May 5, 2013 - Apr 15, 2023. CoinMarketCap [1] 14
Fig. 1.2. Various models for long-term stock price prediction of 5767 European companies performance measured in median AUC, Ballings et al. [2] 18
Fig. 1.3. Results from Chen [3, p. 14] paper on used classification methods accuracy and MCC, where *n* denotes a percentile of extreme news used from classification . . . 21
Fig. 1.4. Mishev et al. [4] chart of sentiment analysis models F1-score for financial phrase dataset 21

Fig. 2.1. Price of most trade volume cryptocurrencies comparison in USD and μBTC 27
Fig. 2.2. Number of Reddit posts in subreddits over time 28
Fig. 2.3. Sample of cryptocurrencies label amount with 10% threshold for the price change in the upcoming 7 days 29
Fig. 2.4. Count of cryptocurrency positive and negative price change labels, calculated as shown in the equation 2.1, over time 29
Fig. 2.5. Split of the dataset for the training and validation, testing for the best model selection, and the final testing in comparison to the Bitcoin price in USD 33
Fig. 2.6. Cryptocurrencies excluded from the model training due to abnormally high volume of Reddit messages in a short period 34
Fig. 2.7. Reddit posts count over time after excluding abnormal cryptocurrencies, which is shown in 2.5 35
Fig. 2.8. Illustration of training and prediction on the holdout data for selected model testing for day *t*. 38

Fig. 3.1. Daily Accuracy and F1 score of Bi-LSTM on the holdout dataset 46
Fig. 3.2. Daily Precision and Recall of Bi-LSTM on the holdout dataset 47
Fig. 3.3. Money, expressed in μBTC, after investing for 15 days depending on the probability thresholds B_T for buying and B_T for holding the cryptocurrency with the initial investment of 1.000 μBTC. The Period for investment is from November 1st, 2021, until November 15th, 2021. 48
Fig. 3.4. Trading strategy daily investment value expressed in μBTC. Trading ended with 973 μBTC with a starting budget of 1,000 μBTC. November 15th, 2021 - December 31st, 2021 49

Fig. A.1. Cryptocurrencies with mid-trade volume price comparison in USD and μBTC . . . 61
Fig. A.2. Sample of cryptocurrencies label amount with 0% threshold for the price change in the upcoming 7 days 62
Fig. A.3. Sample of cryptocurrencies label amount with 5% threshold for the price change in the upcoming 7 days 63
Fig. A.4. Sample of cryptocurrencies label amount with 20% threshold for the price change in the upcoming 7 days 64
Fig. A.5. Sample of cryptocurrencies label amount with 0% threshold for the price change in the upcoming 7 days 65

Fig. A.6. Monthly labeled (as per equation 2.2) Reddit posts count per subreddit 65
Fig. A.7. Monthly labeled (as per equation 2.2) Reddit posts count per label 66

List of Tables

Table 1.1. Table from Zhai et al. [5, p. 1092] paper, showing an accuracy increase due to combining price information and news data . . . 16
Table 1.2. Table from Ullah et al. [6, p. 7] paper, comparing various classification methods on accuracy for stock prediction . . . 17
Table 1.3. Table from Yang et al. [7, p. 7] paper, comparing XLNet and BERT on different text data sources . . . 20
Table 1.4. Other related work . . . 24

Table 2.1. Example of random Reddit text messages . . . 28
Table 2.2. Tain-test split amount of labeled data . . . 30
Table 2.3. List of selected cryptocurrencies for the model training data . . . 34

Table 3.1. Hyperparameters search results with the performance of differnt models . . . 40
Table 3.2. Performance of fine-tuned BigBird-RoBERTa models on different timeframes on validation dataset . . . 41
Table 3.3. Averages of fine-tuned BigBird-RoBERTa models on different timeframes performance mentrics (reference for the detailed table 3.2). Weighted average is calculated based on validation dataset size . . . 41
Table 3.4. Example of wrong predictions in Reddit text messages classification . . . 42
Table 3.5. GPT-3.5 sentiments crossed with sentiments from fine-tuned BigBird-RoBERTa model . . . 42
Table 3.6. Randomm Forest best hyperparameters after hyperparameter search . . . 43
Table 3.7. Gradient Boosting best hyperparameters after hyperparameter search . . . 43
Table 3.8. Comparing model predictions on Validation dataset (CV Accuracy) and other metrics on Test dataset given different set of features. RF refers to Random Forest algorithm, GB - Gradient Boosting. . . . 44
Table 3.9. Bi-LSTM and Random Forest performance metrics comparison on the test data for TA_GPT model . . . 46
Table 3.10. Performance of Bi-LSTM on the holdout data . . . 46
Table 3.11. Daily performance of Bi-LSTM on the holdout data, summary statistics . . . 47
Table 3.12. Summary statistics of trading strategy investment expressed in μBTC. Trading ended with 973 μBTC with the starting budget of 1,000 μBTC. November 15th, 2021 - December 31st, 2021 . . . 49

Introduction

Cryptocurrencies have changed the financial markets. Everything began with Bitcoin, and now there are over 9.900 different cryptocurrencies worldwide[1]. Their acceptance by businesses is steadily growing, reflecting the increasing number of companies accepting Bitcoin and other cryptocurrencies as a valid form of payment. The large volatility of cryptocurrency markets not only carries huge investment risks but a potential for big returns as well. This volatility attracts both investors and researchers. This thesis aims to explore the influence of sentiment analysis on social media, specifically Reddit, combined with technical indicators and machine learning techniques in predicting the price movement direction of various cryptocurrencies.

While traditional technical indicators are well-researched in traditional financial markets, their application is not that largely explored in the cryptocurrency space. A similar statement holds for social media. Reddit, a social media platform with a large database of user-generated content, with users' views on different cryptocurrencies, could influence the price, as already seen in multiple pieces of research exploring various social media platforms' impact on traditional financial markets. Therefore, a considerable part of this research is dedicated to understanding sentiments expressed in Reddit posts.

This study employs two distinct classification machine learning algorithms - one for deducing the sentiment of Reddit messages and another for predicting the direction of cryptocurrency price movements. For both tasks, hyperparameter tuning strategies are employed, and several algorithms are tested to identify the optimal one.

Transformers - BERT, XLNet, and BigBird-RoBERTa - pre-trained on a large corpus of documents, are fine-tuned to predict the sentiment of the Reddit messages. These models often outperform classical machine learning methods that learn exclusively from the provided data. Additionally, OpenAI's GPT-3.5 - a neural network-based large language model - is also evaluated for sentiment analysis as well.

For the price movement direction prediction, two traditional machine learning models - Random Forest and Gradient Boosting - are tested with various feature sets. The objective is to ascertain if predicted Reddit sentiments have a notable impact on the model's performance. Subsequently, a recurrent neural network (Bi-LSTM) is evaluated in an effort to improve the performance of Random Forest and Gradient Boosting.

Finally, a straightforward hypothetical trading strategy is proposed and backtested on the holdout data. This strategy uses the optimal combination of predicted Reddit sentiments and the price movement direction. However, this thesis does not intend to present a foolproof model for predicting cryptocurrency prices. Instead, it seeks to explore and highlight new methods and strategies. The insights and methods presented here could serve as a base for further research and refinement, possibly leading to more powerful models in the future.

[1] source: https://currency.com/how-many-cryptocurrencies-are-there

Thesis Research Problem. Despite the many studies addressing the predictability of cryptocurrency prices, few have tried to study the combined effects of sentiment analysis from social media, especially Reddit, technical indicators, and machine learning models on multiple cryptocurrencies. This thesis aims to fill this gap in research with the theory that a combined approach could result in a more powerful and accurate predictive model.

Research Question. How effective is the combined use of sentiment analysis from Reddit posts and technical indicators in predicting future cryptocurrency prices movement using machine learning models?

Object of Research. The object of this research is the volatile cryptocurrency market, with a special focus on predicting price changes using a varied approach that includes sentiment analysis, technical indicators, and machine learning.

Tasks Implementing the Research Objective:

1. Data collection - cryptocurrency price history and Reddit messages
2. Predicting the sentiment of Reddit messages by using the optimal methodology
3. Processing, cleaning, and transforming collected data into usable features
4. Using machine learning models, like Random Forest, Gradient Boosting, and Bi-LSTM, for the prediction of cryptocurrency price movement direction
5. Determining if Reddit sentiments has a notable impact to the price movement prediction
6. Implementing a simple trading strategy for the proposed machine learning model

1. Literature Review

1.1. Introduction

Cryptocurrencies are a form of digital assets built using cryptographic technologies. It started by the *Bitcoin* creator Nakamoto [8]. Bitcoin, the pioneer of all cryptocurrencies, was established in 2009, and since then, the market capitalization of cryptocurrencies has surged, reaching $800 billion at the end of 2022 according to CoinMarketCap [1]. The unique features of cryptocurrencies, such as decentralization, pseudo-anonymity, and global availability, have increased adoption and investment in Bitcoin [9, 10].

Moreover, *Bitcoin* price is sensitive to the media (Bouoiyour and Selmi [11]). Kim et al. [12] analyzed online communities' impact on the volume of the three biggest cryptocurrencies and concluded that a number of transactions are impacted by the user's comments and replies in those communities.

The rise of social media transformed how individuals consume news (including financial news), so more researchers examined its effect on the financial market. For instance, Bollen et al. [13] investigated Twitter messages' sentiment on the Dow Jones Industrial Average (DJIA) index, concluding that sentiment plays a significant role in predicting DJIA index movement.

1.2. Short history of cryptocurrency

Pre-Bitcoin. Before even Bitcoin was introduced to public by Nakamoto [8] in 2008, other attempts for *digital money* already existed. Cryptographer Chaum back in 1983 published a paper describing the first cryptocurrency, which was privacy-focused. eCash was the first fully electronic money found in 1993, yet it went bankrupt in 1998 after failing to land deals from big partners [15]. Later there were more attempts - E-Gold, B-Money, Hashcash [16].

Although Bitcoin is not the oldest cryptocurrency, it is the oldest and currently tradable one.

Bitcoin was introduced in 2008 by the author named Nakamoto [8], which identity is held in secret [2] (it is not even known if it was a single person or a group of people). On January 3rd, 2009, he mined the first block of the Bitcoin network (it resulted in 50 bitcoins).

The first sale using Bitcoin happened in 2010 with the purchase of two pizzas for 10 000 Bitcoins.

Ethereum network - the second largest cryptocurrency to date, was launched not long after Bitcoin - on July 30th, 2015.

[2] as of April 2023

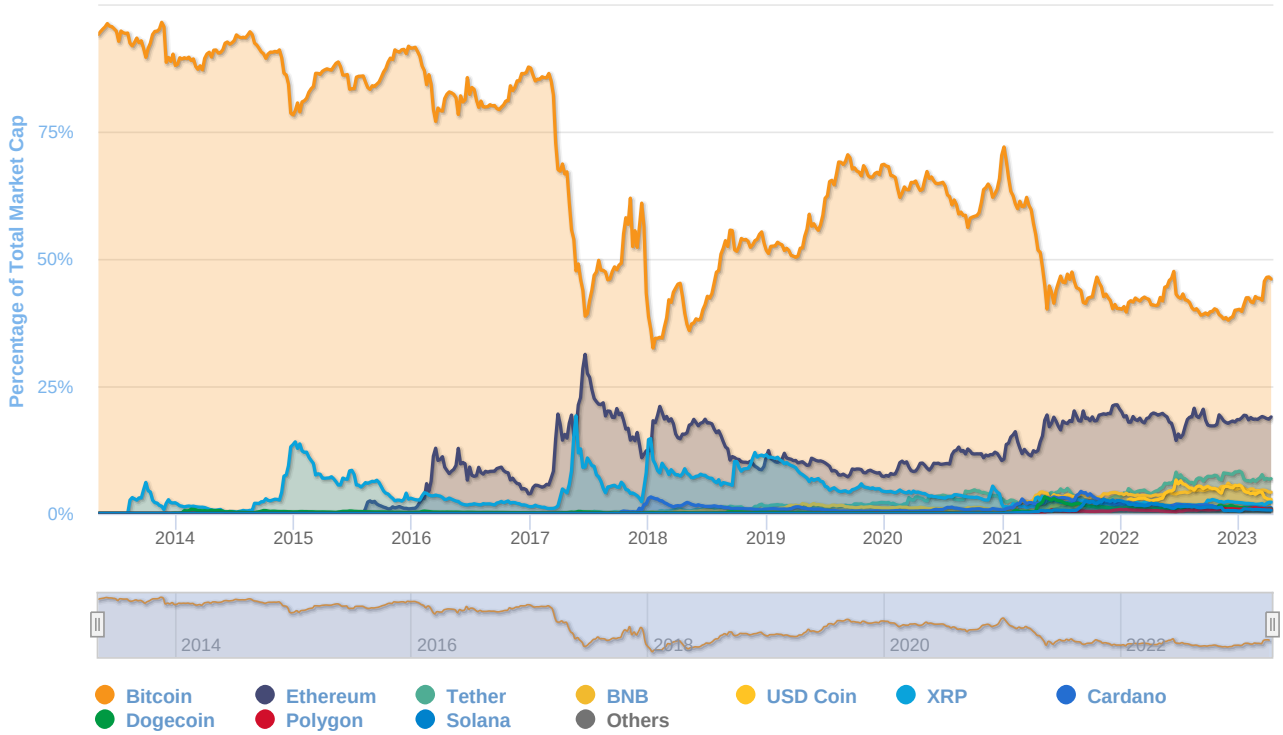


Figure 1.1: Major Cryptoassets By Percentage of Total Market Capitalization (Bitcoin Dominance Chart), May 5, 2013 - Apr 15, 2023. CoinMarketCap [1]

Nowadays Bitcoin price increased from a few cents in 2010 to over $60.000 in 2021, influencing the creation of various new cryptocurrencies. On April 2023, Bitcoin still is the leading cryptocurrency, and it never gave up its number-one status. Currently, it has over 40% of the cryptocurrency market share, while Ethereum stands in the solid second place with a little bit below 20% share. The top 10 cryptocurrencies occupy over 80% of the market (see Figure 1.1 for more details).

1.3. Technical indicators in predicting price movement

Technical indicators are calculated from historical trading data using mathematical functions, which traders use to make better trading decisions. Some examples of trading indicators are moving averages (MA), relative strength index (RSI), momentum (MOM), etc. The following descriptions of technical indicators are based on the book *Technical Analysis for Algorithmic Pattern Recognition* by Tsinaslanidis and Zapranis [17]:

- **Moving averages (MA)** are the averages of past values in time series. Moving averages can be weighted or not. The main purpose of MA is to smooth the changes in price in a time series.
  - **Simple moving average (SMA)** is just a simple rolling average of the time series, defined by the function

$$SMA_{t|w} = \frac{P_t + P_{t-1} + \ldots + P_{t-w+1}}{w}, \forall t \in [w, \ell],$$

    where $P_t \equiv \{P_1, P_2, \ldots, P_\ell\}$ is the price series length ℓ , ℓ - the last price in the series and w - the size of the rolling window. The rolling window size w is chosen depending on whether the long-term or short-term trend is analyzed.

– **Linearly Weighted Moving Average (LWMA)** is very similar to SMA, just having higher weight for more recent data, and is expressed as:

$$LWMA_{t|w} = \frac{wP_t + (w-1)P_{t-1} + \ldots + 2P_{t-w+2} + P_{t-w+1}}{w(w+1)/2}, \forall t \in [w, \ell].$$

– **Exponential Moving Average (EMA)** gives more weight to recent data points, so it becomes more important in trend changes than LWMA. Still, on the other hand, it is more sensitive to fluctuations. EMA is expressed as:

$$EMA_{t-1|w,\lambda} = (1-\lambda)EMA_{t-1|w,\lambda} + \lambda P_t, \forall t \in [w+1, \ell],$$

where $0 \leq \lambda \leq 1$ and starting value of $EMA_{t-1|w,\lambda}$ is the same as $SMA_{w|w}$. λ is set depending on how much importance for the recent data is wanted to have.

- **Moving Averages Crossovers (MAC)** indicates the crossing of a longer and shorter period moving average and can be calculated as follow:

$$MAC_{t|w_S,w_L} = SMA_{t|w_S} - SMA_{t|w_L}, \quad \forall t \in [w_L, \ell]$$

MAC can generate a trading signal (an indicator that tells if an investor should buy or sell a security) depending on the value it gets

- **Relative Strength Index (RSI)** analyses variation and pace of price movement. It ranges between 0 and 100 and measures average gains to the average losses ratio over a time period. Calculation of RSI requires a few steps - first of all, upward (ΔP_t^+) and downward (ΔP_t^-) changes need to be detected, and they are defined as:

$$\Delta P_t^+ = \max(P_t - P_{t-1}, 0)$$
$$\Delta P_t^- = |\min(P_t - P_{t-1}, 0)|.$$

Then a *Relative Strength (RS)* at time t for the w time period is defined by:

$$RS_{t|w} = \frac{\sum_{i=t-w+1}^{t} \Delta P_t^+}{\sum_{i=t-w+1}^{t} \Delta P_t^-}, \forall t \in [w+1, \ell].$$

Finally, $RSI_{t|w}$ is a simple function:

$$RS_{t|w} = \begin{cases} 100, & if \sum_{i=t-w+1}^{t} \Delta P_t^- = 0 \\ 100 - \frac{100}{1+RS_{t|w}}, & otherwise \end{cases}.$$

There are various methods how interpreting RSI; for example, it is recommended to purchase a security if RSI crosses the pre-defined level while increasing and sell - if it crosses it while decreasing.

Table 1.1: Table from Zhai et al. [5, p. 1092] paper, showing an accuracy increase due to combining price information and news data

Data sets	Accuracy (%)
Price	58.8
Direct news	62.5
Indirect news	50.0
Combined news	64.7
Price & News	70.1

More technical indicators are categorized into five segments: trend, mean reversion, relative strength, volume, and momentum[3]. Some studies use technical indicators to predict securities' price or price movement by using them as features in machine learning techniques. For example, Mudassir et al. [18] used ANN, SANN, SVM, and LSTM to predict Bitcoin price as a regression task and as a classification task with over 700 features based on technical indicators at the beginning. The F-Score of classification ranged from 0.56 to 0.71 depending on the model and how many days ahead the price change was calculated. Another study, published by Akyildirim et al. [19], analyzed 12 liquid cryptocurrencies' on a daily and minute level by using machine learning algorithms (SVM, Logistic regression, ANN, and RF) for the classification of a price direction. They also used various technical indicators as features, and the predictive accuracy was around 55–65%. Shynkevich et al. [20] uses just technical indicators and SVM, ANN, and kNN machine learning techniques for stock price movement prediction and getting significant results; similarly Yun et al. [21] as well predicts stock movement with just technical indicators using GA-XGBoost algorithm. Zhai et al. [5] also added news data to the technical indicators (combined), which increased the accuracy of the prediction (see the table 1.1).

