844G1: Dissertation

Social Media Data Analysis of Cryptocurrency Markets

Prepared by: Mert OLCAMAN

MSc Data Science, University of Sussex, Brighton, UK
August, 2024

Contents

Table of Figures

Table of Tables

1. Abstract

This study explores how social media sentiment influences cryptocurrency markets, using advanced natural language processing and machine learning techniques. As social media plays an increasing role in financial markets, especially in the unpredictable world of cryptocurrencies, this research focuses on building a model that uses sentiment analysis to predict price movements. By analyzing data from Reddit and other financial sources, the study employs Long Short-Term Memory (LSTM) models to make sense of time-series data and sentiment trends. The results show that including sentiment analysis significantly improves the accuracy of price predictions, highlighting a strong link between what people are saying online and how cryptocurrency prices move. However, the study also points out some limitations, like relying on data from just one social media platform and covering only a short period. This suggests that future research should look at a wider range of data and sentiment sources. Overall, this research adds valuable insights into financial forecasting and offers practical tools for traders and analysts to make more informed decisions based on social media trends.

2. Introduction

The concept of cryptocurrency has its roots in the early 1990s when pioneers like David Chaum and his colleagues introduced the idea of electronic money, aiming to create untraceable transactions without the need for banks (Chaum et al., 1990). Satoshi Nakamoto's dream came true with the concept of the first cryptocurrency called Bitcoin in 2009, a decentralized system relying on blockchain technology to ensure transparency, security, and trust in digital transactions (Nakamoto, 2008).

Bitcoin has a maximum supply of 21 million coins. The current number of existing bitcoins can be checked on the website called bitbo.

Cryptocurrencies gained significant popularity during the pandemic as people embraced the digital era, leading to the creation of many new cryptocurrencies. The number of cryptocurrencies surged post-COVID. Besides Bitcoin, other notable cryptocurrencies include Ethereum, Tether, BNB, and XRP.

During the early stages of COVID-19, people consumed more digital content from platforms like social media, becoming more familiar with cryptocurrencies. Over time, cryptocurrencies emerged as an alternative investment method due to their perceived potential.

Consequently, many content creators started to share their thoughts on cryptocurrencies, generating substantial data. Some traders, particularly Bitcoin whales (those holding more than 10,000 BTC), used cryptocurrencies as speculative tools. For instance, Elon Musk's tweets caused significant price fluctuations in cryptocurrencies like Dogecoin. His tweets led to a 24,000% increase in Dogecoin's price, with another 400% leap in April 2021.

Traditional financial analysis methods became insufficient for predicting cryptocurrency trends due to the dynamic data flow on the internet and the 24/7 nature of crypto markets. Researchers turned to social media to correlate trends with community sentiments and influential posts.

Some studies have analyzed social media data using Natural Language Processing (NLP) methods to gauge sentiment. However, understanding the nuances of human language, such as sarcasm, idioms, and negation, remains challenging for NLP models. Despite these challenges, large language models (LLMs) have made it possible to roughly understand the sentiment of social media posts. This capability has opened up new avenues for analyzing vast amounts of unstructured data, allowing researchers and traders alike to tap into the collective mood of the market in near real-time.

Cryptocurrency markets are known for their rapid fluctuations and complex dynamics, making it crucial to use advanced time-series forecasting models to anticipate price changes. Among these, Long Short-Term Memory (LSTM) models, a type of Recurrent Neural Network (RNN), have proven to be particularly effective due to their ability to learn and remember patterns over long periods (Hochreiter & Schmidhuber, 1997).

When LSTMs are combined with sentiment analysis from social media, they offer a robust forecasting approach that considers both historical trends and real-time market sentiment, leading to trading strategies that can significantly enhance decision-making processes for investors and market participants (Seabe et al., 2023).

The primary objective of this study is to investigate the integration of sentiment analysis with LSTM modeling to predict cryptocurrency prices. Specifically, this research aims to:

1. **Develop a Social Sentiment Index:** Analyze social media content to create an index which can reflect the collective sentiment relevant to cryptocurrencies.

2. **Enhance Time-Series Forecasting:** Integrate sentiment analysis into LSTM models to improve the accuracy of cryptocurrency price predictions.

3. **Evaluate the Effectiveness of Combined Models:** Compare the performance of traditional financial analysis method with models which incorporate sentiment analysis to see which one performs better in predicting.

By achieving these objectives, this study seeks to provide actionable insights for traders and investors, enabling more informed decision-making in the cryptocurrency market.

3. Literature Review

3.1. History of Cryptocurrency

The concept of cryptocurrency dates back to the 1990s, with the idea of electronic money introduced by David Chaum, Amos Fiat, and Moni Naor. Their goal was to create an efficient system for untraceable electronic money that would eliminate the need for bank confirmation. They proposed using RSA (digital signature) related to cryptographic techniques (Chaum et al., 1990).

In 2009, Satoshi Nakamoto published a paper introducing Bitcoin, a new form of cryptocurrency. According to the paper, relying on financial institutions for electronic payments leads to trust issues, extra transaction fees, and inefficiencies. Nakamoto described an electronic coin as a chain of digital signatures, facilitating transfers by referencing the hash of the previous transaction and the public key of the next owner. However, this method does not prevent double-spending (Nakamoto, 2008).

To address this issue, Nakamoto proposed a decentralized timestamp server to create a chain of timestamps through cryptographic hashing. Each timestamp links to the previous one, forming a chain of blocks known as the blockchain. This blockchain acts as a public ledger that transparently and immutably records all transactions. Essentially, a blockchain is a shared digital record of transactions in a cryptocurrency network, secured with advanced cryptographic methods (Nakamoto, 2008).

3.2. Text Representation in NLP

Initially, traditional methods such as Bag of Words (BoW) were used, which faced challenges such as high dimensionality and sparse representations. To address these, dimensionality reduction techniques such as Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) were used, allowing for more manageable and interpretable latent variables.

Recent advances have introduced neural network-based word embedding techniques. For example, the word2vec method has been effective in capturing both syntactic and semantic relationships between words. More complex approaches, such as transformer-based language models such as BERT, have significantly improved the ability to represent deep semantic and syntactic information, making them highly effective in financial text analysis (Farimani et al., 2022).

3.3. Word Embedding

Word embedding techniques have revolutionized the field of NLP by providing dense vector representations for words based on their contextual usage. The review highlights the various applications of these techniques in financial market forecasting. The word2vec method introduced by Mikolov is widely used to create word vectors that capture semantic relationships. Further developments such as BERT (Bidirectional Encoder Representations from Transformers) have improved the ability to more accurately understand the context of words in a document.

BERT showed superior performance, especially in financial text analysis. It uses a bidirectional training approach that allows the model to consider context from both aspects. This method has been effectively applied to summarize news and select important information for investors. Other new models, such as GPT-3, continue to push the boundaries of what can be achieved with word embeddings, offering deeper contextual understanding and better performance on a variety of NLP tasks (Farimani et al., 2022).

3.4. Large Language Models (LLMs)

3.4.1. Transformers

Transformer model, a novel network architecture for sequence transduction tasks that relies entirely on self-attention mechanisms. The model shows superior performance on machine translation tasks, demonstrating higher quality, better parallelizability, and reduced training time compared to previous state-of-the-art models. The goal of attention mechanism is to determine which parts of the input sequence are most relevant to each position in the output sequence (Vaswani et al., 2017).

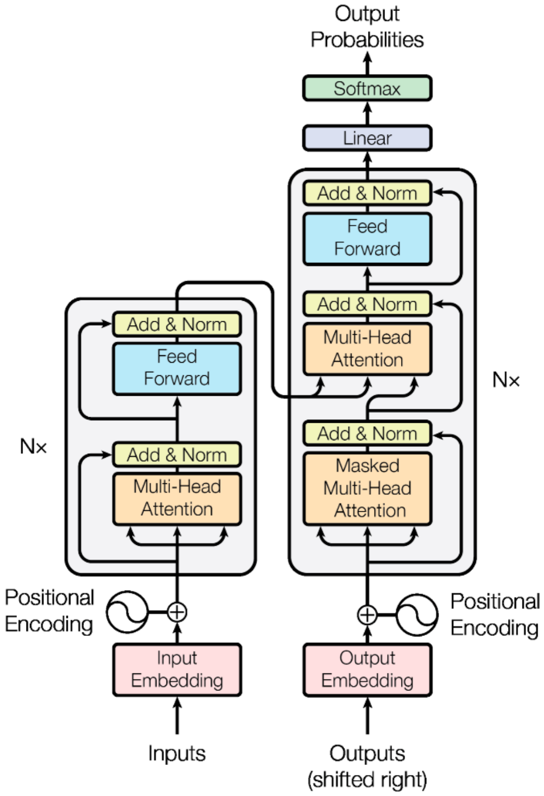


Figure 1: Transformers model representation

Transformers consist of an encoder-decoder structure, each built with self-attention and feed-forward layers as it can be seen in Figure 1. The encoder includes a stack of identical layers, each with multi-head self-attention and a feed-forward network, employing residual connections and layer normalization. The decoder is similar but includes an additional sub-layer for multi-head attention over the encoder's output and masking to prevent positions from attending to subsequent positions. This model demonstrates superior performance in machine translation tasks, offering higher quality, better parallelizability, and reduced training time compared to previous models.

By replacing recurrent layers with self-attention mechanisms, the Transformer allows for significant parallelization. It connects all positions with a constant number of sequential operations, providing computational efficiency, especially for shorter sequences. Moreover, self-attention models are more interpretable, with individual attention heads performing different tasks. Trained on the WMT 2014 English-to-German and English-to-French datasets, the Transformer achieved state-of-the-art BLEU scores with reduced training costs and generalized well to other tasks, such as English constituency parsing. The Transformer is an efficient and scalable model for sequence transduction tasks, leveraging self-attention to replace recurrent layers, and future work includes extending the model to other modalities and investigating local attention mechanisms for large inputs and outputs (Vaswani et al., 2017).