1.4. Machine Learning and Statistical Methods to predict price

Techniques used for cryptocurrency and traditional financial markets prediction are similar. It ranges from the econometric approach, where researchers can quickly build explainable models. Time series analysis is often used, such as ARIMA.

Yet, ML has emerged, and it is also used for financial market price prediction. Here are some of the ML techniques which can be used to predict financial markets:

- **Support Vector Machines (SVM)**, proposed by Cortes and Vapnik [22], is a supervised machine learning algorithm. The algorithm is used for classification and regression tasks. SVM's goal is to find an optimal hyperplane that separates different classes of data points with the maximum distance between them. This distance between the hyperplane and the nearest data points from each class is called the margin. Data points on the edge of the margin are support vectors. SVM can handle linear and non-linear data by using kernel functions. Kernel functions transform non-linear data to higher-dimensional space, making it linearly separable. With linearly separable data points, SVM can find an optimal hyperplane. SVM effectively handles high-dimensional data and performs well with less training data.

[3] investopedia.com

Table 1.2: Table from Ullah et al. [6, p. 7] paper, comparing various classification methods on accuracy for stock prediction

Name of the Algorithm	Test accuracy
Naive Bayes(NB)	51.21%
Logistic Regression(LR)	51.77%
Stochastic Gradient Descent (SGDC)	50.56%
Support Vector Machine (SVM)	54.06%
Adaboost	53.29%
Random Forest	52.43%
Ensemble 1 (predict top and bottom)	99.25%
Ensemble 2 (predict top and bottom)	74.23%

Multiple publications are using SVM to predict financial markets. Huang et al. [23] used it to predict NIKKEI 225 index weekly movement with the economic variables in the dataset, such as gross domestic product (GDP), consumer price index (CPI), and more. Results showed SVM outperforming random walk (RW), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and Elman Back-propagation neural networks (EBNN) in terms of the Hit ratio [4]. Kara et al. [24] used technical indicators for price movement prediction of the Istanbul Stock Exchange (ISE) National 100 Index. They compared SVM performance to ANN and concluded that ANN performs significantly better. Some researchers used SVM to classify price movement by using just textual data as an input; for example, Wang and Zhao [25] explored this method and found that news influences investors in stock decision-making (using just news data for stock prediction), labeling data simply by *rising* and *falling*, getting an accuracy score of 58% [25, p. 198]. Panwar et al. [26] compared SVM to logistic regression and got better results with the SVM model [26, p. 1933]). Ullah et al. [6] compared SVM with other methods for a similar classification problem, and SVM was leading there as well (see table 1.2)

- **Artificial neural network (ANN)** inspiration was the structure and functionality of biological neural networks. ANNs consist of nodes (also called neurons), which are organized in layers. The algorithm process and transmit information through weighted connections between these nodes. These weights are adjusted during the learning phase with optimization algorithms like gradient descent to minimize errors between actual and predicted data. ANNs do not require a linear relationship between input and output variables, making the algorithm useful in many tasks, such as regression, classification, or pattern recognition.

  Already mentioned Kara et al. [24] showed ANN over-performing SVM.Vijh et al. [27] predicted the stock price returns of five different companies. As data input, they used time series of stock prices, including generated variables, for example, moving averages or differences in historical price changes (kind of the basics of technical indicators). They used ANN and compared the results to Random Forest (RF). Results showed ANN performing better regarding all compared performance metrics: RMSE, MAPE, and MBE.

- **Random Forest (RF)** creates a lot of decision trees during the training process and then combines these trees' predictions to get more accurate results. Single decision trees tend to overfit

[4] usually referred to as Recall in classification: $TP/(TP+FN)$, where *TP - True Positive*, *FN - False Negative*

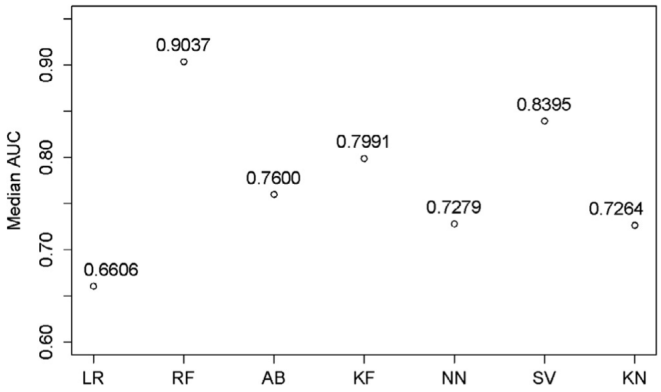


Figure 1.2: Various models for long-term stock price prediction of 5767 European companies performance measured in median AUC, Ballings et al. [2]

training data, so combining multiple trees helps to deal with it. Trees in RF are built independently one from another - it is achieved by giving a random subset of data points and features at each split. As a result, diverse trees are created, and these decision trees can capture different aspects of data. Then, the majority voting (if it is a classification task) or an average of outputs (if it is a regression) is performed to make a final prediction. The random forest can handle high-dimensional data. It is somewhat resistant to overfitting and is applied for various classification and regression tasks, including financial markets prediction. Already mentioned Vijh et al. [27] showed ANN performing better than Random Forest in short-term stock price prediction. Meanwhile, Ballings et al. [2] analyzed models for long-term stock price prediction of 5767 European companies. They used various variables from company books as input, such as cash flow yield, company size, financial indicators, and more. Ballings et al. measured the performance of Support Vector Machines (SVM), Neural Networks (NN), Logistic Regression (LR), K-nearest neighbors (KN), AdaBoost (AB), Kernel Factory (KF), and Random Forest (RF). Results, measured as AUC median, for long-term stock prediction, there were different - Random Forest, not just outperformed Neural Network [5]. Still, it was the number one performer, followed by SVM, KF, and all the rest (see figure 1.2 for complete comparison).

- **Deep learning (DL)** focuses on ANN with multiple layers (deep neural networks). They can automatically learn complex features from input data, making feature engineering less relevant. Deep learning can be applied to various tasks, such as natural language processing, image or speech recognition. A few examples of deep learning algorithms are Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Long Short Term Memory Networks (LSTMs).

  Deep learning is applied in financial markets predictions as well. LSTM and CNN are the top ones in published papers predicting stock and forex exchange based on Hu et al. [28] survey Mehtab et al. [29] analyzed NYFTY 50, a benchmark of the Indian stock market index, comparing machine learning and deep learning methods. The dataset contained stock index time series and engineered features from it. Mehtab et al. evaluated the performance of models using RMSE and correlation between actual and predicted values. They compared eight machine learning models to deep learning models (4 different variations of LSTMs). Out of the machine

[5] author calls Artificial Neural Network (ANN) as Neural Network (NN) in the paper

learning models, the best performance, based on two measures mentioned before, were multivariate regression and Random forest. Yet, all variations of LSTMs outperformed machine learning models by a huge margin measuring RMSE. Vargas et al. [30] used deep learning methods (RNN, RCNN) to predict stock price movement by combining both news data and technical indicators, with the best method being SI-RCNN and achieving 56.84% accuracy score [30, p. 5]. Nelson et al. [31] explored LTSM in the same matter.

- **Reinforcement learning (RL)** uses training agents to make decisions based on interactions with their environment. It differs from supervised learning, as RL learns through trial and error - by optimizing its actions to achieve the target. The system rewards an agent's actions, where negative rewards indicate undesirable actions and positive rewards are given for correct behavior. RL learns by exploration and exploitation, where exploration encourages the agent to try new actions and learn from mistakes or successes. In contrast, exploration gains knowledge and makes better decisions based on it: balancing these two aspects in reinforcement learning is essential. RL found its usage in various applications, such as games and robotics. While deep learning focuses on feature extraction through deep neural networks, RL learns optimal decision-making strategies and can be used without labeled data. Unlike previously discussed systems, Reinforcement learning for trading is used for trading strategy. Wu et al. [32] used Gated Recurrent Unit (GRU) to get features from the data of the U.S., the U.K., and the Chinese stock markets. These features were used in Reinforcement learning to make trading decisions. The authors proposed Gated Deep Q-learning (GDQN) and Gated Deterministic Policy Gradient (GDPG) trading strategies. They compared the results to the state-of-the-art direct reinforcement learning (DRL) strategy. All trading strategies achieved profitability (yet it is not compared to some benchmarks, e.g., S&P 500). DRL had losses on some symbols and generally was less stable and had fewer signals to buy and sell the stock compared to GDQN and GDPG. The former two methods were profitable, with GDPG being more stable.

1.5. Sentiment analysis methods

Sentiment analysis methods can be classified as Natural Language Processing, Sentiment lexicon, Machine Learning, and Deep Learning techniques. This section briefly reviews these listed methods.

Natural Language Processing (NLP) plays a main role in translating human language to one computer can understand by processing and analyzing textual data. Therefore, it is crucial in sentiment analysis. A few of the NLP tasks are [33]:

- Text manipulation - first of all, tokenization, which breaks down text into words (and words are called tokens). Another part is stemming, which truncates the words to their root form, so words having the same root become matching ones: for example, words *cryptocurrency* and *crypto* become *crypto*). Finally, there is a lemmatization, which converts the words to their dictionary form. For example, the lemma of the word *cryptocurrencies* is *cryptocurrency*.
- Part of speech tagging (POS) assigns a part of speech to the token (for example, noun, verb, adverb, etc.). Adjectives and adverbs are essential in understanding sentiment, so POS helps to

Table 1.3: Table from Yang et al. [7, p. 7] paper, comparing XLNet and BERT on different text data sources

Model	SQuAD1.1	SQuAD2.0	RACE	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B
BERT-Large (Best of 3)	86.7/92.8	82.8/85.5	75.1	87.3	93.0	91.4	74.0	94.0	88.7	63.7	90.2
XLNet-Large-wikibooks	88.2/94.0	85.1/87.8	77.4	88.4	93.9	91.8	81.2	94.4	90.0	65.2	91.1

find them.

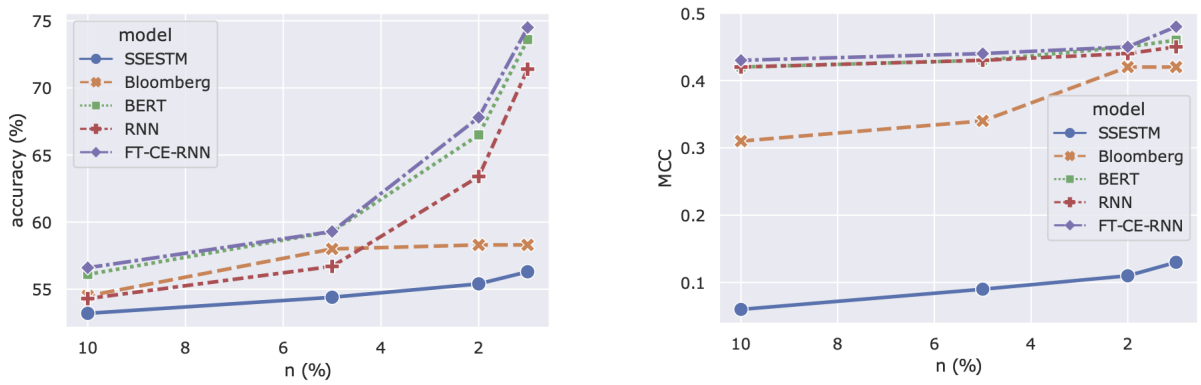
- Named entity recognition (NER) identifies entities in the text, such as locations, organizations, etc.

Sentiment lexicons is another approach to use for sentiment analysis. In short form, it is a list of words and their sentiment score. Lexicons can be manually made - it is mainly used in cases where some specific domain knowledge for sentiment is needed. Still, it requires analyzing domain-specific text and assigning sentiment scores to sentiment-related words [34]. The list and scores can also be built automatically, for example, using a Graph model as presented by Widdows and Dorow [35]. Pre-built lexicons, based on advanced research, can be used for sentiment analysis relatively easily. Some of such pre-built lexicons are SentiWordNet, which is free for non-profit research [36], or VADER, designed for sentiment in social media text [37].

Machine learning is used for sentiment analysis as well. Broadly, it can be divided into unsupervised, supervised, and deep learning techniques. Unsupervised machine learning, for example, clustering or topic modeling, does not need labeled data. They use patterns detected in the data to make groups of similar text. Topic modeling (for example, Latent Dirichlet Allocation (LDA), proposed by Blei et al. [38]) cluster text into topics, and these topics can be associated with sentiments. Supervised algorithms, on the other hand, require labeled data. Some examples of supervised algorithms are Naive Bayes, K-Nearest Neighbors (Naive Bayes showed better results for movie and hotels reviews by Dey et al. [39]), Support Vector Machines (for example, Ahmad et al. [40] analyzed Twitter messages sentiment), and Random Forest (Fauzi [41] analyzed review data using Random Forest) - they use various features from the text (for example a number of specific words) to make predictions.

Deep learning methods, such as word embeddings, Recurrent Neural Networks, Convolutional neural networks, etc. [42]. Word embeddings represent vectors of words that have their semantic meaning and enable better sentiment analysis considering the relationships among words. Word2Vec, proposed by Mikolov et al. [43], is an example of such a word embedding method.

Another deep learning method is transformers: a pre-trained sentence or word embedding. One of the methods is Bidirectional Encoder Representations from Transformers (**BERT**), published by Devlin et al. [44] and applied to English Google search queries in 2019 [45]. While Sonkiya et al. [46] used BERT as a feature for stock price prediction, Chen [3] solved market movements as a classification



(a) Accuracy **(b)** MCC

Figure 1.3: Results from Chen [3, p. 14] paper on used classification methods accuracy and MCC, where *n* denotes a percentile of extreme news used from classification

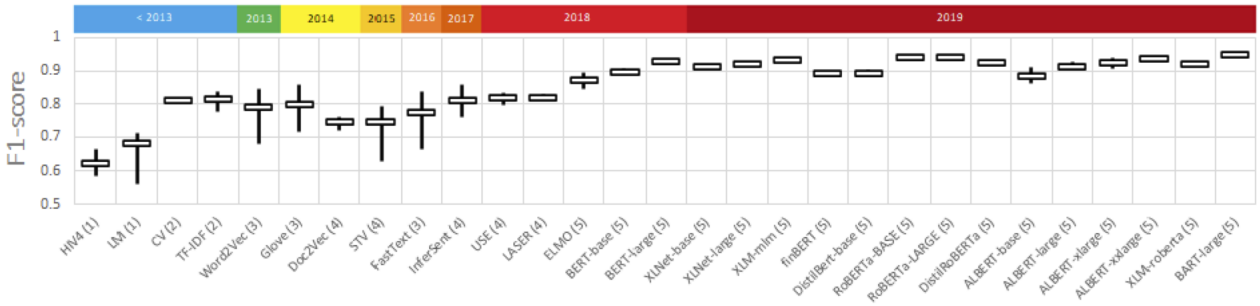


Figure 1.4: Mishev et al. [4] chart of sentiment analysis models F1-score for financial phrase dataset

problem. Chen got a very high accuracy score on extreme results (top 1% and top 2% of extreme probability news) simply using the BERT classifier (see figure 1.3). **XLNet** is another Transformer method, and, based on its author's Yang et al. [7] publication, "outperforms BERT on 20 tasks, often by a large margin" (see table 1.3). Yet, there are few, if any, easily findable publications where someone has attempted to predict the stock market using predicted sentiment by fine-tuning XLNet. To summarize, Mishev et al. [4] showed that transformers outperformed all other methods, including word and sentence embeddings, on financial phrases text by a huge margin, as shown in Figure 1.4.

1.6. Sentiment Analysis and its Role in Financial Markets

Sentiment analysis is a process to identify or categorize opinions written in text, most commonly to know if that opinion is positive, neutral, or negative. Sentiment analysis gained popularity in various domains due to the ability to extract people's emotions and opinions. Among the various uses of sentiment analysis, including marketing, where sentiment helps to understand customer feedback; public relations, where it allows companies to understand their reputation; politics, where sentiment analysis helps to understand public opinion on various issues or policies, sentiment analysis gained its popularity in finance, where it helps to make an investment decision.

In finance, it is usually combined with other data to create trading strategies. Moreover, sentiments are extracted not just from text created by regular people but also from the news. So, sentiment analysis can be used to analyze news and social media posts to identify views on specific stocks,

cryptocurrencies, or the whole market.

1.7. Social Media Sentiment and Financial Markets

Financial markets, including stocks, have way more analysis using sentiment in the news and social media than cryptocurrency. Since stocks and cryptocurrency can be viewed as investments, the literature review of the financial markets is beneficial.

1.7.1. Early Studies on the Role of Internet Discussions and Stock Market.

Antweiler and Frank [47] conducted one of the earliest studies on the relationship between online discussions and financial markets. The authors investigated the information content of internet stock message boards - Yahoo! Finance and Raging Bull - and their impact on stock prices and trading volume. Their findings showed that the messages on these boards are important in predicting stock volatility and trading volume.

1.7.2. Twitter Sentiment and Stock Market

Twitter is a popular platform for studying the relationship between social media sentiment and financial markets. Bollen et al. [13] were among the first ones in this area, showing a connection between Twitter sentiment and the Dow Jones Industrial Average (DJIA) index. They analyzed a large dataset of tweets and found that the Twitter messages sentiment was significant in the index's price movement.

Expanding on this, Rao and Srivastava [48] back in 2012 proposed a model that uses sentiment analysis on tweets (they analyzed more than 4 million tweets) to predict stock market trends. They detected a high correlation between tweets and stock returns. Similarly, Ranco et al. [49] investigated the relations between Twitter sentiment and stock price returns for the 30 companies forming Dow Jones Industrial Average (DJIA) index. The study found that the sentiment expressed in tweets had an abnormal influence on stock returns. Another study by Si et al. [50] used Twitter data to create a topic-based sentiment analysis model predicting S&P100 index price using sentiment time series from Twitter data. Their method outperformed traditional, non-topic-based, time-series models.

1.7.3. Reddit Sentiment and Financial Market

Research papers focusing on Reddit sentiment for stock prediction are limited compared to other social media, such as Twitter and Facebook.

Chacon et al. [51] built a trading strategy a person reading Reddit subreddit WallStreetBets would use - they tried to mimic human behavior. They observed the positive relationship between the number of messages in the subreddit and the stock's trading volume, but the final model was not profitable.