3.4.2. BERT

Bert (Bidirectional Encoder Representations from Transformers) is pre-trained for deep bidirectional representations from unlabeled data, using left-to-right and right-to-left architecture to better capture the meanings of words with a masked language model. Published by Google researchers. In the masked language model, some tokens from the input are randomly masked and the model predicts it based on the context (Devlin et al., 2018).

3.4.3. RoBERTa

The paper "RoBERTa: A Robustly Optimized BERT Pretraining Approach" presents a comprehensive study on optimizing the pretraining methodology of BERT (Bidirectional Encoder Representations from Transformers). The authors identify significant undertraining in the original BERT model and propose a series of modifications to enhance its performance. These modifications include extending the training duration, increasing batch sizes, eliminating the next sentence prediction objective, training on longer sequences, and implementing dynamic masking patterns (Liu et al., 2019).

3.4.4. DistilBERT

A distilled version of BERT. The size of the model is reduced by 40%, while 97% of its capability is the same as the normal model and 60% faster. DistilBERT is optimized for efficiency, making it suitable for on-device applications with limited computational resources (Sanh et al., 2019).

3.4.5. FinBERT

FinBERT is a specialized language model developed specifically for financial texts, utilizing a training set comprised of corporate reports, earnings call transcripts, and analyst reports. This model is built on the BERT architecture and is fine-tuned to enhance performance on financial NLP tasks, particularly sentiment analysis. It significantly surpasses general BERT models by leveraging a domain-specific corpus of approximately 4.9 billion tokens. FinBERT provides a valuable tool for financial practitioners, offering improved accuracy in interpreting market sentiments, which is crucial for informed trading and strategic decisions in finance. The availability of FinBERT enables wider access to advanced NLP technology in the financial sector, streamlining the adoption of deep learning tools without extensive resource investment (Yang et al., 2020).

3.4.6. sBERT

Sentence-BERT (SBERT) is an improved version of the BERT model designed to efficiently calculate semantic text similarity. Unlike the standard BERT model, which requires a significant amount of computational resources and time due to the need to input both sentences into the network for comparison, SBERT uses Siamese and ternary network structure. This innovative architecture allows SBERT to generate semantically meaningful sentence embeddings that can be compared using cosine similarity. As a result, SBERT significantly reduces computational effort from approximately 65 hours to just a few seconds when processing tasks such as finding the most similar pair in a large collection of sentences, while maintaining the high accuracy of BERT (Reimers & Gurevych, 2019).

3.5. Sentiment Analysis

3.5.1. Text Blob

TextBlob is a Python library which helps to process text data and perform a variety of Natural Language Processing (NLP) tasks. It is used in common NLP tasks, including rule-based parsing and sentiment analysis based on a combination of a predefined dictionary. TextBlob assigns a polarity score ranging from -1 (very negative) to 1 (very positive) and a subjectivity score ranging from 0 (very objective) to 1 (very subjective) to a given text. This approach makes TextBlob easy to use for basic sentiment analysis tasks (Loria, S., 2018).

3.5.2. VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER is a popular sentiment analysis tool generally used for social media texts. It uses a lexicon of words with associated emotional intensities as well as a set of grammatical and syntactic rules. VADER provides individual scores for positive, negative, and neutral emotions. Besides that, a normalized composite emotion score ranging from −1 (most negative) to 1 (most positive) is given. Its design enables effective sentiment analysis of short, informal texts such as tweets and reviews (Hutto & Gilbert, 2015).

3.6. Market Prediction

3.6.1. Recurrent Neural Network

Recurrent Neural Networks (RNNs) are designed to handle sequence-based tasks by using loops in their architecture, allowing information to persist across time steps. This persistence is crucial for tasks like language modeling, where understanding each word depends on the context provided by previous words (Goodfellow et al., 2016).

RNNs can struggle with long-term dependencies, where relevant information needed for a task is located far back in the sequence. This issue, known as the vanishing gradient problem, makes it difficult for RNNs to learn from long-term patterns (Bengio et al., 1994).

3.6.2. LSTM Networks

LSTM networks, a specialized type of RNN, were introduced by Hochreiter and Schmidhuber (1997) to overcome the vanishing gradient problem. LSTMs are designed to remember information for long periods, making them particularly effective for tasks involving long sequences, such as speech recognition, language modeling, and financial forecasting (Hochreiter&Schmidhuber, 1997).