Recently, more studies have been focusing on specific so-called *meme stocks* and their relation to Reddit messages, examining the impact of public sentiment on the fluctuation of these stocks. Car-

vajal [52] model for AMC, GME and NOK was not profitable. Long et al. [53] also did not find the impact of sentiment averages on the GME stock performance. They noticed stronger ties between NET sentiment and stock on the Bullish market.

1.8. Other related work

Sentiment analysis has been used to get additional information for financial asset predictions in many text channels beyond social media, like news data, various stock forms, and in many other techniques than discussed before. This section briefly overviews other related work, shown in the table 1.4.

Rao et al. [54] adopted the EMMS and time series approach to forecast one day ahead, utilizing Naive Bayesian classification method for sentiment analysis. They found that Twitter significantly influences stock prices such as DJIA and NASDAQ-100.

In the work by Khedr et al. [55], the K-nearest neighbor was used to predict price direction movement and Naive Bayes for sentiment analysis. They found that news data assists in improving the prediction for stocks such as Yahoo Inc, Microsoft Corporation MSFT, and Facebook Inc (FB Inc).

Mittal and Goel [56] employed Self Organizing Fuzzy Neural Networks (SOFNN) for price direction movement and a custom multi-label solution for sentiment analysis. They concluded that the mood in Twitter messages affects DJIA prices.

Sousa et al. [57] explored the relationship between stock price movement and sentiment without making a prediction, using BERT with manually labeled news for fine-tuning. They reported that BERT outperformed convolutional neural networks, and DJI movement and news sentiment were consistent just for the few days in the analyzed set.

Abraham et al. [58] used linear regression, with Google trends data as an additional variable, and VADER sentiment analysis. They found that the volume of tweets, but not sentiment, is a predictor for the price direction of cryptocurrencies BTC and ETH.

Valencia et al. [59] used MLP, SVM, and RF with VADER sentiment analysis, demonstrating that different models work better for different cryptocurrencies.

Kim et al. [60] employed Hidden Markov model (HMM) for prediction, finding that social sentiment is more relevant during a bull market for BTC.

In the work by Parekh et al. [61], DL-GuesS was used to predict price one day ahead and VADER sentiment analysis was used. They concluded that DL-GuesS outperformed traditional systems for predicting cryptocurrency prices.

Jing et al. [62] used LSTM for predicting price one day ahead and CNN for sentiment analysis. They found that including sentiment in a model improved performance for various stocks in Shanghai Stock Exchange.

Jin et al. [63] predicted daily close price using LSTM and CNN with word2vec for sentiment analysis. They found that their proposed model - LSTM with sentiment index - performed better than compared

models for predicting Apple's stock price.

Lastly, Wu et al. [64] used LSTM model with sentiment index and technical indicators for closing price prediction and CNN for sentiment analysis. They concluded that sentiment impacts price prediction for stocks in the China Shanghai A-share market.

Table 1.4: Other related work

Reference	Methodology for prediction	Methodology for sentiment analysis	Text channel	Main assets	Results
Rao et al. [54] 2012	EMMS, time series based, forecast one day ahead	Naive Bayesian classification method: accuracy of about 82.7%	Twitter	Stocks: DJIA, NASDAQ-100, 13 big cap technological stocks	Based on Granger's Causality Analysis, Twitter greatly affects stock prices
Khedr et al. [55] 2017	Price direction movement, K-nearest neighbor	Naive Bayes	News data set	Stocks: 3 companies - yahoo Inc, Microsoft Corporation MSFT, and Facebook Inc (FB Inc)	News data helps to improve the prediction
Mittal and Goel [56] 2012	Price direction movement, Self Organizing Fuzzy Neural Networks (SOFNN)	Custom multi-label solution	Twitter	Stock: DowJonesIndustrialAverage(DJIA)	The mood in Twitter messages affects DJIA prices
Sousa et al. [57] 2019	No prediction, but the relationship between stock price movement and sentiment	BERT with manually labeled news for fine-tuning	News: financial news sources	Stock: Dow Jones Industrial (DJI) Index	BERT outperformed convolutional neural networks, DJI movement, and news sentiment consistent just for the few days in the analyzed set
Abraham et al. [58] 2018	Linear regression, google trends data as an additional variable	VADER sentiment analysis	Twitter	Crypto: BTC and ETH	Tweets volume, but not sentiment, is a predictor for the price direction
Valencia et al. [59] 2019	MLP, SVM, RF	VADER sentiment analysis	Twitter	Crypto: Bitcoin, Ethereum, Ripple, and Litecoin	Different models work better for different cryptocurrencies

Kim et al. [60] 2022	Hidden Markov model (HMM)	Not specified	Twitter	Crypto: BTC	The social sentiment is more relevant during bull market
Parekh et al. [61] 2022	Price one day ahead, DL-GuesS	VADER sentiment analysis	Twitter	Crypto: BTC, LTC, Dash, Bitcoin-cash	DL-GuesS outperformed mentioned traditional systems
Jing et al. [62] 2021	Price one day ahead, LSTM	CNN	Chinese posts in the stock forum	Stocks: six industries from Shanghai Stock Exchange (SSE), randomly selected stocks from each industry	Including sentiment in a model improved performance
Jin et al. [63] 2020	Daily close price, LSTM	CNN with word2vec, trained on users' labels from StockTwits	Yahoo Finance comments used for prediction	Stocks: Apple	Proposed model - LSTM with sentiment index - performed better than compared ones
Wu et al. [64] 2022	Closing price prediction, LSTM model with sentiment index and technical indicators	CNN, labeling based on price change after news release	News headlines and stock forum posts	Stock: China Shanghai A-share market, five listed companies	Sentiment impacts price prediction

1.9. Summary

The domain of cryptocurrency price prediction is witnessing an increase in interest, necessitating comprehensive research that integrates traditional financial indicators, machine learning techniques, and sentiment analysis. The dynamics of cryptocurrency markets, which trade continuously and are impacted by regulatory influences, requires research tailored to the specifics of cryptocurrencies. Furthermore, the research will explore the profound influence of Reddit - a social media - sentiments, on the cryptocurrency market. In light of the limited studies focusing on cryptocurrency markets compared to traditional ones, this research aims to fill the gap by utilizing a comprehensive approach that involves technical indicators and sentiment analysis. Consequently, this endeavor promises to enhance our understanding of the cryptocurrency market, aid investors in decision-making, and promote further research and developments in this growing field.

2. Methodology

2.1. Data collection and short overview

2.1.1. Cryptocurrency price data

Cryptocurrency price data is taken from Binance, one of the leading cryptocurrency exchanges, by using its API [6]. It is daily time series, where *Open price*, *High price*, *Low price*, *Close price*, *Volume*, and *Number of trades* was taken from it.

It contains data on 353 cryptocurrencies for two years, 2020 and 2021. Some of the cryptocurrencies are tied to the value of fiat currency (called *stablecoins*) - such, so a hardcoded list of stablecoins was excluded from the analysis (*BUSD*, *USDS*, and *USDSB*). Then, since all the other cryptocurrencies (called *altcoins*) are correlated to the price of Bitcoin Meynkhard [65], the price of each altcoin is expressed in micro-bitcoin (μBTC also called *bit*), equal to 0.000001 BTC. By doing so, it is expected to reduce the impact of market behavior: Figure 2.1 demonstrates the difference among the most traded cryptocurrencies (as expressed in the value of trades in USD), where it is seen that cryptocurrencies are growing in the price of USD but less in μBTC (especially *XRP*). Fig. A.1 shows the same for the less traded cryptocurrencies.

2.1.2. Reddit data

Reddit data was extracted by using *Pusshift API* [66] for the same period (the year 2020 and 2021) from two subreddits *r/CryptoCurrency* and *r/wallstreetbets*. Posts were extracted if they mentioned any cryptocurrency from the Binance dataset and its variations (e.g., for Bitcoin, it was any of *BTC*, *Bitcoin*, or *bitcoin*). These cryptocurrency names variations were taken by using CoinMarketCap API [7], one of the most well-known cryptocurrency price tracking websites. Amount of posts totaled 1.290.577, yet, *Pusshift API* extracted everything, even if part of the word matched the cryptocurrency symbol. After filtering for the word match (lowercase), the number of posts decreased to 101,262 posts, with *r/CryptoCurrency* having 65,378 posts.

Figure 2.2 illustrates a number of posts over time in both subreddits. The number of posts increased in 2021 compared to 2020 by almost eight times, with posts in *r/wallstreetbets* surging at the end of January 2021 and the beginning of February with many discussions revolving around the price of *DogeCoin (DOGE)*, which has increased by ten times from December 31st, 2021 to January 29th, 2022.

Pre-processing Reddit data. Reddit posts consist of two parts - the title and the post content. Some Reddit posts' content - 40,577 to be exact - are unavailable and marked as *[removed]*, *[deleted]*, or

[6] Documentation of Binance API: `https://binance-docs.github.io/apidocs/spot/en/#introduction`
[7] Documentation of CoinMarketCap API: `https://coinmarketcap.com/api/documentation/v1/`

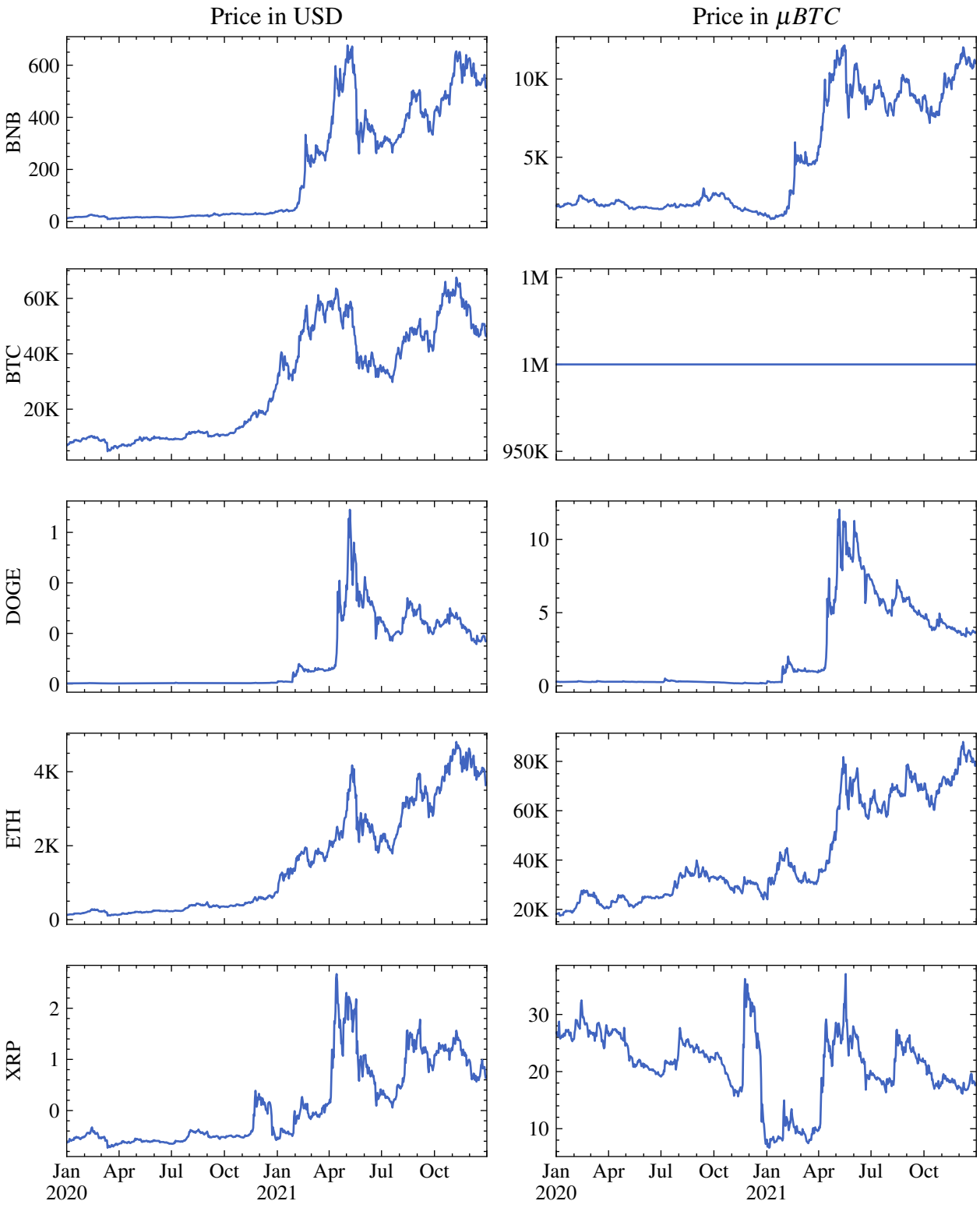


Figure 2.1: Price of most trade volume cryptocurrencies comparison in USD and μBTC

Table 2.1: Example of random Reddit text messages

Title	Content
TRON	Wall Street bets let's get TRON to 25 ce...
Badger DAO	I don't see much about badger DAO on thi...
Poly Network Hacking: Oversimplified	The Poly Network hacking was a big deal....
Cardano Shuffles Crypto Markets As ADA C...	
Solana Ventures, Forte and Griffin Gamin...	

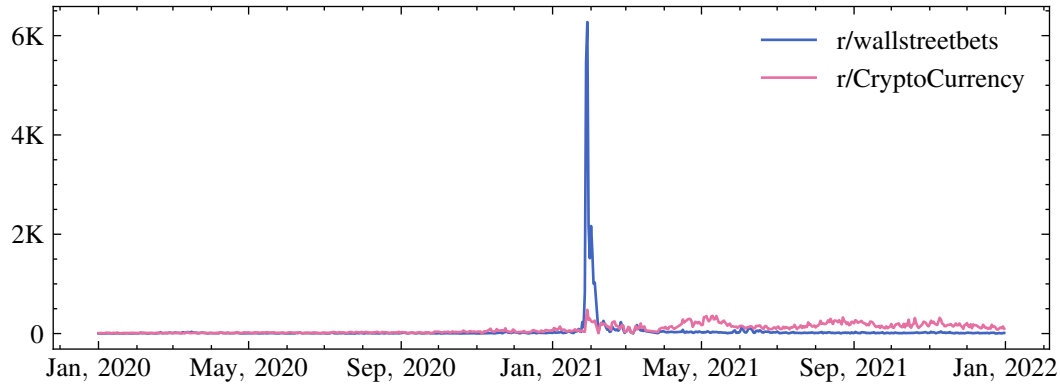


Figure 2.2: Number of Reddit posts in subreddits over time

nan, so they are replaced with empty strings (see table 2.1 for an example). Then, both *Title* and *Content* are lowercased and all the mentions of cryptocurrencies names, symbols, and slugs (for example, *BTC*, *Bitcoin*, or *bitcoin*) are removed.

2.2. Data labeling

2.2.1. Cryptocurrency price change labeling

It is a classification task, and data needs to be labeled. A threshold of 10% for the price change is chosen to ensure a sufficient amount of price changes are labeled while avoiding a very small threshold:

$$Y = \begin{cases} 1, & \text{if next day returns are bigger than } 10\% \\ 0, & \text{if next day returns are smaller than -}10\%. \end{cases} \tag{2.1}$$

By utilizing this labeling approach, a considerable number of data points are labeled: 6,260 as negative and 8,717 as positive. More often, less traded cryptocurrencies are labeled, as can be seen in the distribution of negative and positive labels among different cryptocurrencies Figure 2.3 (refer to the appendix A.2 for different threshold figures). Furthermore, positive and negative labels are distributed over time, as illustrated in Figure 2.4.

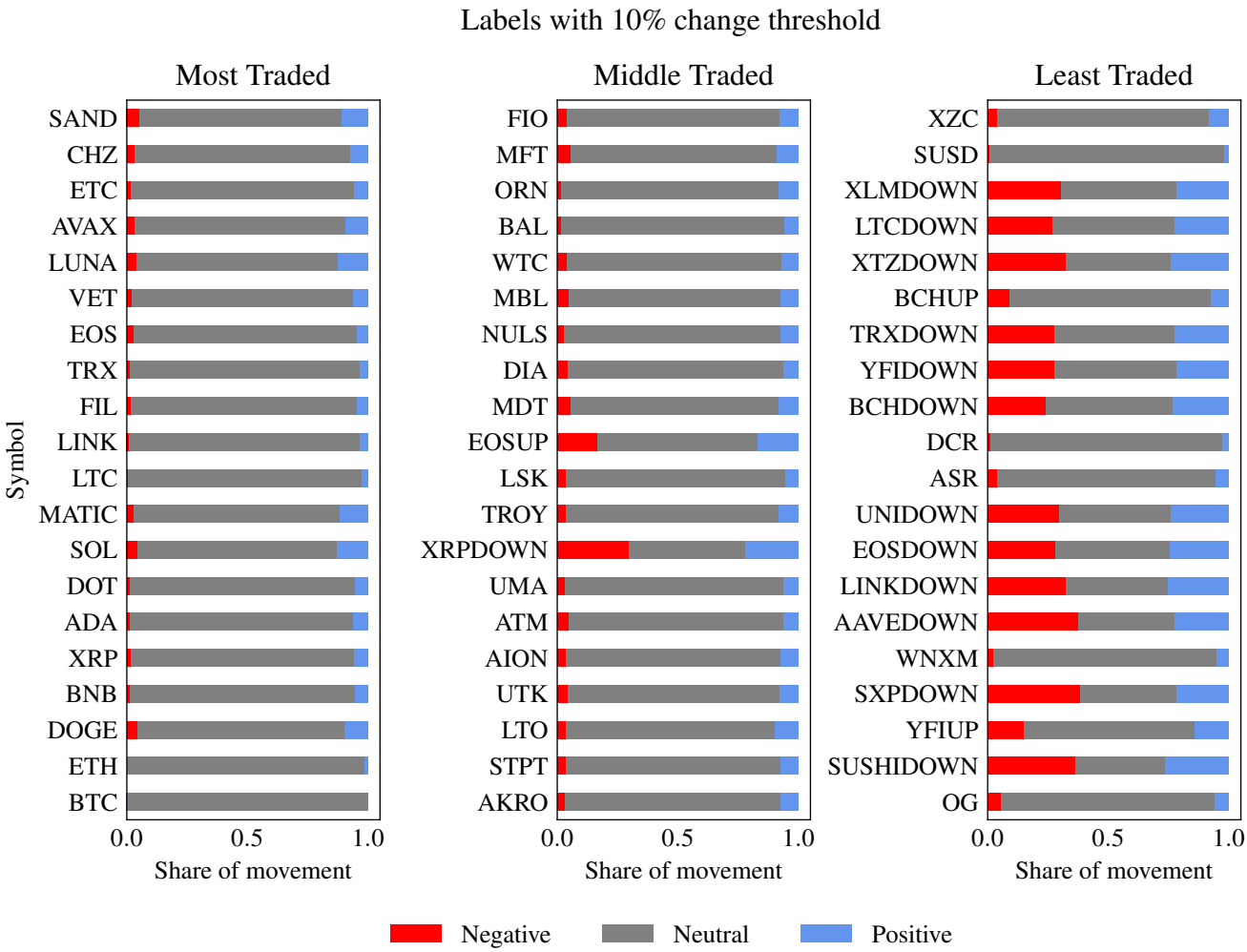


Figure 2.3: Sample of cryptocurrencies label amount with 10% threshold for the price change in the upcoming 7 days