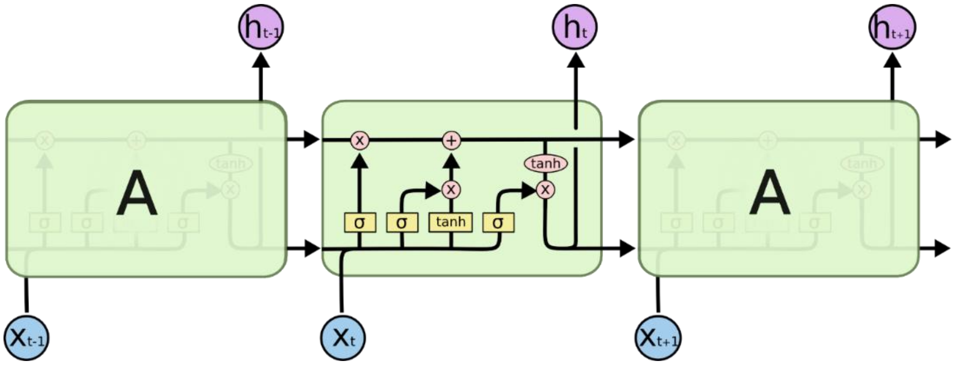


Figure 2: The LSTM architecture (t-1: previous time step, t: current time step, t+1: next time step)

LSTMs consist of repeating modules, each containing interacting layers that manage the flow of information through memory cells. These cells are regulated by gates as it can be seen in Figure 2.

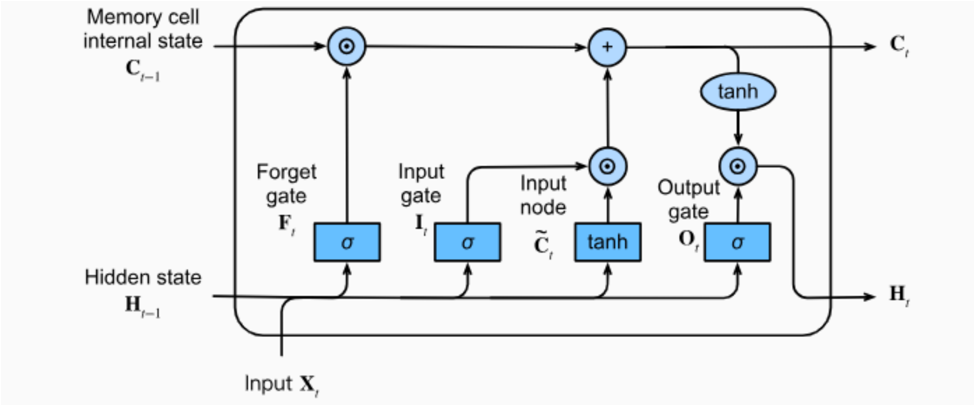


Figure 3: LSTM inner architecture

In Figure 3, the detailed architecture of a LSTM can be seen. In an LSTM cell, the hidden state from the previous time step (H_{t-1}) is combined with the current input (X_i) to determine how much past information should be retained and how much new information should be added. This is controlled through three gates (Zhang et al., 2021, pp. 420–423):

- **Forget Gate (F_t)**: It helps to decide the model which information is excluded.
- **Input Gate (I_t)**: It helps to decide which information is added.
- **Output Gate (O_t)**: It helps to decided which information is passed to another (Gers et al., 2000)

This structure allows LSTMs to effectively manage and retain relevant information across long sequences.

Several LSTM variants exist, such as Bidirectional LSTMs, which process input sequences in both directions (left-to-right, and right-to-left), and Gated Recurrent Units (GRUs), which simplify the LSTM architecture while still performing well on various tasks (Cho et al., 2014).

While LSTMs have revolutionized what RNNs can achieve, attention mechanisms and newer models like Grid LSTMs are emerging as the next steps in sequence learning, promising even more powerful and flexible neural networks (Kalchbrenner et al., 2015).

4. Previous Studies

4.1. Sentiment Analysis

In this (Arslan, 2024), the short tweets which had less than 5 words are excluded from the dataset. Then, all characters in the tweets are converted to lowercase followed by removing unnecessary elements such as non-alphabetical characters, emojis, and website URLs. Stemming method is applied, the method where a couple of characters at the end of a word are removed without the knowledge, after removing stop words and splitting the sentences into word representations. As for sentiment analysis, TextBlob is used and same-day tweets are grouped together to calculate the ratio for each day using a threshold for the score.

In another research (Pekin et al., 2024), the punctuation is removed and divided into tokens. Then, the stopwords are removed and lemmatising is applied on finance news data. As for the sentiment part, TextBLOB, VADAR, and FinBERT is applied.

There is a research (Vlahavas & Vakali, 2024), where the emojis were replaced with their own text representations. If there is a smiling face, it was replaced by the text of smiley_face. However, there might no need to convert them into text versions since most of the LLMs have capability of understanding the meaning of emojis. However, it can be used with VADER to understand the sentiment of a sentence. In preprocessing part, there was no different method than other researchs by applying the methods such as tokenization, lowercasing, removing unnecessary elements etc.

In Figure 4, text has been represented by word embeddings more than other representations among the researchs according to one paper.

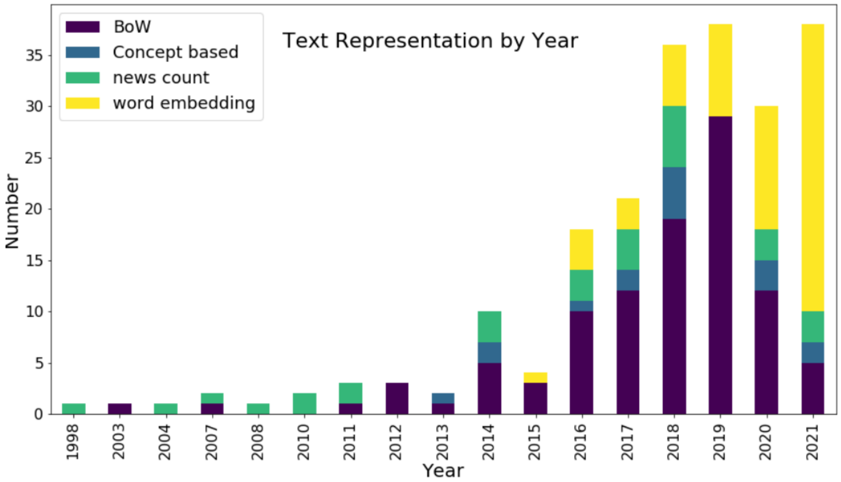


Figure 4: Distribution of reviewed works based on the text representation methods (Farimani, Jahan, & Fard, 2022).

According to the research, sentiment analysis was performed in different ways. The models which gave the best result for each approach can be seen in Table 1. In the research, Financial Phrase Bank and SemEval 2017 Task 5 datasets were used. Additionally, some pre-processing steps, such as tokenization, and stop-word removal, and stemming were applied by extracting named entities, the process in which the tags are added instead of the words (ex: London is converted into <CITY>).

Method Name	Best Model	Accuracy	Precision	Recall	F1-Score
Lexical Rule-based	LM + XGB	0.65	0.60	0.86	0.71
Statistical methods	TF – IDF + SVD	0.83	0.82	0.85	0.84
Fixed word embedding encoders	GloVe (concat) + BiGRU + Attention	0.85	0.84	0.88	0.85
Sentence encoders	InferSent (FastText) + SVC	0.86	0.88	0.83	0.85
Contextual word embedding encoders	ELMo (concat) +BiGRU + Attention	0.89	0.89	0.89	0.89
Transformers	BART-Large	0.95	0.95	0.95	0.95

Table 1: Sentiment Analysis Performance by Different Methods

The research evaluates different sentiment analysis methods, showing that advanced models like Transformers significantly outperform traditional methods and simpler models. Also, it is showed that transformers have much better performance in sentiment analysis compared to lexical and statistical methods (Mishev et al., 2020).

In the research where LLMs are used in cryptocurrency sentiment analysis (Roumeliotis et al., 2024) the language models such as GPT-4, BERT, and FinBERT were used by fine-tuning using different optimizers (Adam and AdamW).

Model	Accuracy	Negative	Positive	Neutral
GPT-4	0.87	0.87	0.87	0.86
BERT-adamw	0.80	0.79	0.81	0.81
BERT-adam	0.83	0.82	0.84	0.85
FinBERT-adamw	0.84	0.84	0.82	0.85
FinBERT-adam	0.84	0.85	0.83	0.86

Table 2: LLM Performances in a Research

As it can be seen through Table 2, GPT-4 outperformed others, but BERT and FinBERT models were not that bad.

In this research (Table 9 in Bhatt, Ghazanfar, and Amirhosseini (2023, p. 8) presents the modeling results and F1 scores.), it was shown that using sentiment data outperformed the without-sentiment models. Also, in case of using roBERTa for sentiment analysis gave better result than using VADER in all machine learning models.

The sentiments of cryptocurrency-related tweets were found by dictionary-based method (*Tweet Sentiment Analysis for Cryptocurrencies*, 2021). At first, a full of words list was creted, and the sentiments were found through this list. Then, the tweets were filtered by taking some specific hastags into account. 95% threshold was specified to determine the neutral labelled tweets (lower than 95% was accepted as neutral). The accuracy in BERT was found as 45% in that way, while it went up to 53% in case of ignoring neutral class. Also, the accuracy score for both training and test set were obtained 82% and 77% respectively in the method where count vectorizer was applied followed by TF-IDF.

Translator	Sentiment Analyzer Model	Accuracy	Precision	Recall	F1	Specificity
LibreTranslate	Twitter-roberta-base	0.55	0.64	0.61	0.62	0.47
	Bertweet-base	0.56	0.66	0.62	0.63	0.48
	GPT-3	0.81	0.69	0.71	0.70	0.45
	Proposed ensemble model	0.85	0.79	0.78	0.78	0.52
Google Translate	Twitter-roberta-base	0.62	0.70	0.65	0.67	0.57
	Bertweet-base	0.63	0.71	0.67	0.69	0.58
	GPT-3	0.86	0.71	0.74	0.72	0.47
	Proposed ensemble model	0.87	0.81	0.81	0.81	0.57

Table 3: LLM Performances in a Sentiment Analysis Task for Multiple Languages (Miah et al., 2024)

In the research where the data was collected from the sources in 4 different languages and translated into English, an ensemble method was used by applying voting method through 3 LLMs in Table 3 (BERT, RoBERTa, and GPT-3). The data was labelled by the label which was found as majority from the outputs of 3 of the models. The F1 score for ensemble method hold the highest ratio in both machine translations 78.31% and 81.06% for Libre Translate and Google Translate respectively. GPT-3 was the second among them with the percentages of 69.93% and 71.99%.