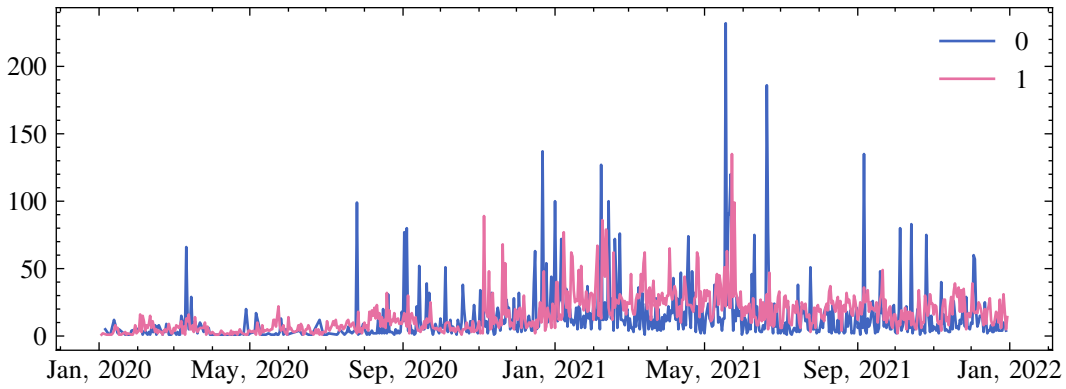


Figure 2.4: Count of cryptocurrency positive and negative price change labels, calculated as shown in the equation 2.1, over time

Table 2.2: Tain-test split amount of labeled data

Data source	Test	Train	Test %	Train %
All Reddit posts	16,478	84,784	16.3%	83.7%
All trading data	29,834	123,555	19.4%	80.6%
Labeled Reddit posts	1,036	15,865	6.1%	93.9%
Labeled trading data	2,728	17,651	13.4%	86.6%

2.2.2. Reddit posts preprocessing and labeling

Reddit posts are labeled using a similar method as cryptocurrency price change labeling (eq. 2.1), but based on the Reddit post publication date:

$$Y = \begin{cases} 1, & \text{if next calendar day cryptocurrency returns are bigger than 10\%} \\ 0, & \text{if next calendar day cryptocurrency returns are smaller than -10\%.} \end{cases} \tag{2.2}$$

After labeling, there are 16,899 unique labeled Reddit posts and 16,901 data points due to some messages mentioning multiple cryptocurrencies with 8,954 positive and 7,947 negative labels (it is fairly equally distributed over time, see Figure A.7 for reference). Like the total amount of Reddit messages, there are many more posts in 2021 compared to 2020 (Figure A.6 shows the monthly count of labeled Reddit data per subreddit). Another step, after having labeled posts, is to pick and train a classification algorithm, but before - leaving some data for the testing purposes of the final model performance.

2.3. Test data holdout

For the testing purposes of the final model, some data is left out: all the Reddit posts and cryptocurrency trading information from 2021-10-01 till 2022-12-31 are left out from model training. This leaves 93.9% of labeled Reddit posts and 80.6% of labeled cryptocurrency price data in the training set (refer to table 2.2 for more detailed data and figure 2.5 to see how the mentioned holdout data looks like in comparison to the Bitcoin price)

2.4. Classification of textual data using BERT, BigBird-RoBERTa, and XLNet

The section covers methods used for cryptocurrency class prediction: BERT, BigBird-RoBERTa, and XLNet.

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer-based machine learning technique for natural language processing. It is designed to understand the context of each word by looking at it as a sequence, i.e., at the words that come before and after the word [44].

BigBird-RoBERTa is a combination of RoBERTa and BigBird. RoBERTa is a variant of BERT, which

trains on larger batches [67]. BigBird can handle longer sequences (text in the sentiment analysis case) [68]. BigBird-RoBERTa combines both qualities, making it better for longer text classification.

XLNet is another transformer-based approach. While BERT is using a cloze task (masking certain words in the input and therefore disrupting the input), XLNet learns of bidirectional contexts by optimizing the expected likelihood over all possible permutations of the input sequence factorization order [7].

Hyperparameter search was done for all three mentioned transformers - BERT, BigBird-RoBERTa, and XLNet - by using population-based learning. Labels of messages are fairly equally distributed. Therefore accuracy was used as an optimization objective. To make calculations faster, 15% of the dataset was sampled for the task. Some of the messages are very short, so it was filtered to include only ones with at least ten characters. The hyperspace for the search was defined as the learning rate in uniform between 1e-6 and 1e-4, train epochs between 2 and 5, and weight decay uniform between 0 and 0.3. BigBird-RoBERTa was identified as the best-performing model from the hyperparameter search, so the research proceeded with this model.

After choosing BigBird-RoBERTa as the best-performing model, a new set of models, fine-tuned with different subsets based on the timeframe, is created. These subsets include ten months of backward Reddit messages and are up to the last day of the previous month for each month from January 2021 till the end of September 2022.

2.5. Classification of textual data using GPT- 3.5

Generative Pre-Trained Transformer-3.5 (GPT- 3.5) is an advanced language model developed by OpenAI. This model is a transformer-based deep learning model, leveraging the principles of unsupervised learning to generate human-like text. In the context of sentiment analysis, GPT-3.5 can be employed to understand and interpret the emotional tone embedded within textual data. This model, owing to its comprehensive training on a diverse range of internet text, exhibits a robust understanding of language semantics, subtleties, and intricacies. Therefore, it is proficient at classifying text into sentiment categories, such as positive, negative, or neutral.

All the labeled Reddit messages, using their titles and content, are classified with Generative Pre-Trained Transformer-3.5 (GPT-3.5) into positive, negative, and neutral sentiments by using the following prompt:

> You are an AI language model trained to analyze and detect the sentiment of reddit messages about cryptocurrencies from the subreddits cryptocurrency and wallstreetbets. Analyze the messages I'll send you and determine if the sentiment is: positive, negative, or neutral. Important: respond with only one word out of this list: [POSITIVE, NEGATIVE, NEUTRAL]

Returned response of the API is assigned as a sentiment class of each message.

2.6. Feature engineering

This section describes the calculations of features, which will be later used for the prediction model creation.

2.6.1. Computation of technical indicators

Classification can be done by only using technical indicators of cryptocurrencies, but technical indicators will be combined with the output of sentiment analysis from the subsection 2.4. A total of 42 technical indicators were calculated using the *ta* Python package [8], which created 86 new features (due to some of the technical indicators having multiple outputs). These features were added to the cryptocurrency sales data. The list of the used technical indicators is provided in the appendix B.

2.6.2. Features from Reddit sentiment analysis

Features were generated from the Reddit data as follows:

- Message counts per subreddit: count of messages written in total and per subreddit.
- Messages counts per sentiment per subreddit: for both, transformer predicted sentiment (including *Short text* messages as another class), and GPT-3.5 predicted sentiment.
- Share of positive and negative sentiment messages: $\sum m_p / \sum m$ and $\sum m_n / \sum m$, where m - all Reddit messages count, m_p - positive sentiment messages count, m_n - negative sentiment messages count.
- Lagged rolling sums of all above measures in the windows of the last 3, 7, 14, and 28 days.

This results in 15 generic features (without a sentiment) and 69 features for each sentiment classification method. After feature generation, a model to predict the price movement of cryptocurrency can be built.

2.7. Data manipulation before price prediction

This section reviews the necessary data preparation for price prediction, like cleansing and selecting relevant cryptocurrencies.

2.7.1. Data split for model training and best model selection

The data was further divided to obtain the best model and corresponding hyper-parameters. The previous train dataset, mentioned in the section 2.3, was further divided into train/test datasets and some data leaving out for the space of technical indicators calculation based on the time. The divisions, illustrated in Figure 2.5, are as follows:

[8] link to the package and documentation: `https://github.com/bukosabino/ta`

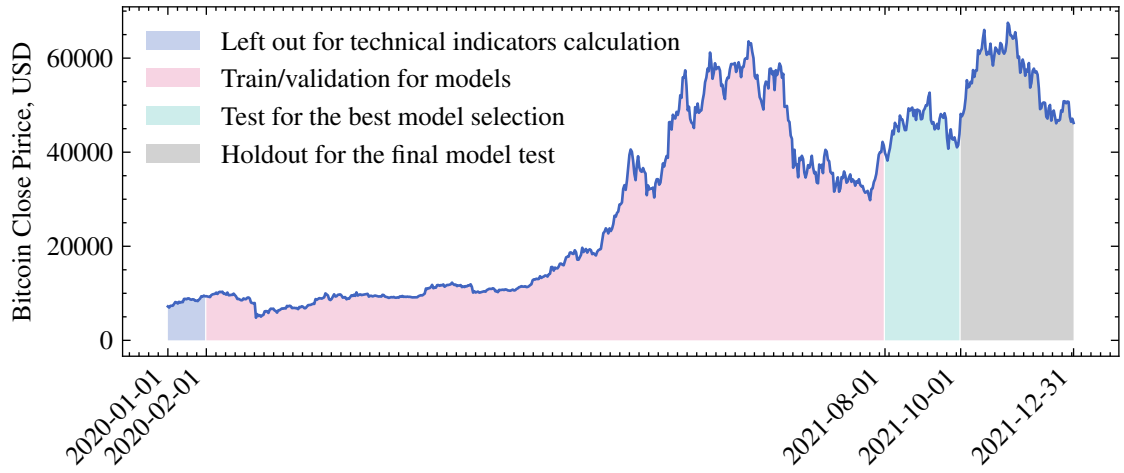


Figure 2.5: Split of the dataset for the training and validation, testing for the best model selection, and the final testing in comparison to the Bitcoin price in USD

- January 2020 was left out to have space for features calculations (technical indicators, moving sum of Reddit messages, etc.)
- February 2020 till July 2021 was used for training, hyper-parameter search, and validation of the models
- August 2021 - September 2021 was used for testing the models and selecting the best model.

2.7.2. Reddit messages cleaning and a subset of data selection

Following insights are made using just the subset of *train/validation for models*, shown in Figure 2.5. There are three cryptocurrencies that had abnormally high mentions in Reddit messages in January and February 2021: DOGE, JST, and XRP. The number of messages mentioning these cryptocurrencies is visualized in Figure 2.6. After removing these three cryptocurrencies from the training dataset, the daily amount of Reddit posts became more consistent, as illustrated in Figure 2.7.

Later, the cryptocurrencies, having at least one Reddit message during the training data period, are selected. It leaves the training and the following data with 212 cryptocurrencies (out of 304 having trading data in the training sample). A full list of cryptocurrency symbols used in the following analysis is presented in Table 2.3.

2.8. Hyperparameters search for Random Forest and Gradient Boost

Several methods were employed for hyper-parameters optimization for predicting price movement: identical variance feature removal, top-K feature selection, and the hyper-parameters of the classification method itself.

Firstly, all the features with identical variance were removed in the pipeline: it eliminates redundant features that do not add value to the classification methods. By removing such features, the model's complexity is also reduced, meaning lower computational resources. Next, the top-K feature selection

Table 2.3: List of selected cryptocurrencies for the model training data

1INCH	BTCDOWN	DNT	HNT	MTL	RSR	TWT
AAVE	BTCST	DOCK	HOT	NBS	RUNE	UMA
ADA	BTG	DODO	ICP	NEAR	RVN	UNFI
ADADOWN	BTS	DOT	ICX	NEO	SAND	UNI
AION	BTT	DOTDOWN	INJ	NKN	SC	USDC
AKRO	BURGER	DREP	IOST	NMR	SFP	UTK
ALGO	BUSD	DUSK	IOTX	NPXS	SHIB	VET
ALICE	C98	EGLD	IRIS	NU	SKL	VITE
ALPHA	CAKE	ENJ	KAVA	NULS	SLP	VTHO
ANKR	CELO	EOS	KEEP	OCEAN	SNX	WAN
ANT	CELR	EPS	KEY	OG	SOL	WAVES
AR	CFX	ETC	KMD	OGN	SRM	WIN
ARDR	CHR	ETH	KSM	OM	STMX	WING
ATA	CHZ	ETHDOWN	LINA	OMG	STORJ	WRX
ATOM	CKB	FET	LINK	ONT	STRAX	WTC
AUDIO	COCOS	FIL	LIT	ORN	STX	XEM
AUTO	COMP	FIO	LPT	OXT	SUPER	XLM
AVA	COS	FIRO	LRC	PAXG	SUSHI	XMR
AVAX	COTI	FIS	LSK	PERL	SXP	XRPDOWN
AXS	CRV	FLM	LTC	PERP	TCT	XTZ
BADGER	CTK	FLOW	LTO	PNT	TFUEL	XVS
BAKE	CTSI	FORTH	LUNA	POLS	THETA	YFI
BAL	CTXC	FTM	MANA	POND	TKO	ZEC
BAND	CVC	FTT	MATIC	QNT	TLM	ZEN
BAT	DASH	FUN	MBL	QTUM	TOMO	ZIL
BCH	DATA	GRT	MDX	RAMP	TORN	ZRX
BEAM	DCR	GTC	MFT	REEF	TRB	
BLZ	DEGO	GTO	MIR	REN	TROY	
BNB	DENT	HARD	MITH	REP	TRU	
BNT	DGB	HBAR	MKR	RLC	TRX	
BOND	DIA	HIVE	MLN	ROSE	TUSD	

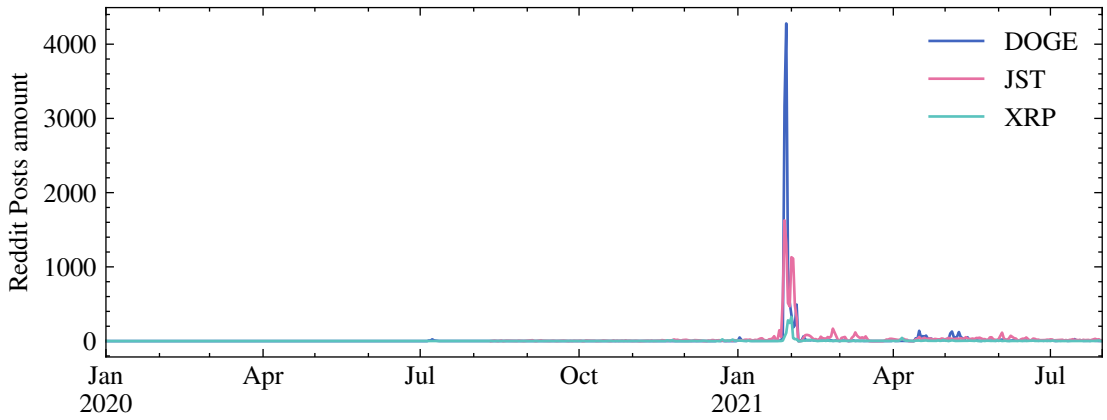


Figure 2.6: Cryptocurrencies excluded from the model training due to abnormally high volume of Reddit messages in a short period

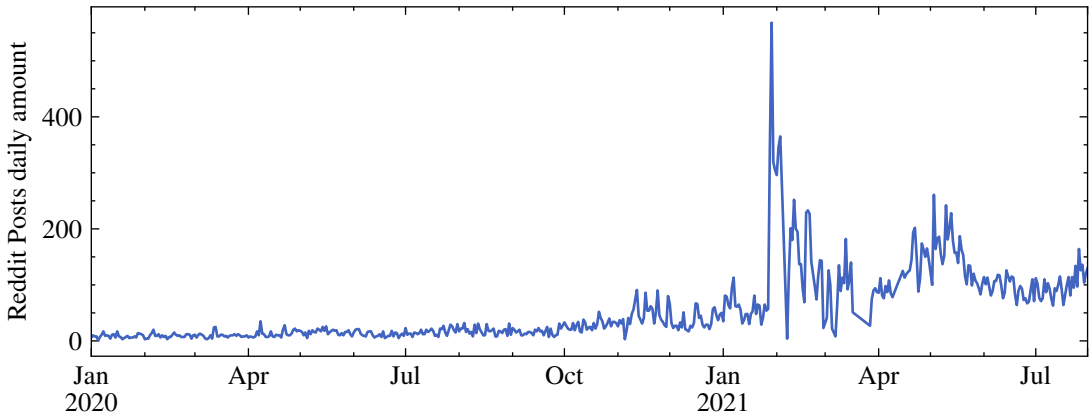


Figure 2.7: Reddit posts count over time after excluding abnormal cryptocurrencies, which is shown in 2.5

was applied. The value of K represents the desired number of features to be selected for the classification method - reducing features helps reduce overfitting. K was included in the hyper-parameters search space. The method chosen for hyper-parameters optimization was Halving Random Search cross-validation (CV) and applied to two classification algorithms - Gradient Boosting and Random Forest.

The hyperparameters for Gradient Boosting were searched in the following ranges:

- **The number of boosting stages to perform:** Uniformly distributed integers between 5 and 200
- **Learning rate:** Uniformly distributed real values between 0.001 and 0.2
- **Max features:** Choice of 1.0 or *sqrt*
- **Max depth:** Choice of 3, 10, 100, 500, or None
- **Min samples split:** Choice of 2, 10, or 15
- **Min samples leaf:** Choice of 1, 4, or 9.
- **K:** Uniformly distributed integers between 5 and all the parameters.

Likewise, for the Random Forest, the hyperparameters were searched in the following ranges:

- **The number of trees in the forest:** Uniformly distributed integers between 5 and 1000
- **Max features:** Choice of 1.0 or *sqrt*
- **Max depth:** Choice of 3, 10, 100, 500, or None
- **Min samples split:** Choice of 2, 10, or 15
- **Min samples leaf:** Choice of 1, 4, or 9.
- **Bootstrap samples:** Choice of True or False
- **K:** Uniformly distributed integers between 5 and the maximum value

2.9. Features combination

Additionally, various feature combinations were examined to assess the impact of Reddit messages and their classification methodology on model performance. Four different subsets of features were considered, with each being assigned a distinct model name:

- **TA**: this feature subset includes features derived from technical indicators solely, as described in section 2.6.1
- **TA_REDD**: includes features from *TA* features subset and adds generic features from Reddit, without including any information about sentiment, as described in section 2.6.2
- **TA_TRA**: has everything from *TA* and *TA_REDD* feature subset, and in addition, features from sentiments, predicted by BigBird-RoBERTa model, is introduced. Again described in the same section 2.6.2
- **TA_GPT**: has everything from *TA* and *TA_REDD* feature subset, and in addition, features from sentiments, predicted by GPT-3.5, is introduced. Again described in the same section 2.6.2

These combinations were evaluated using hyperparameters search for both Random Forest and Gradient Boosting classification methods, as described in section 2.8. The results of this hyper-parameter search were used to select the best-performing combination of hyper-parameters and feature combinations.

2.10. Bidirectional Long Short-Term Memory (BI-LSTM)

To improve the performance of the best-performing classification model with its features, a Bidirectional Long Short-Term Memory (BI-LSTM) was incorporated. BI-LSTM is a recurrent neural network (RNN) that effectively captures temporal dependencies in sequential data. To optimize BI-LSTM performance, a Hyperband algorithm was conducted for hyperparameters search. The Hyperband hyperparameters search combines random search with early stopping. Because of it, it efficiently explores the hyperparameter space by iteratively training a set of models with different hyperparameter configurations and discarding poorly performing ones.

The BI-LSTM architecture consists of the following components:

- **Bidirectional LSTM layer**, the input units were defined as a hyperparameter. The LSTM layer also allowed for regularization, the type of which was another hyperparameter
- **Batch Normalization layer**, applied to standardize the outputs from the preceding LSTM layer. This aids in augmenting network stability and performance efficiency
- **Second bidirectional LSTM layer**, with the number of neurons defined as a hyperparameter. This layer only returns the final sequence output

- **Dropout layer**, included to impede overfitting through random nullification of a proportion of input units during the training process. The dropout rate constitutes a hyperparameter
- **Dense output layer** with one neuron. The activation function employed in this layer is chosen from two hyperparameter options.

The specific hyperparameters and their respective ranges or choices were established as follows:

- **Input Units:** The number of neurons in the first LSTM layer. The range for this hyperparameter was set from 288 to 1024 with a step of 92.
- **Regularization Type:** The regularization is applied on the LSTM layers. The options were None, L1, and L2 regularization. For L1 and L2 regularization, the specific regularization value was a hyperparameter, sampled logarithmically from 1e-5 to 1e-2.
- **Layer Neurons:** The number of neurons in the second LSTM layer. This was set to be searched from 96 to 1024 with a step of 32.
- **Dropout Rate:** The proportion of neurons to be turned off during training to prevent overfitting. This was sampled from 0 to 0.8 with a step of 0.1.
- **Activation Function:** The activation function will be used in the final dense layer. The choices were *relu* and *sigmoid*.
- **Learning Rate:** The step size that the optimizer takes to adjust the weights. This was sampled logarithmically from a range of 1e-8 to 1e-3.
- **Batch Size:** The number of training examples utilized in one iteration. The choices for this hyperparameter were 16, 32, 64, and 128.

The model was built using binary cross-entropy as the loss function, with the Adam optimization algorithm. Also, the **Early Stopping** technique was utilized during the training process. During the training, it monitored the model's *accuracy* - if the accuracy did not improve over three consecutive epochs, the training was prematurely terminated. Although not a tunable hyper-parameter, Early Stopping served as an effective control during training, reducing overfitting risks and computational costs.

2.11. Data Scaling for BI-LSTM

A custom scaler was developed and implemented when preparing the data for the LSTM model. This scaler was designed to perform Min-Max feature normalization on a per-cryptocurrency basis, which ensured that each cryptocurrency's time series data was scaled independently. Min-Max scaling transformed each cryptocurrency data to be between 0 and 1, as required for the LSTM algorithm. This scaler is defined as:

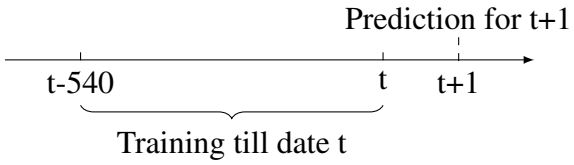


Figure 2.8: Illustration of training and prediction on the holdout data for selected model testing for day t.

$$X_{\text{scaled},c} = \frac{X_c - X_{\min,c}}{X_{\max,c} - X_{\min,c}} \tag{2.3}$$

where X_c is the original feature value for a specific cryptocurrency c, $X_{\min,c}$ refers to the minimum value of the feature for cryptocurrency c, and $X_{\max,c}$ stands for the maximum feature value for cryptocurrency c. Finally, $X_{\text{scaled},c}$ denotes the feature value after scaling for the specific cryptocurrency c.

To ensure appropriate data normalization, the Mix-Max Scaler was fitted just on the training data for each cryptocurrency: it prevents information leak from the test data, as test data should be completely unseen by any part of the model process. Then, the fitted scaler was applied to the test data, transforming test features based on the min-max range learned from the training dataset: because of it, some values on the test data can be out of the desired range of [0,1]. A cap and a bottom were set for the transformed test data values to avoid being out of range. Specifically, any scaled value exceeding one was capped at 1, and any value below 0 was set to 0, enduring transformed values fitting in the predefined [0,1] range.

2.12. Testing and training strategy on the held-out data

After a model to predict cryptocurrency price movement is selected, another test data is performed. This test data was put aside from the beginning, as described in section 2.3. To better illustrate real-life performance, the model is retrained with 540 days of backward data (the same amount as it was done for training) each day to predict the price movement of the next day, as visualized in Figure 2.8.

A simple trading strategy is applied to the same holdout data. This trading strategy identifies cryptocurrencies with the probability of increasing to be than the pre-defined threshold, buys them in equal amounts expressed in μBTC (with the remaining free amount of μBTC), and sells if it is not lower than another pre-defined threshold for holding the asset. The steps of the strategy are as follows:

1. Determine the list of coins, referred as c, for purchase based on the probability of price increase (P_c) for cryptocurrency c exceeding a pre-determined buy threshold (B_T):

$$Buy = \{\text{c} \,|\, P_c \geq B_T\}$$

2. Select the list of coins to hold, chosen from the ones already in the portfolio ($Portfolio$), with

a probability of price increase (P_c) above the holding threshold (H_T):

$$Hold = \{Portfolio \,|\, P_c \geq H_T\}$$

3. Determine the total cash ($Cash$) from selling coins not in the $Hold$. The closing price of a coin ($Price_c$) is the final price at which the coin traded during the specific trading period:

$$Cash = \sum_{\mathrm{c} \in Portfolio \setminus Hold} (Price_{\mathrm{c}} \cdot U_c),$$

where U_c represents the number of coins for cryptocurrency c existing in portfolio to sell

4. Calculate the number of units ($Units$) for each coin in Buy. This is done by dividing the $Cash$ evenly among all coins in the Buy: $N_{Buy} = |Buy|$. It means each new cryptocurrency is bought in equal amounts in μBTC and is expressed as:

$$U_c = \frac{Cash}{N_{Buy} \cdot Price_{\mathrm{c}}},$$

where

- c: A specific cryptocurrency coin
- P_c: Probability of price increase for coin c
- B_T: Buy Threshold, the pre-determined probability threshold for buying a coin
- H_T: Hold Threshold, the pre-determined probability threshold for holding a coin
- Buy: Set of coins to be bought
- $Hold$: Set of coins to be held from the portfolio
- $Portfolio$: The current portfolio of coins.
- $Price_c$: The closing price of coin c at the end of the trading period.
- U_c: Number of units of coins c to buy or sell (as well as in portfolio)
- $Cash$: Total cash available for buying new coins
- N_{Buy}: Number of coins in Buy

3. Research results

3.1. Reddit sentiment analysis

The section describes sentiment analysis by fine-tuning transformers - BERT, BigBird-RoBERTa, and XLNet. It overviews hyperparameter search, selecting the best-performing model, and finally fine-tuning the best-performing model while avoiding data leakage.

3.1.1. Hyperparameter search and method selection

Results (table 3.1) showed BigBird-RoBERTa outperforming others in most metrics, yet all the models were very close. After BigBird-RoBERTa was chosen for the text classification task, the next step is to train it on more data while avoiding data leakage.

3.1.2. Fine tuning BigBird-RoBERTa

Results on all the models are shown in table 3.2, and a summary of it in table 3.3. Accuracy (average 0.53) and precision (average 0.58) are more consistent over different timeframes with standard deviations of 0.03 and 0.07, respectively. At the same time, recall (average 0.51) differs more with a standard deviation of 0.22, also affecting variance in the F1 score (average 0.51).

Each fine-tuned model is used to predict the sentiment of the upcoming month; for example, fine-tuned model on the dates between 2020-08-01 and 2021-05-31) is used to predict the sentiment of the messages written between 2021-06-1 and 2021-07-01: it helps to avoid data leakage in the later research.

3.1.3. Overview of sentiment predictions and the labeling problem

The sentiment of Reddit messages was predicted from 2021-01-01 till 2021-10-01, having 15,399 data points: all the possible dates with the models trained and without going into the test dataset. Results are more or less equally distributed, with 4534 positive predicted messages, 3871 - negative, and 6994 having too short text to predict (less than ten characters).

Table 3.1: Hyperparameters search results with the performance of differnt models

Model	Accuracy	F1	Precision	Recall
BERT	0.5488	0.6713	0.5502	0.8607
XLNet	0.5507	0.6750	**0.5508**	0.8714
BigBird-RoBERTa	**0.5545**	**0.6914**	0.5495	**0.9321**

Table 3.2: Performance of fine-tuned BigBird-RoBERTa models on different timeframes on validation dataset

Training dataset date range	Accuracy	Recall	Precision	F1	Train dataset size	Validation dataset size
(2020-03-01, 2020-12-31)	0.4533	0.6571	0.4423	0.5287	299	75
(2020-04-01, 2021-01-31)	0.5751	0.2773	0.6933	0.3962	2983	746
(2020-05-01, 2021-02-28)	0.5266	0.2280	0.6316	0.3350	4425	1107
(2020-06-01, 2021-03-31)	0.5326	0.4235	0.5570	0.4812	4596	1149
(2020-07-01, 2021-04-30)	0.5431	0.5347	0.5763	0.5547	5198	1300
(2020-08-01, 2021-05-31)	0.5418	0.6602	0.5493	0.5996	5979	1495
(2020-09-01, 2021-06-30)	0.5227	0.4414	0.5692	0.4972	6067	1517
(2020-10-01, 2021-07-31)	0.5333	0.3684	0.6224	0.4628	6066	1517
(2020-11-01, 2021-08-31)	0.5531	0.5297	0.6187	0.5707	6247	1562
(2020-12-01, 2021-09-30)	0.5356	1.0000	0.5356	0.6975	6466	1617

Table 3.3: Averages of fine-tuned BigBird-RoBERTa models on different timeframes performance mentrics (reference for the detailed table 3.2). Weighted average is calculated based on validation dataset size

Measure	Average	Weighted average	Standard deviation
Accuracy	0.5317	0.5384	0.0313
Recall	0.5120	0.5254	0.2239
Precision	0.5796	0.5875	0.0678
F1	0.5124	0.5256	0.1033

Yet, the price change might not necessarily be a good indication of text being positive or negative, but it is expected with the large sample to be able to classify it. Therefore, the classification accuracy of positive or negative sentiment might be different than what is calculated in model performance metrics. Table 3.4 shows randomly selected posts from the dataset with either false positive or false negative classification results. By reading some of the texts, for example, *DOGECOINUP 500% ! ! ! !*, it could mean a positive attitude despite it being labeled negative. So, it is expected that with the accuracy being just a bit over 0.5, the model has learned which texts reflect positive or negative attitudes and true accuracy is higher. If not, text data can be classified without labeling and fine-tuning using a pre-trained language model.

3.2. Sentiment classification using OpenAI's Generative Pre-Trained Transformer-3.5 (GPT-3.5)

Since classification with transformers and labels extracted based on future price changes is suffering from the wrong labels issue (as discussed in the section 3.1.3), another classification algorithm, which is already trained and does not require fine-tuning, is used.

GPT-3.5, using the mentioned prompt in section 2.5, classifies most messages as either positive or neutral (see table 3.5 for details), marking only 4.8% as negative. There is not a lot of correlation between both values: excluding neutral and short text messages, out of positive sentiment predicted by GPT-3.5, 55.6% is also marked as positive by the transformer. In contrast, the negative sentiment

Table 3.4: Example of wrong predictions in Reddit text messages classification

Mistake	Title and text
False positive	Dogecoin: So I tried buying crypto today on Robinhood, but the order was pending for hours!!! It didn't go through and didn't even let me cancel. only went through when it went down, and now I'm just ...
False positive	Jasmy-MGT on Twitter binance listing
False positive	XRP in forbes today
False positive	DOGECOIN UP 500% !!!!
False positive	Decentraland's "Vegas City," the first OASIS? (Ready player one): This looks amazing and right out of Ready Player One. ABOUT VEGAS CITY Vegas City, as innovative as it is immersive, is one of the l...
False negative	XRP missing from Coinbase homepage
False negative	UNISwap hit a all time high at 20 USD !!!!!! WOOOOOO
False negative	Just an FYI. Any fud needs to have real proof not just finger pointing.
False negative	Just a dumb housewife, they said, why would they take my stock advice?
False negative	Data Sharing on Robinhood: DISABLE NOW!

Table 3.5: GPT-3.5 sentiments crossed with sentiments from fine-tuned BigBird-RoBERTa model

		Fine-tuned BigBird-RoBERTa			All
		Negative	Short Text	Positive	
GPT-3.5	Negative	257	51	426	734
	Neutral	1543	5971	1511	9025
	Positive	2071	972	2597	5640
	All	3871	6994	4534	15399

Table 3.6: Randomm Forest best hyperparameters after hyperparameter search

Hyperparameter	TA_TRA	TA	TA_REDD	TA_GPT
Bootstrap samples	False	True	True	False
Max depth	500	500	500	None
Max features	sqrt	sqrt	1.000000	sqrt
Min samples leaf	4	1	4	4
Min samples split	15	2	2	15
The number of boosting stages to perform	725	475	106	892

Table 3.7: Gradient Boosting best hyperparameters after hyperparameter search

Hyperparameter	TA_TRA	TA	TA_REDD	TA_GPT
Learning rate	0.099255	0.185515	0.123909	0.099255
Max depth	None	500	100	None
Max features	1.000000	sqrt	sqrt	1.000000
Min samples leaf	4	4	4	4
Min samples split	10	15	10	10
The number of boosting stages to perform	196	144	125	196

matches 62% of messages. Interestingly, 85% of short text messages were classified as neutral by the GPT-3.5. Which method is more valuable in predicting cryptocurrency price changes will be seen later.

3.3. Price movement classification models

The current train dataset was split into another train-test data to select the best model, leaving 20% for the testing. Then, the models were trained by searching for hyperparameters in the train data with cross-validation (CV) and evaluated on the test data.

3.3.1. Model features selection and Hyperparameters search

The optimal hyperparameters for Random Forest and Gradient Boosting models were searched using the method described in section 2.8. The results for each model are listed in table 3.6 and 3.7, respectively.

The key takeaways from the performance results of optimized models after hyperparameter search can be found in table 3.8. Here are the main points regarding the performance of the models:

- The **TA_TRA** feature set tends to yield the highest Cross-Validation Accuracy (in the validation data), both for the Gradient Boosting and the Random Forest models, indicating its ability to generalize well in unseen data
- Despite being lower in Cross-Validation Accuracy, the **TA_GPT** feature set results in the highest Accuracy in the test set for both the Gradient Boosting and the Random Forest models,

Table 3.8: Comparing model predictions on Validation dataset (CV Accuracy) and other metrics on Test dataset given different set of features. RF refers to Random Forest algorithm, GB - Gradient Boosting.

Features Category	Model	CV Accuracy	Accuracy	Precision	Recall	F1
TA_TRA	GB	**0.6477**	0.6107	0.7820	0.6351	0.7010
	RF	0.6399	0.6184	0.7877	0.6419	0.7074
TA	GB	0.6305	0.6049	0.7780	0.6297	0.6960
	RF	0.6422	0.6068	0.7893	0.6176	0.6929
TA_REDD	GB	0.6296	0.6068	0.7796	0.6311	0.6975
	RF	0.6391	0.6019	0.7787	0.6230	0.6922
TA_GPT	GB	0.6455	0.6136	0.7831	0.6392	0.7039
	RF	0.6383	**0.6252**	**0.7921**	**0.6486**	**0.7132**

suggesting its good performance on specific unseen data

- Regarding Precision, the **TA_GPT** and **TA_TRA** feature sets stand out, particularly when using the Random Forest model, so these feature sets are the most reliable in terms of positively predicting the target class
- In terms of Recall, the **TA_GPT** feature set outperforms the others when used with the Gradient Boosting model
- Finally, when considering the F1 score, the **TA_GPT** and **TA_TRA** feature sets show the highest values, particularly for the Random Forest model, suggesting a balanced trade-off between Precision and Recall.

It is worth noting that TA and TA_REDD performed very similarly, meaning simply adding the volume of Reddit messages (and features generated from it), but without sentiments, brings close to no value compared to the models using solely technical indicators. On the other hand, if sentiments are among feature sets - regardless if they are calculated without fine-tuning the classifier (TA_GPT) or fine-tuning it with labeling depending on the future change of the cryptocurrency price (TA_TRA) - it slightly improves the model's performance. Despite TA_TRA and TA_GPT being very close in performance metrics, TA_GPT is chosen for future analysis due to slightly better scores in the test data and its simplicity of gathering the sentiments of Reddit messages.

3.4. Bidirectional Long Short-Term Memory (Bi-LSTM)

The Bidirectional Long Short-Term Memory (Bi-LSTM) model was introduced to improve performance over Random Forest and Gradient Boosting. It utilized the same feature set as the highest-performing model from the previous section, TA_GPT, which included technical indicators and Reddit messages with their estimated sentiments using the GPT prompt. The Bi-LSTM's hyperparameters were optimized using the training data and evaluated using the test data, just like the Random Forest and Gradient Boosting methods.