4.2. Market Prediction with LSTM

In a 2018 study (*Predicting the Price of Bitcoin Using Machine Learning*, 2018), models like LSTM, RNN, and ARIMA were used to predict financial trends in the cryptocurrency market. Among them, LSTM emerged as the most effective due to its ability to capture long-term patterns in time-series data, which significantly improves financial forecasting. In contrast, ARIMA models, while straightforward and precise in certain scenarios, often fall short when handling the complexities of financial markets, making them less reliable for predictions. This study highlighted LSTM's potential in financial predictions.

Similarly, in another 2018 paper titled *A New Forecasting Framework for Bitcoin Price with LSTM*, LSTM models demonstrated strong potential in predicting cryptocurrency prices, outperforming traditional models like ARIMA, which often struggle with market unpredictability. While these studies underscore LSTM's advantages in financial forecasting.

Recent research by Maheshwari et al. (2024) also emphasized LSTM's superiority in capturing complex patterns in volatile markets like Bitcoin, offering an edge over traditional models. However, ongoing improvements are needed to fully harness its predictive power in financial forecasting.

In a recent research (Wang et al., 2024), the power of LSTM was referred through the gradient vanishing issue, when processing large datasets. LSTM can overcome this challenge, while traditional RNNs cannot.

5. Possible Problems and Limitations

5.1. Complex Structure of the Language

Even though there are many state-of-the-art fine-tuned models which can be used in understanding the sentiment of a sentence, the data on the internet is not similar to the training data generally. Because of that, some points might not be clear for the model. While there are experts who explain their predictions about the trend, still there are more people who share their ideas about cryptocurrencies in an informal way. However, the data used to train the language models contain well-structured sentences with formal expressions. Also, it's expected to see some different structures on any social media platforms such as idiom, negation, sarcasm and irony. These challenges make labelling in a correct way hard for a language model. The key method can be to use LLMs which detect irony and sarcasms (Sharma et al., 2024).

5.2. non-Text Data

The posts shared on social media don't always contain text. They sometimes can contain images and/or audio such as graphs, images, videos, and voice records. Due to this reason, comprehending the whole size of the iceberg might not be possible from time to time. For example, there are many meme subreddits on Subreddit, which means that none of the language models can understand what they mean and they miss the part under the water of the iceberg. In this work, the images won't be taken into account, but it can be examined in a further research, and it can even make the model more solid (Sharma et al., 2024).

5.3. Various Languages

In today's world, social media platforms are used by millions and even billions of users across the world. Besides that, cryptocurrencies are held by numerous people in different parts of the world. Sometimes, governments can set new rules which can affect the price of any cryptocurrencies. To understand what is happening in the world, thoughts in all languages must be checked to keep up with the recent happenings. In spite of the fact that many people speak English, but the number of people who cannot speak English shouldn't be underestimated and they prefer to share posts in their own languages. Besides that, some breaking news may appear on local news websites way before global ones. Unless people build up a model that can understand the sentiment in each language, the models which predict the trend of cryptocurrencies don't give precise results. To use miltilingual sentiment analysis LLM can be solution for this problem.

5.4. Exist LLMs' Abilities

Some of the latest language models are known as fascinating tools which can solve almost every problem, but they can have lack of ability to figure out when it comes to domain-specific subjects. All of the capabilities of a language model is related to the data used to train. If it faces new data, which is completely irrelevant to training data, it might not give an accurate result. Because the fine-tuned ones generally are trained by some datasets (not specifically crypto terminologies), the reliability of the sentiment found from the data might not be correct. As a solution, to use different approaches by finding the sentiment by taking multiple LLM's result into account in a combination way can be helpful to overcome this problem.

5.5. Fake or Scam Posts

Despite having some benefits of sharing fake news such as warning people about possible risks by trying to change the way how they act, it can lead to serious consequences in terms of the cryptocurrency market. One of the examples of this fact is that people can lose their assets due to a piece of wrong information which comes up on social media (Husbands, Perach, & Buchanan, 2024).

In today's world where the majority of people use social media, the intention of each person is not the same as each other. Fake data can come up among the data collected from social media. It can change the result of the analysis (Sharma et al., 2024).

5.6. About Datasets Used to Train LLMs

There are specific number of datasets that every model is fine-tuned with. It might not resulted in a good model accuracy because of the application type. The usage area of the model can be related to finance. In that case, some words can mean something meaningless for a domain, while it can empasize something really important to understand the sentiment of a sentence. To get the best homogeinity, to use many external dataset and combine them together in a different way can be the key which solves the problem (Mishev et al., 2020).

5.7. About LSTM

LSTM models are powerful as for time-series forecasting, but there are some challenges which should be taken into account such as high computational demands and the risk of overfitting, especially while working with smaller datasets. Effective hyperparameter tuning is crucial to optimize performance and prevent these issues, highlighting the need to balance model complexity with efficiency (Greff et al., 2017).

6. Methodology

6.1. Data

6.1.1. External Datasets for Sentiment Analysis

Using an external dataset was must to figure out the each LLM performance in terms of sentiment analysis. Totally 6 financial related and 5 non-financial datasets were used as can be seen in Table 4.

Category	Dataset Name	Explanation
Financial Datasets	SemEval 2017 - Task5 Subtask1	Part of the SemEval 2017 competition, specifically for Task 5 Subtask 1, which focused on fine-grained sentiment analysis on financial news.
	Forex News	News articles related to the foreign exchange (Forex) market.
	Twitter Financial News	Tweets related to financial news
	Yahoo Finance News	News articles and headlines from Yahoo Finance
	FIQA + Financial Phrasebank	FIQA stands for Financial Opinion QA, and Financial Phrasebank is a collection of common phrases used in financial news
non-Financial Datasets	Stanford Sentiment Treebank (SST)	A well-known dataset for sentiment analysis developed by Stanford
	IMDB	Movie reviews from the Internet Movie Database (IMDB)
	Twitter Samples	A collection of tweets for sentiment analysis
	Sentiment Polarity v1.0	Created for sentiment analysis research
	Sentiment Polarity v2.0	Updated version of the Sentiment Polarity dataset

Table 4: The External Datasets Used in the Research

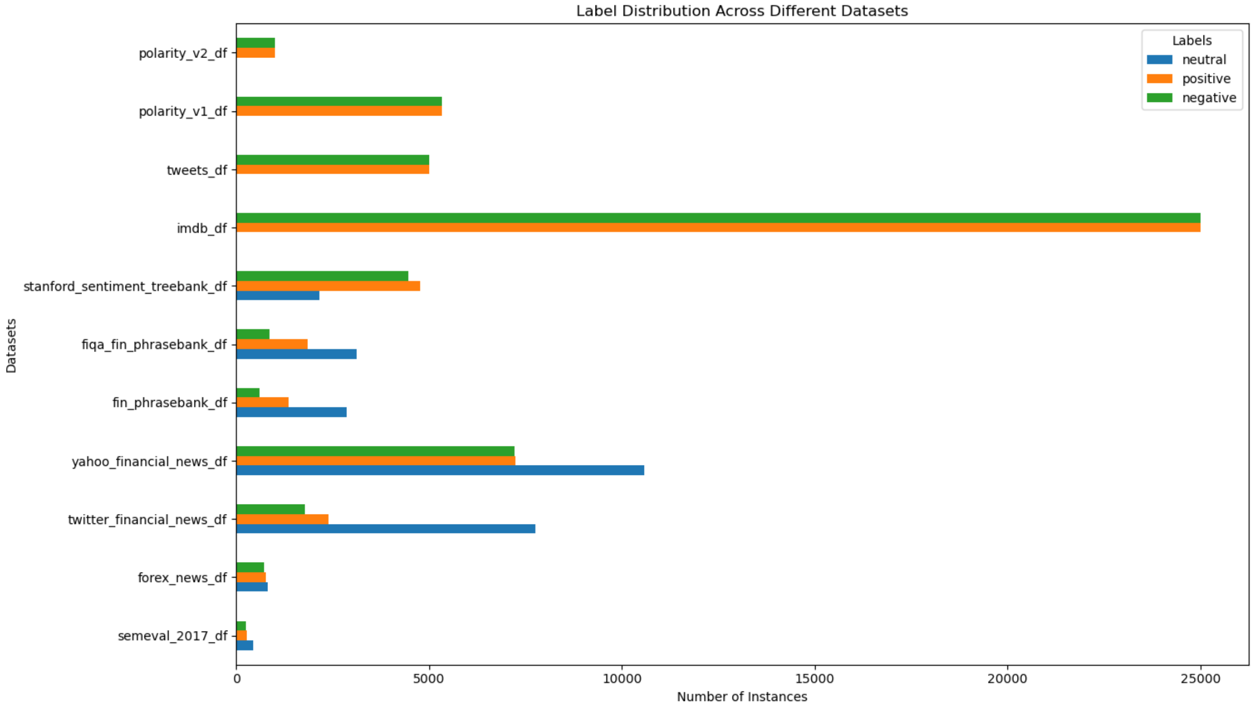


Figure 5: Label Distribution across Different Datasets

The distribution of the labels for each dataset can be seen in Figure 5. IMDB datasets hold the highest amount of data instances, while Semeval 2017 was the lowest one.

The financial dataset was created by using only datasets related to finance domain. The mixed dataset was created by using half from financial and half from non-financial datasets. While creating the datasets, the labels which contained the minimum number of instances were taken into account in order to save the homogeneity. The number of each label in each dataset can be seen in Table 5 below.

	negative	neutral	positive	total
financial	9618	9618	9618	28854
mixed	4304	4304	4304	12912

Table 5: Created Datasets and Label Counts

6.1.2. Social Media Data

Reddit was used in this project as a social media. Reddit is a popular social media and community platform where users can share content. Reddit hosts a wide variety of user-generated content organized into thematic categories called "subreddits," covering nearly every topic, from news and entertainment to special interests and hobbies. Because of that, it is known as the front page of the internet. Users can post text, links, images, and videos, while other members can comment, making it a go-to source for social interaction.