3.4.1. Results of the hyperparameter search

The best-performing model, measured with an accuracy score, was identified after completing a hyperparameters search using the Hyperband optimization algorithm for the space and model structure described in section 2.10. The optimal hyperparameters for the model were:

- **Input Units:** 564 neurons in the first LSTM layer
- **Regularization Type:** No regularization was applied
- **Layer Neurons:** 576 neurons in the second LSTM layer
- **Dropout Rate:** The model employed no Dropout
- **Activation Function:** The *sigmoid* function was employed in the final dense layer
- **Learning Rate:** The optimizer adjusted the weights using a step size of approximately 0.000633
- **Batch Size:** 64 training examples were utilized in each iteration.

It is worth noting that even though the model has options for L1 and L2 regularization, the optimal model did not utilize any form of regularization. It indicates that the introduction of a regularization term did not enhance the model's performance for analyzed train data. Moreover, no Dropout was employed, which might suggest that the model did not experience overfitting during the training process. This finding is corroborated by the choice of the sigmoid function in the output layer, often favored for its ability to handle two-class problems well. The best model achieved an accuracy score of 0.672 on the validation set.

3.4.2. Performance of Bi-LSTM and comparison with other methods on test data

Upon comparing Bi-LSTM with Random Forest, which showed the best performance previously, it can be observed that both models exhibit similar accuracy, yet they differ in the precision, recall, and F1 score. The accuracy of the Bi-LSTM model (0.635) is slightly higher than that of the Random Forest (0.625), while the precision of the Bi-LSTM model (0.762) is slightly lower than that of the Random Forest (0.792). In terms of recall, the Bi-LSTM model outperforms the Random Forest, with a recall of 0.715 compared to 0.649. Measuring F1, which balances recall and precision, the Bi-LTSM returns a slightly higher F1 score (0.738) than the Random Forest (0.713). All results are shown in the table 3.9.

In summary, although both models showcase comparable performance, the Bi-LSTM model appears to perform slightly better in terms of accuracy, recall, and the F1 score. Therefore, due to better F1 score and better accuracy, Bi-LSTM is chosen as the model for testing.

Table 3.9: Bi-LSTM and Random Forest performance metrics comparison on the test data for TA_GPT model

	Accuracy	Precision	Recall	F1
Bi-LSTM	**0.6350**	0.7622	**0.7149**	**0.7378**
Random Forest	0.6252	**0.7921**	0.6486	0.7132

Table 3.10: Performance of Bi-LSTM on the holdout data

Accuracy	Precision	Recall	F1
0.6332	0.7741	0.6706	0.7186

3.5. Bi-LSTM performance on holdout data

After Bi-LTSM is chosen as the best model, its performance is measured on the data, which was put aside from the beginning, as described in section 2.12. The performance of Bi-LTSM on the holdout data is very close to what was achieved in the test data described in section 3.4.2. The model on the holdout dataset achieved a 0.7186 F1 score and 0.6332 Accuracy, while on the test data Accuracy was at the level of 0.6350, and F1 score at 0.7378. All main metrics can be seen in table 3.10.

On the other hand, performance on a daily basis varies. Daily accuracy and F1 scores are visualized in Figure 3.1, while recall and precision are in Figure 3.2. The standard deviation of all the metrics stands between 0.19 and 0.23, with maximum values reaching one for all the metrics and minimum being close to zero for Precision, Recall, and F1 (Accuracy minimum is 0.21). All the summary statistics of daily performance are shown in table 3.11.

3.6. Trading strategy and backtesting trading results

A simple trading strategy, described in section 2.12, for the Bi-LSTM model is simulated. A chosen trading strategy requires the selection of probability (P_c) thresholds to buy (B_T) the cryptocurrency or keep (HT) it in the portfolio after buying. To determine the thresholds of buying or keeping it, the

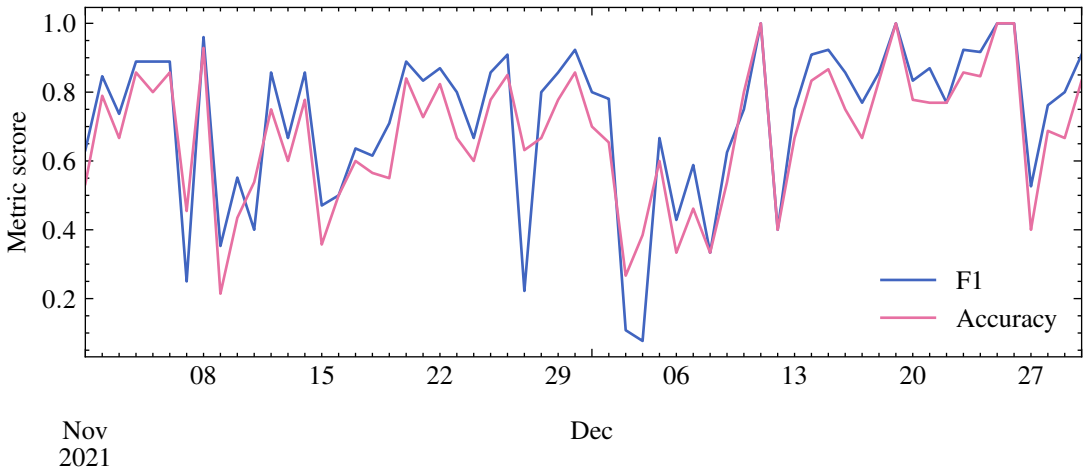


Figure 3.1: Daily Accuracy and F1 score of Bi-LSTM on the holdout dataset

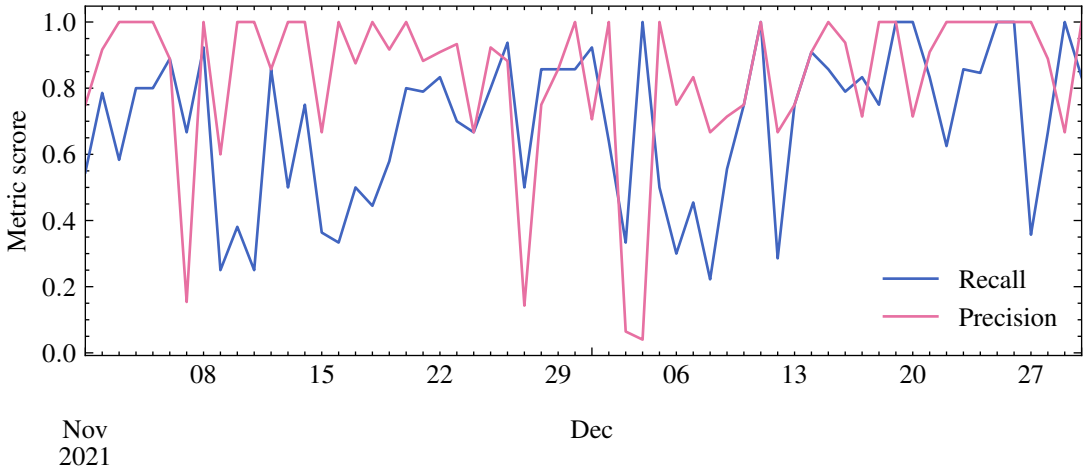


Figure 3.2: Daily Precision and Recall of Bi-LSTM on the holdout dataset

Table 3.11: Daily performance of Bi-LSTM on the holdout data, summary statistics

	Accuracy	Precision	Recall	F1
mean	0.6781	0.8375	0.6945	0.7211
std	0.1931	0.2334	0.2298	0.2277
min	0.2143	0.0400	0.2222	0.0769
25%	0.5471	0.7500	0.5000	0.6226
50%	0.6937	0.9129	0.7679	0.8000
75%	0.8333	1.0000	0.8571	0.8889
max	1.0000	1.0000	1.0000	1.0000

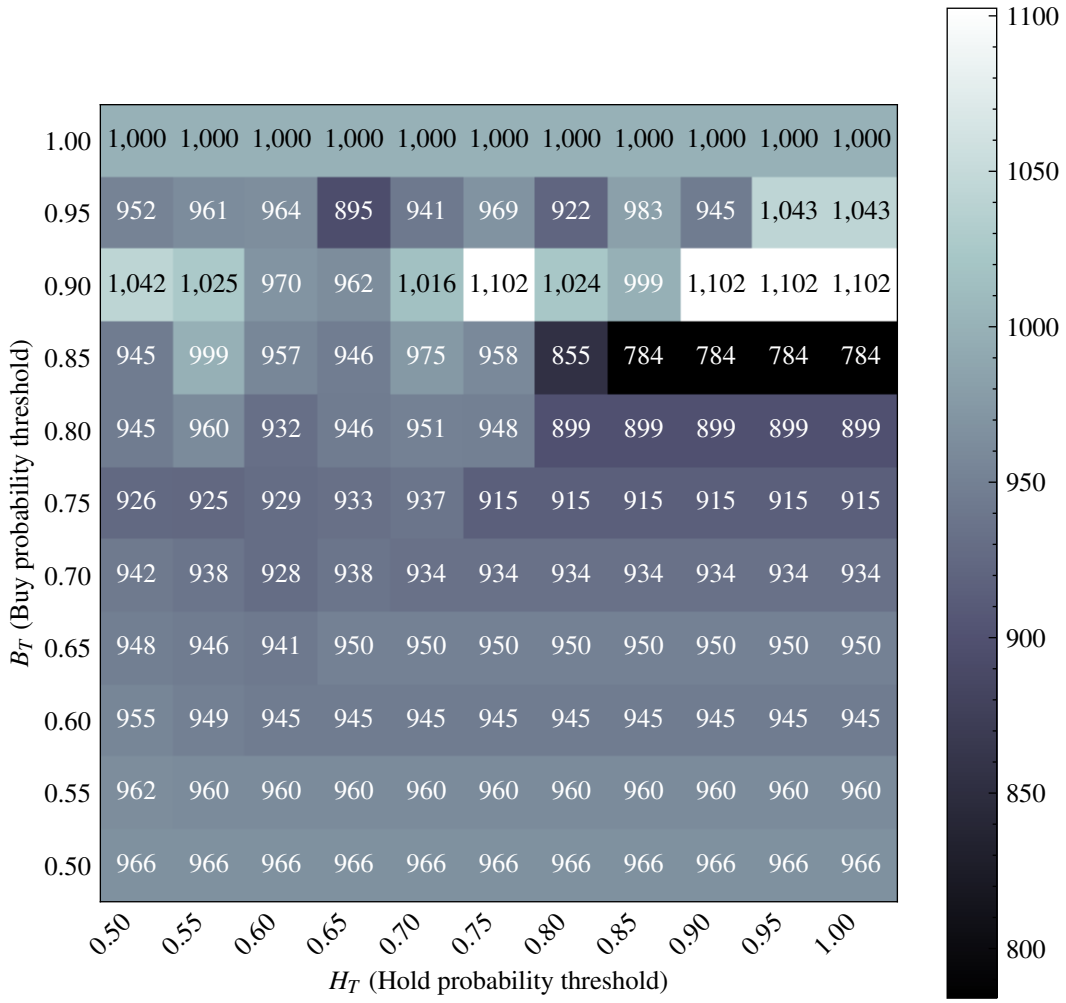


Figure 3.3: Money, expressed in μBTC, after investing for 15 days depending on the probability thresholds B_T for buying and B_T for holding the cryptocurrency with the initial investment of 1.000 μBTC. The Period for investment is from November 1st, 2021, until November 15th, 2021.

test was run on the first 15 days of November 2021 for all the probabilities between 0.5 and 1.0 with the step of 0.05. Results, which can be seen in figure 3.3, showed the optimal choice would be as follows:

- $B_T = 0.90$
- $H_T \in \{0.75, 0.90, 0.95, 1.00\}$.

Trading fees must be kept in mind; the longer the holding period of cryptocurrency, the fewer fees will be paid. With this in mind, H_T is chosen at the probability of 0.75.

With chosen thresholds, trading is simulated for the remaining holdout data, i.e., from November 15th, 2021, to December 31st, 2021, with an initial budget of 1.000 μBTC. Results of such backtesting ended with 973 out of 1,000 μBTC, meaning -2.7% return of investment over 45 days. The daily value of the investment is displayed in Figure 3.4.

Summary statistics of daily *Cash*, shown in the table 3.12, shows the volatility in the investment amount, ranging from the minimum balance of 961 μBTC to the maximum of 1,181 μBTC with the

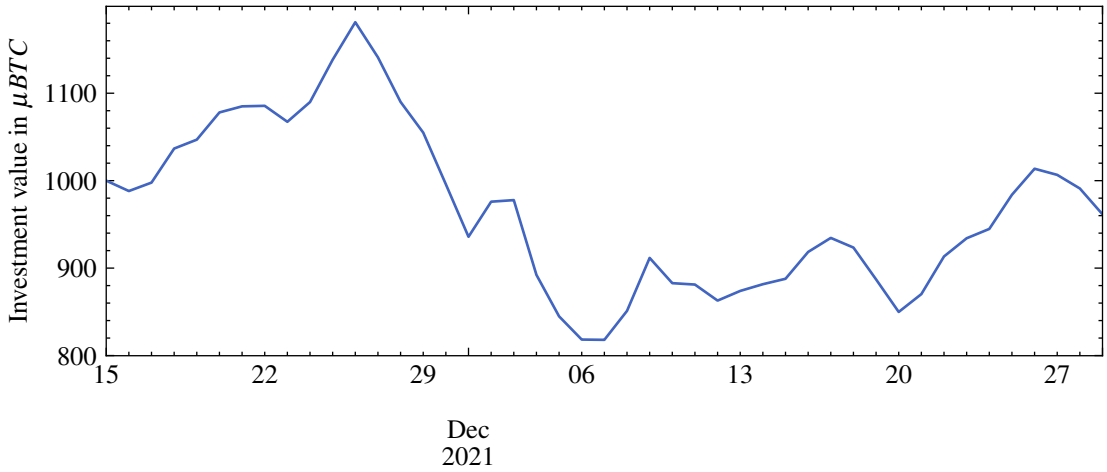


Figure 3.4: Trading strategy daily investment value expressed in μBTC. Trading ended with 973 μBTC with a starting budget of 1,000 μBTC. November 15th, 2021 - December 31st, 2021

Table 3.12: Summary statistics of trading strategy investment expressed in μBTC. Trading ended with 973 μBTC with the starting budget of 1,000 μBTC. November 15th, 2021 - December 31st, 2021

	Cash
mean	966.78
std	93.22
min	817.96
25%	887.06
50%	961.49
75%	1036.72
max	1181.20

standard deviation of 93.

The summarized results of the investment strategy are as follows:

- Total Return: 0.96
- Annualized Return: -14.06%
- Annualized Volatility: 0.674
- Sharpe Ratio: -21.01

It is worth noting that this trading strategy and model performance is evaluated for 45 days only. In these 45 days, a negative Sharpe ratio might indicate that the risk-free returns might be greater than using an analyzed trading strategy.

Conclutions

1. Sentiments extracted from Reddit messages improve the performance of price movement prediction models. Specifically, they improve the model's accuracy, precision, and recall, which are crucial metrics in evaluating the performance of predictive models.

2. Features derived directly from Reddit messages, excluding the sentiment estimation, offer minimal value in price movement prediction. This suggests that the emotional tone or sentiment embedded in the messages plays a significant role in prediction accuracy

3. OpenAI's Generative Pretrained Transformer 3.5, often referred to as ChatGPT, demonstrates performance that is comparable to, or even surpasses, a custom-built sentiment prediction model utilizing state-of-the-art Transformer models, specifically Bidirectional Encoder Representations from Transformers (BERT), XLNet, and BigBird-RoBERTa

4. The Bidirectional Long Short-Term Memory (Bi-LSTM) machine learning model outperforms traditional machine learning models, namely, the Random Forest and Gradient Boosting algorithms. This showcases the potential of Bi-LSTM in handling complex prediction tasks.

5. Despite the superior performance of the Bi-LSTM model in price movement prediction, the implementation of a straightforward trading strategy using this model does not result in profitable outcomes when backtested on holdout data, even before taking trading fees into account

6. The study, however, has limitations:

   (a) Amount of data used. The research covers a vast group of cryptocurrencies traded at Binance, one of the leading cryptocurrency exchanges. Yet, there are more cryptocurrencies. Additionally, the time period used in the study - years 2020 and 2021 - may return different results compared to other timeframes. The Reddit data messages were used from two English subreddits - *WallStreetBets* and *Cryptocurrency*. It provides just a partial view of the discussions on Reddit about cryptocurrencies, as there are numerous more subreddits on this social media platform.

   (b) Limitations related to the choice of machine learning algorithms. This study employed a select number of algorithms, namely Bidirectional Long Short-Term Memory, Random Forest, and Gradient Boosting. While these were chosen for their robustness and perceived effectiveness, it is possible that other machine-learning models may yield better results for price movement prediction.

   (c) Limitations regarding chosen trading strategy. Robust and simple trading strategy was employed to backtest the returns of using model's prediction for trading cryptocurrencies, but there are a lot of different trading strategies

7. Future research topics based on the limitatations:

(a) Expanding the dataset: The research could be expanded by incorporating data from other cryptocurrency exchanges, additional cryptocurrencies, and a longer time frame. This would provide a broader base for analysis and possibly improve the accuracy of predictions.

(b) Exploring other machine learning models: Further research could be directed towards exploring and comparing the effectiveness of other machine learning algorithms in predicting cryptocurrency price movements. More advanced or specialized models could potentially improve prediction accuracy.

(c) Customizing models for individual cryptocurrencies or cryptocurrency groups: Given the unique features and community sentiments associated with individual cryptocurrencies, future studies could focus on building tailored prediction models for each cryptocurrency. This could potentially yield more accurate predictions for individual cryptocurrencies.

(d) Developing advanced trading strategies: Future work could focus on developing and testing more complex or varied trading strategies using the predictive model. This might improve the profitability of the trading strategy when tested.

References

[1] CoinMarketCap. Global Cryptocurrency Market Charts | CoinMarketCap. URL https://coinmarketcap.com/charts/.

[2] Michel Ballings, Dirk Van den Poel, Nathalie Hespeels, and Ruben Gryp. Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056, 2015. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2015.05.013. URL https://www.sciencedirect.com/science/article/pii/S0957417415003334.

[3] Qinkai Chen. Stock movement prediction with financial news using contextualized embedding from bert, 2021.

[4] Kostadin Mishev, Ana Gjorgjevikj, Irena Vodenska, Lubomir T. Chitkushev, and Dimitar Trajanov. Evaluation of sentiment analysis in finance: From lexicons to transformers. *IEEE Access*, 8:131662–131682, 2020. doi: 10.1109/ACCESS.2020.3009626. URL https://ieeexplore.ieee.org/abstract/document/9142175.

[5] Yuzheng Zhai, Arthur Hsu, and Saman K. Halgamuge. Combining news and technical indicators in daily stock price trends prediction. In Derong Liu, Shumin Fei, Zengguang Hou, Huaguang Zhang, and Changyin Sun, editors, *Advances in Neural Networks – ISNN 2007*, pages 1087–1096, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-72395-0.

[6] A. K. M. Amanat Ullah, Fahim Imtiaz, Miftah Uddin Md Ihsan, Md. Golam Rabiul Alam, and Mahbub Majumdar. Combining machine learning classifiers for stock trading with effective feature extraction, 2021.

[7] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

[8] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, page 21260, 2008.

[9] Elie Bouri, Peter Molnár, Georges Azzi, David Roubaud, and Lars Ivar Hagfors. On the hedge and safe haven properties of bitcoin: Is it really more than a diversifier? *Finance Research Letters*, 20:192–198, 2017. ISSN 1544-6123. doi: https://doi.org/10.1016/j.frl.2016.09.025. URL https://www.sciencedirect.com/science/article/pii/S1544612316301817.

[10] David Yermack. Is bitcoin a real currency? an economic appraisal. NBER Working Papers 19747, National Bureau of Economic Research, Inc, 2013. URL https://EconPapers.repec.org/RePEc:nbr:nberwo:19747.

[11] Jamal Bouoiyour and Refk Selmi. What does bitcoin look like? *Annals of Economics and Finance*, 16(2):449–492, 2015. URL https://EconPapers.repec.org/RePEc:cuf:journl:y:2015:v:16:i:2:bouoiyour.

[12] Young Bin Kim, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, and Chang Hun Kim. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PLOS ONE*, 11(8):1–17, 08 2016. doi: 10.1371/journal.pone.0161197. URL https://doi.org/10.1371/journal.pone.0161197.

[13] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011. ISSN 1877-7503. doi: https://doi.org/10.1016/j.jocs.2010.12.007. URL https://www.sciencedirect.com/science/article/pii/S187775031100007X.

[14] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto 82*, pages 199–203. Springer, 1983.

[15] ec.europa.eu. The story of digicash and its ecash. https://ec.europa.eu/newsroom/cef/items/658303/en, 09 2019.

[16] Lam Pak Nian and David LEE Kuo Chuen. Chapter 1 - introduction to bitcoin. In David Lee Kuo Chuen, editor, *Handbook of Digital Currency*, pages 5–30. Academic Press, San Diego, 2015. ISBN 978-0-12-802117-0. doi: https://doi.org/10.1016/B978-0-12-802117-0.00001-1. URL https://www.sciencedirect.com/science/article/pii/B9780128021170000011.

[17] P.E. Tsinaslanidis and A.D. Zapranis. *Technical Analysis for Algorithmic Pattern Recognition*. Springer International Publishing, 2016. ISBN 9783319353951.

[18] Mohammed Mudassir, Shada Bennbaia, Devrim Unal, and Mohammad Hammoudeh. Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*, 2020. doi: 10.1007/s00521-020-05129-6. URL https://doi.org/10.1007/s00521-020-05129-6.

[19] Erdinc Akyildirim, Ahmet Goncu, and Ahmet Sensoy. Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*, 297(1):3–36, 2021. doi: 10.1007/s10479-020-03575-y. URL https://doi.org/10.1007/s10479-020-03575-y.

[20] Yauheniya Shynkevich, T.M. McGinnity, Sonya A. Coleman, Ammar Belatreche, and Yuhua Li. Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, 264:71–88, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.11.095. URL https://www.sciencedirect.com/science/article/pii/S0925231217311074. Machine learning in finance.

[21] Kyung Keun Yun, Sang Won Yoon, and Daehan Won. Prediction of stock price direction using a hybrid ga-xgboost algorithm with a three-stage feature engineering process. *Expert Systems with Applications*, 186:115716, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.115716. URL https://www.sciencedirect.com/science/article/pii/S0957417421010988.

[22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[23] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32(10):2513–2522, 2005. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2004.03.016. URL https://www.sciencedirect.com/science/article/pii/S0305054804000681. Applications of Neural Networks.

[24] Yakup Kara, Melek Acar Boyacioglu, and Ömer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert Systems with Applications*, 38(5):5311–5319, 2011. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2010.10.027. URL https://www.sciencedirect.com/science/article/pii/S0957417410011711.

[25] Diya Wang and Yixi Zhao. Using news to predict investor sentiment: Based on svm model. *Procedia Computer Science*, 174:191–199, 2020. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2020.06.074. URL https://www.sciencedirect.com/science/article/pii/S187705092031588X. 2019 International Conference on Identification, Information and Knowledge in the Internet of Things.

[26] Bhawna Panwar, Gaurav Dhuriya, Prashant Johri, Sudeept Singh Yadav, and Nitin Gaur. Stock market prediction using linear regression and svm. In *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 629–631, 2021. doi: 10.1109/ICACITE51222.2021.9404733.

[27] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar. Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, 167: 599–606, 2020. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2020.03.326. URL https://www.sciencedirect.com/science/article/pii/S1877050920307924. International Conference on Computational Intelligence and Data Science.

[28] Zexin Hu, Yiqi Zhao, and Matloob Khushi. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1), 2021. ISSN 2571-5577. doi: 10.3390/asi4010009. URL https://www.mdpi.com/2571-5577/4/1/9.

[29] Sidra Mehtab, Jaydip Sen, and Abhishek Dutta. Stock price prediction using machine learning and lstm-based deep learning models. In Sabu M. Thampi, Selwyn Piramuthu, Kuan-Ching Li, Stefano Berretti, Michal Wozniak, and Dhananjay Singh, editors, *Machine Learning and Metaheuristics Algorithms, and Applications*, pages 88–106, Singapore, 2021. Springer Singapore. ISBN 978-981-16-0419-5. URL https://link.springer.com/chapter/10.1007/978-981-16-0419-5_8#citeas.

[30] Manuel R. Vargas, Carlos E. M. dos Anjos, Gustavo L. G. Bichara, and Alexandre G. Evsukoff. Deep leaming for stock market prediction using technical indicators and financial news articles. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018. doi: 10.1109/IJCNN.2018.8489208.

[31] David M. Q. Nelson, Adriano C. M. Pereira, and Renato A. de Oliveira. Stock market's price movement prediction with lstm neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1419–1426, 2017. doi: 10.1109/IJCNN.2017.7966019.

[32] Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, 538:142–158, 2020. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2020.05.066. URL https://www.sciencedirect.com/science/article/pii/S0020025520304692.

[33] A.A. Patel and A.U. Arasanipalai. *Applied Natural Language Processing in the Enterprise*. O'Reilly Media, 2021. ISBN 9781492062523.

[34] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco. *A Practical Guide to Sentiment Analysis*. Socio-Affective Computing. Springer International Publishing, 2017. ISBN 9783319553948.

[35] Dominic Widdows and Beate Dorow. A graph model for unsupervised lexical acquisition. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://aclanthology.org/C02-1114.pdf.

[36] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani, et al. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204, 2010. URL http://lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf.

[37] C. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225, May 2014. doi: 10.1609/icwsm.v8i1.14550. URL https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[38] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. URL https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?ref=https://githubhelp.com.

[39] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, and Sweta Tiwari. Sentiment analysis of review datasets using naive bayes and k-nn classifier. *arXiv preprint arXiv:1610.09982*, 2016. URL https://arxiv.org/abs/1610.09982.

[40] Munir Ahmad, Shabib Aftab, and Iftikhar Ali. Sentiment analysis of tweets using svm. *Int. J. Comput. Appl*, 177(5):25–29, 2017. URL https://www.researchgate.net/profile/Shabib-Aftab-2/publication/321084834_Sentiment_Analysis_of_Tweets_using_SVM/links/5a1497b90f7e9b925cd514b0/Sentiment-Analysis-of-Tweets-using-SVM.pdf.

[41] M Ali Fauzi. Random forest approach fo sentiment analysis in indonesian. *Indones. J. Electr. Eng. Comput. Sci*, 12:46–50, 2018. URL https://www.researchgate.net/profile/Muhammad-Fauzi-6/publication/327060733_Random_Forest_Approach_for_Sentiment_Analysis_in_Indonesian_Language/links/5d305ff3458515c11c39adfd/Random-Forest-Approach-for-Sentiment-Analysis-in-Indonesian-Language.pdf.

[42] Ashima Yadav and Dinesh Kumar Vishwakarma. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385, 2020. doi: 10.1007/s10462-019-09794-5. URL https://doi.org/10.1007/s10462-019-09794-5.

[43] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[45] Ben Lutkevich. Bert language model, 2020. URL https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model.

[46] Priyank Sonkiya, Vikas Bajpai, and Anukriti Bansal. Stock price prediction using bert and gan, 2021.

[47] Werner Antweiler and Murray Z. Frank. Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, 59(3):1259–1294, 2004. doi: https://doi.org/10.1111/j.1540-6261.2004.00662.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2004.00662.x.

[48] Tushar Rao and Saket Srivastava. Twitter sentiment analysis: How to hedge your bets in the stock markets. *CoRR*, abs/1212.1107, 2012. URL http://arxiv.org/abs/1212.1107.

[49] Gabriele Ranco, Darko Aleksovski, Guido Caldarelli, Miha Grčar, and Igor Mozetič. The effects of twitter sentiment on stock price returns. *PLOS ONE*, 10(9):1–21, 09 2015. doi: 10.1371/journal.pone.0138441. URL https://doi.org/10.1371/journal.pone.0138441.

[50] Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. Exploiting topic based Twitter sentiment for stock prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 24–29, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-2005.

[51] Ryan G. Chacon, Thibaut G. Morillon, and Ruixiang Wang. Will the reddit rebellion take you to the moon? evidence from wallstreetbets. *Financial Markets and Portfolio Management*, 37(1):1–25, 2023. doi: 10.1007/s11408-022-00415-w. URL https://doi.org/10.1007/s11408-022-00415-w.

[52] Juan Andrés Talamás Carvajal. Social media effects on the market: Reddit data analysis on stocks.

[53] Suwan (Cheng) Long, Brian Lucey, Ying Xie, and Larisa Yarovaya. "i just like the stock": The role of reddit sentiment in the gamestop share rally. *Financial Review*, 58(1):19–37, 2023. doi: https://doi.org/10.1111/fire.12328. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/fire.12328.

[54] Tushar Rao, Saket Srivastava, et al. Analyzing stock market movements using twitter sentiment analysis. 2012.

[55] Ayman E Khedr, Nagwa Yaseen, et al. Predicting stock market behavior using data mining technique and news sentiment analysis. *International Journal of Intelligent Systems and Applications*, 9(7):22, 2017.

[56] Anshul Mittal and Arpit Goel. Stock prediction using twitter sentiment analysis. *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)*, 15:2352, 2012.

[57] Matheus Gomes Sousa, Kenzo Sakiyama, Lucas de Souza Rodrigues, Pedro Henrique Moraes, Eraldo Rezende Fernandes, and Edson Takashi Matsubara. Bert for stock market sentiment analysis. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1597–1601, 2019. doi: 10.1109/ICTAI.2019.00231.

[58] Jethin Abraham, Daniel Higdon, John Nelson, and Juan Ibarra. Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1(3):1, 2018.

[59] Franco Valencia, Alfonso Gómez-Espinosa, and Benjamín Valdés-Aguirre. Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. *Entropy*, 21(6), 2019. ISSN 1099-4300. doi: 10.3390/e21060589. URL https://www.mdpi.com/1099-4300/21/6/589.

[60] Kwansoo Kim, Sang-Yong Tom Lee, and Said Assar. The dynamics of cryptocurrency market behavior: sentiment analysis using markov chains. *Industrial Management & Data Systems*, 122(2):365–395, 2022.

[61] Raj Parekh, Nisarg P. Patel, Nihar Thakkar, Rajesh Gupta, Sudeep Tanwar, Gulshan Sharma, Innocent E. Davidson, and Ravi Sharma. Dl-guess: Deep learning and sentiment analysis-based cryptocurrency price prediction. *IEEE Access*, 10:35398–35409, 2022. doi: 10.1109/ACCESS.2022.3163305.

[62] Nan Jing, Zhao Wu, and Hefei Wang. A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. *Expert Systems with Applications*, 178:115019, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.115019. URL https://www.sciencedirect.com/science/article/pii/S0957417421004607.

[63] Zhigang Jin, Yang Yang, and Yuhong Liu. Stock closing price prediction based on sentiment analysis and lstm. *Neural Computing and Applications*, 32(13):9713–9729, 2020. doi: 10.1007/s00521-019-04504-2. URL https://doi.org/10.1007/s00521-019-04504-2.

[64] Shengting Wu, Yuling Liu, Ziran Zou, and Tien-Hsiung Weng. S_i_lstm: stock price prediction based on multiple data sources and sentiment analysis. *Connection Science*, 34(1):44–62, 2022. doi: 10.1080/09540091.2021.1940101. URL https://doi.org/10.1080/09540091.2021.1940101.

[65] Artur Meynkhard. Effect of bitcoin volatility on altcoins pricing. In Radek Silhavy, Petr Silhavy, and Zdenka Prokopova, editors, *Software Engineering Perspectives in Intelligent Systems*, pages 652–664, Cham, 2020. Springer International Publishing. ISBN 978-3-030-63322-6.

[66] Pushshift reddit api. URL https://github.com/pushshift/api.

[67] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[68] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Appendices

A. Data overview and labels distribution

A.1. Data Overview

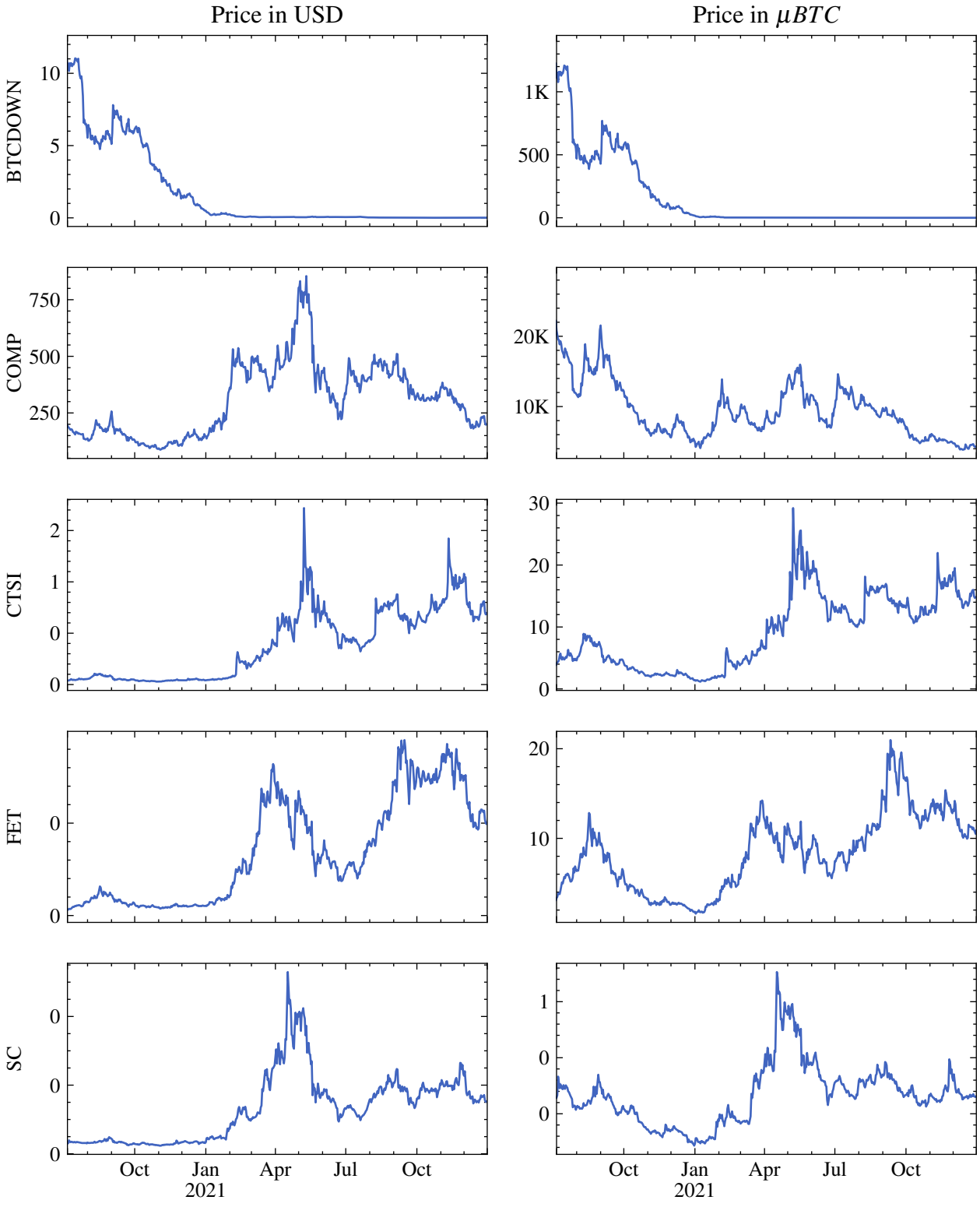


Figure A.1: Cryptocurrencies with mid-trade volume price comparison in USD and μBTC

A.2. Labels distribution among cryptocurrencies with the different threshold for the price change

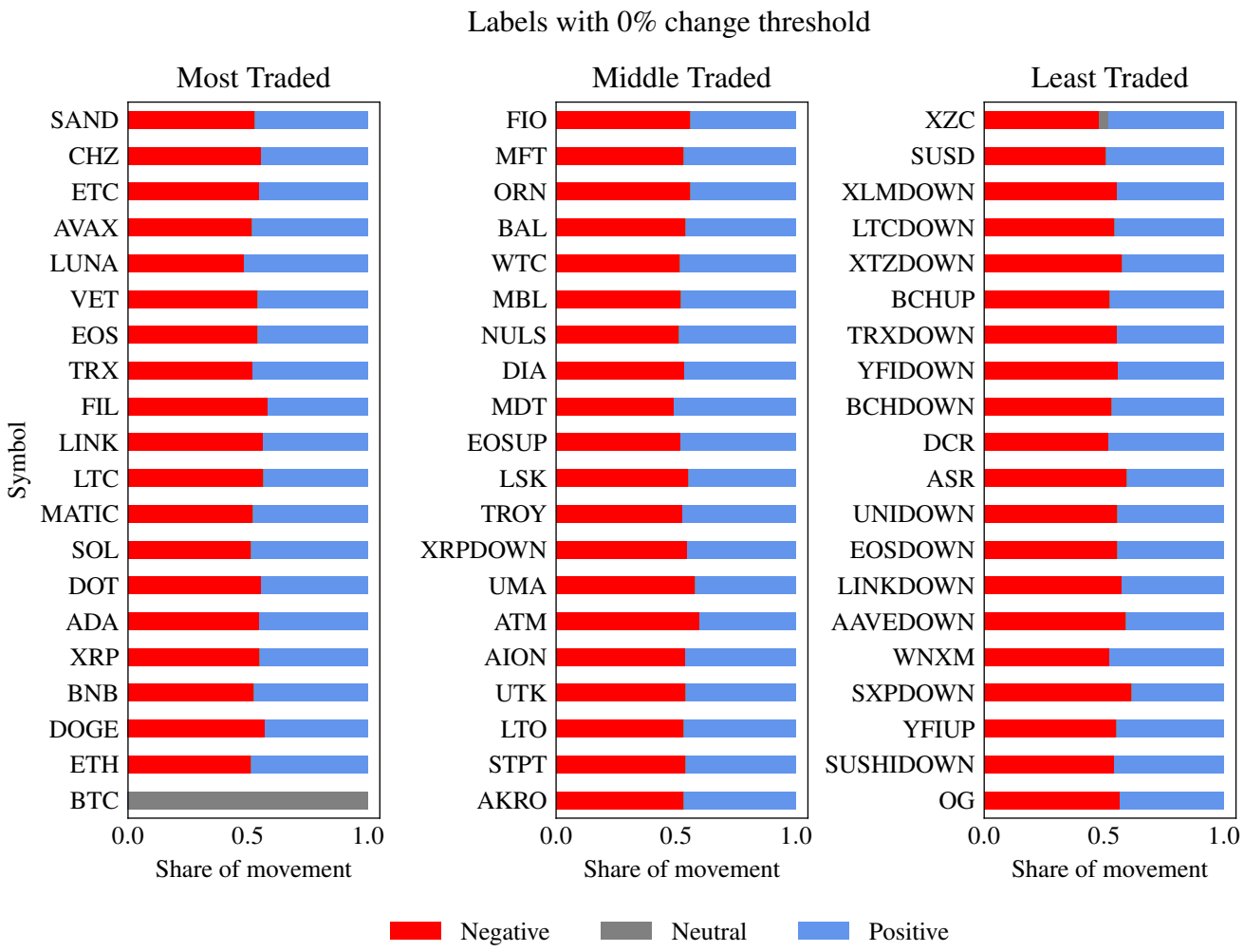


Figure A.2: Sample of cryptocurrencies label amount with 0% threshold for the price change in the upcoming 7 days

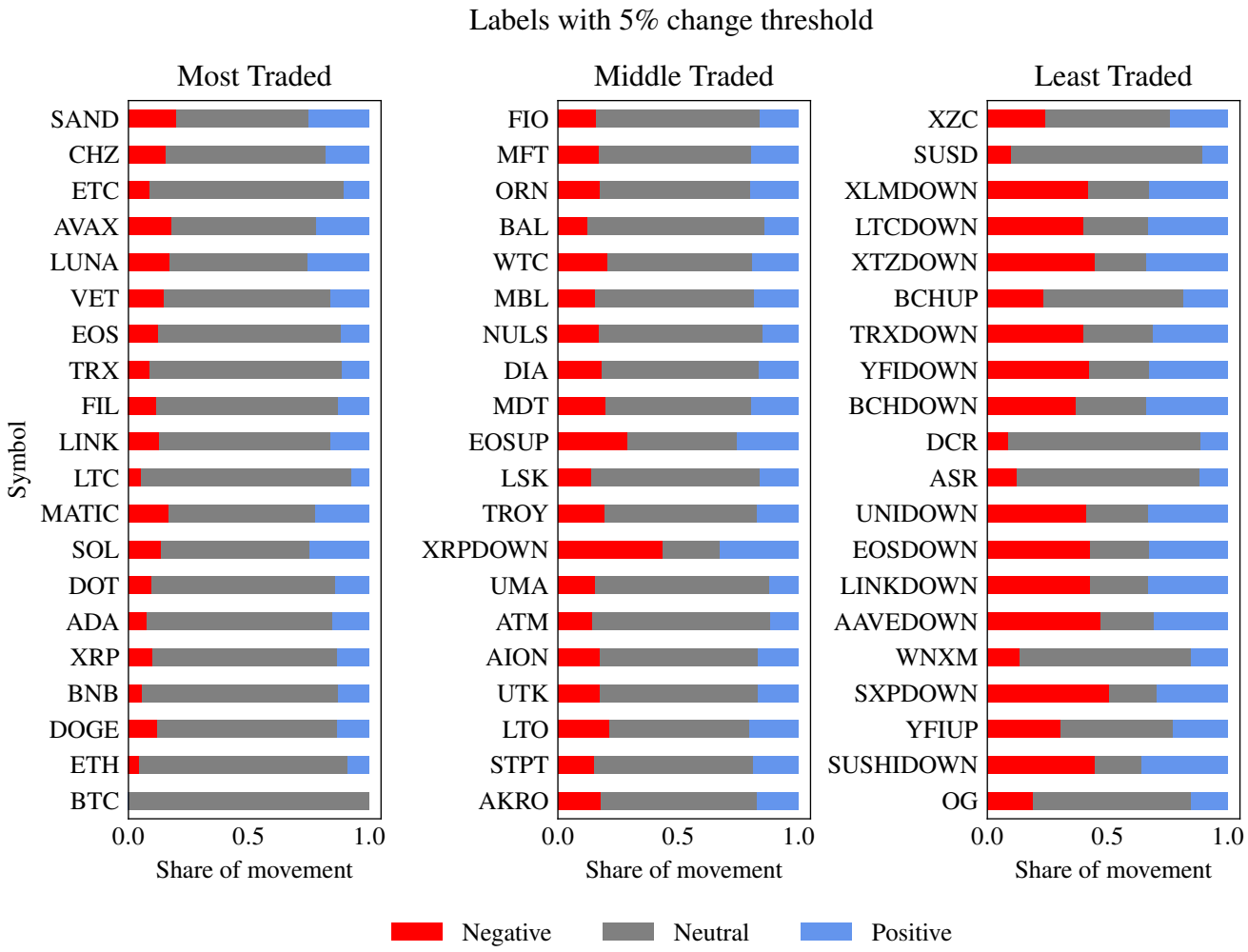


Figure A.3: Sample of cryptocurrencies label amount with 5% threshold for the price change in the upcoming 7 days

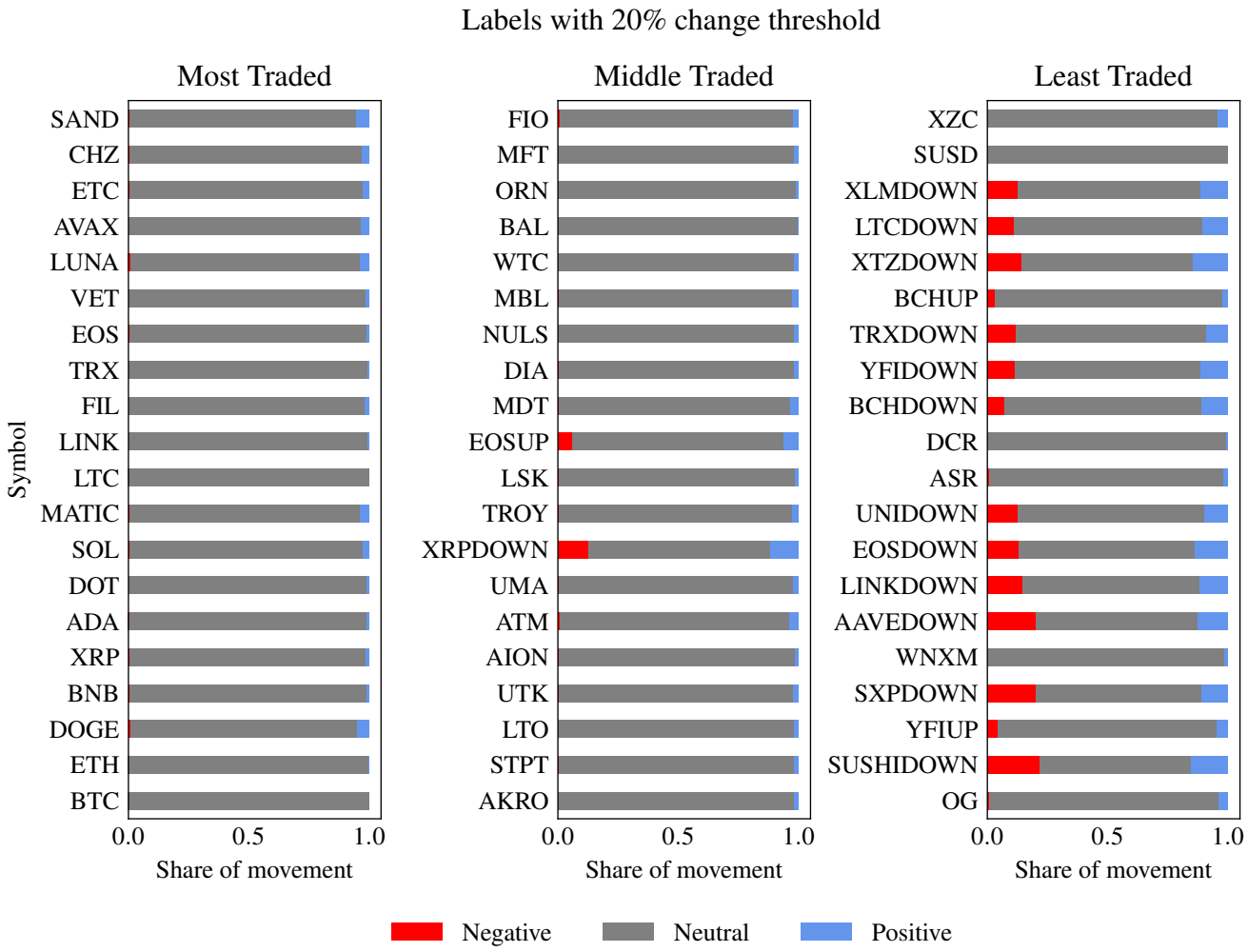


Figure A.4: Sample of cryptocurrencies label amount with 20% threshold for the price change in the upcoming 7 days

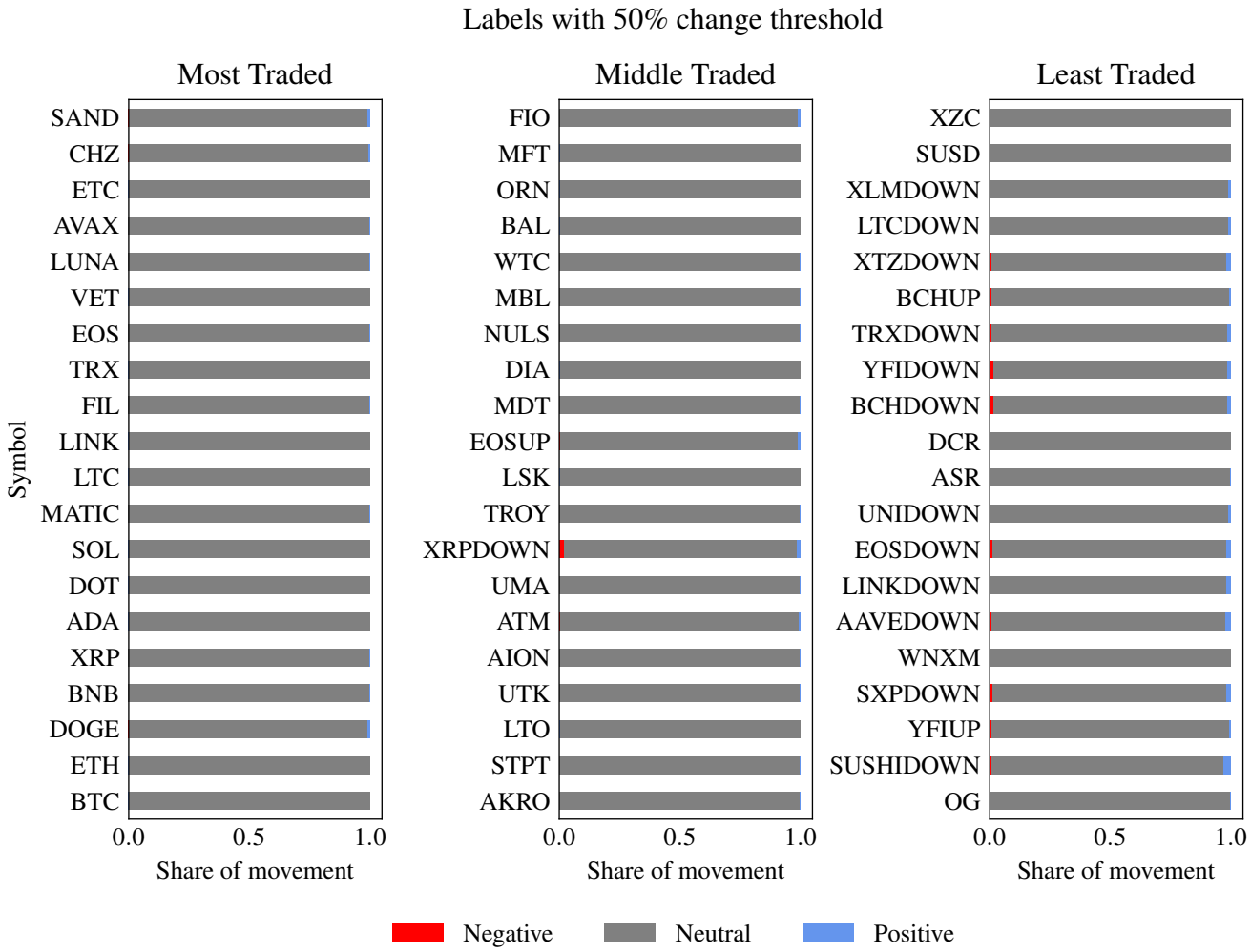


Figure A.5: Sample of cryptocurrencies label amount with 0% threshold for the price change in the upcoming 7 days

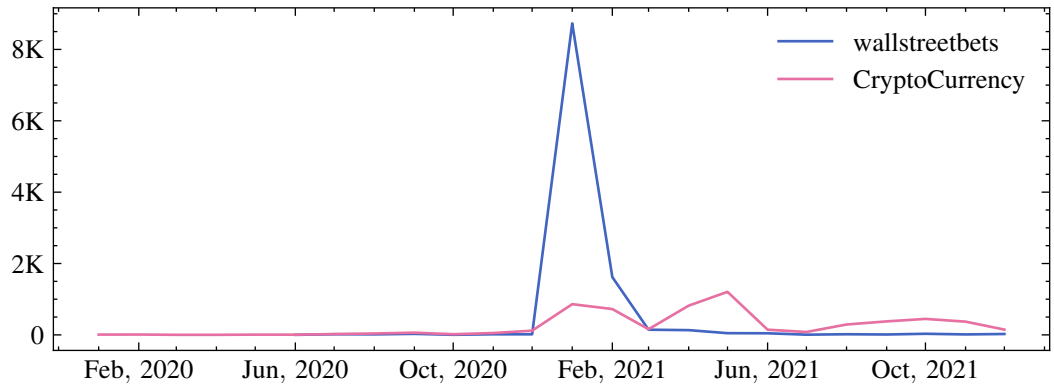


Figure A.6: Monthly labeled (as per equation 2.2) Reddit posts count per subreddit

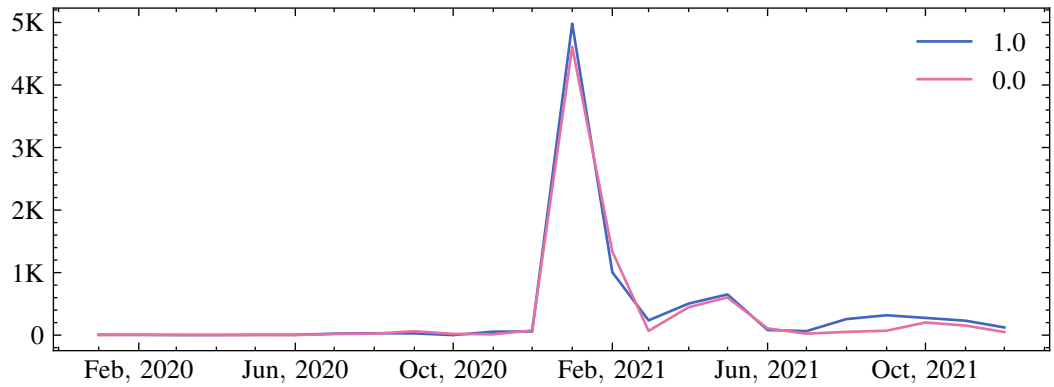


Figure A.7: Monthly labeled (as per equation 2.2) Reddit posts count per label

B. Technical indicators list

List of used technical indicators, quoted from a python package *ta*[9] (which was used for technical indicators computation) readme page:

Volume:

- Money Flow Index (MFI)
- Accumulation/Distribution Index (ADI)
- On-Balance Volume (OBV)
- Chaikin Money Flow (CMF)
- Force Index (FI)
- Ease of Movement (EoM, EMV)
- Volume-price Trend (VPT)
- Negative Volume Index (NVI)
- Volume Weighted Average Price (VWAP)

Volatility:

- Average True Range (ATR)
- Bollinger Bands (BB)
- Keltner Channel (KC)
- Donchian Channel (DC)
- Ulcer Index (UI)

Trend:

- Simple Moving Average (SMA)
- Exponential Moving Average (EMA)
- Weighted Moving Average (WMA)
- Moving Average Convergence Divergence (MACD)
- Average Directional Movement Index (ADX)
- Vortex Indicator (VI)
- Trix (TRIX)
- Mass Index (MI)
- Commodity Channel Index (CCI)
- Detrended Price Oscillator (DPO)
- KST Oscillator (KST)

[9] List to the package GitHub page: `https://github.com/bukosabino/ta`

- Ichimoku Kinkō Hyō (Ichimoku)
- Parabolic Stop And Reverse (Parabolic SAR)
- Schaff Trend Cycle (STC)

Momentum:

- Relative Strength Index (RSI)
- Stochastic RSI (SRSI)
- True strength index (TSI)
- Ultimate Oscillator (UO)
- Stochastic Oscillator (SR)
- Williams
- Awesome Oscillator (AO)
- Kaufman's Adaptive Moving Average (KAMA)
- Rate of Change (ROC)
- Percentage Price Oscillator (PPO)
- Percentage Volume Oscillator (PVO)

Others:

- Daily Return (DR)
- Daily Log Return (DLR)
- Cumulative Return (CR)