At first, the crypto-relevant subreddits with the number of subscribers were searched by using different keywords through the API, and saved as unique values to prevent the repetition.

keywords: **crypto, cryptocurrency, cryptocurrencies, crypto+currency, crypto+currencies, bitcoin, btc, ethereum, eth, tether, binance, bnb, solana, xrp, dogecoin, cardano, shiba, tron**

After that, the ones which had less than specific number of subscribers were excluded from the subreddit list and saved as dataframe. Afterwards, each subreddit was checked one by one to make sure that each of them is related to cryptocurrency, and the irrelevant ones were removed from the list. At the end, there were 331 subreddits, but some of them were banned and the data could be collected only short period of time from banned ones.

Since some of the subreddits have millions of users, it would take much time to get all the comments and replies for them. Hence, a couple of sequential days were watched, and the top 10 subreddits were used as main subreddits to collect the comment and reply data, while all post title and content were collected from all.

Only some specific data were collected from the Reddit API such as subreddit name, post text, post title, comments, replies, and the time & date when they were shared.

The data were taken in json format, and the date & time was gathered as UTC time format. A function was created to convert UTC into timestamp according to London time zone.

Additionally, to secure the ethical concideration in the project, the text data wasn't saved as it was. Each text was changed by using a private method and saved in seperated dataframes for posts and comments.

On the other hand, there were some texts which consisted non-English characters. These characters were removed from the data by a simple function. The texts were not preprocessed since the LLMs have ability to deal with the texts.

6.1.3. Cryptocurrency Market Data

The prices of the cryptocurrencies were taken from CoinGecko through the API. CoinGecko, one of the best cryptocurrency data platforms, offers comprehensive and reliable information on various digital assets. It provides real-time price tracking, market cap, trading volumes and other important indicators for thousands of cryptocurrencies. Users can track market movements through the comprehensive historical data, charts and research tools available to them.

From the API, the data of the top 12 most popular cryptocurrencies were taken such as cryptocurrency name, date, time, price, market cap, and total volume.

The most popular 12 cryptocurrencies: **“bitcoin”, “ethereum”, “tether”, “bnb”, “solana”, “usdc”, “xrp”, “lido staked ether”, “dogecoin”, “toncoin”, “cardano”, “tron”.**

Market cap is the total market value of a cryptocurrency, calculated as the current price multiplied by the total supply of coins in circulation.

The total volume indicates the trading volume in a specific period of time.

The free API let the users take the detailed data for the last 3 months. In the detailed data, almost every 1 hour’s data can be seen.

6.2. Sentiment Analysis

As for labelling, 7 different pre-trained LLMs, VADER, and TextBlob were tested on external datasets, financial and mixed seperately. The accuracy was considered as a success metric for sentiment analysis. Then, the best models were compared by precision, recall, and F1 score.

Accuracy is the simplest measurement among the all. It is basically calculated dividing the correct prediction by the total number.

$$Accuracy = \frac{The\ number\ of\ correct\ prediction}{The\ total\ number}$$

Precision shows that the ratio that how the model is good in positive prediction among all the positive predicted data.

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$$

Recall is an indicator which shows that how the model can correctly predict the positive labels among all the actual positive ones.

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}$$

F1 Score is the harmonic mean of both precision and recall.

$$F1\ Score = 2\ x \frac{Precision\ x\ Recall}{Precision + Recall}$$

6.2.1. Models Used for Sentiment Analysis and Results

All models were found on Huggingface, an open-source website that provides not only models and tools, but also datasets for various fields of AI such as NLP, and machine learning. Additionally, since there are many transformers libraries, they can be imported and trained on a specific datasets easily. On the other hand, the models can be used through Huggingface API.

6.2.1.1. Finding the Best Threshold for Twitter RoBERTa

Since the pretrained model's output is represented by the number of stars from 1 to 5 for Twitter RoBERTa, different thresholds were chosen to specify negative, neutral, and positive labels. By 2 different ways were approached.

I) "3 stars" was accepted as neutral
II) "2-3-4 stars" were accepted as neutral

condition	accuracy_financial	accuracy_mixed
3-star: neutral	55.48	60.54
2,3,4-star: neutral	15.89	41.26

Table 6: Twitter RoBERTa Accuracies in 2 Datasets by choosing different thresholds

The first approach outperformed (Table 6). After this point, only "3 star" was accepted as "neutral" label.

6.2.1.2. Finding the Best Threshold for TextBLOB

TextBlob gives output between -1 and 1 as a sentiment score for a sentence. Because of that to find the best threshold is important to get the best accuracy.

Threshold	Financial_Accuracy	Mixed_Accuracy
0.0001	43.754765	49.666976
0.001	43.737437	49.612763
0.01	43.505233	49.210037
0.05	42.278367	47.862454
0.1	40.205864	46.050186
0.15	38.753726	43.620684
0.2	37.201081	41.666667
0.25	35.842518	39.52912
0.3	35.014209	38.003408
0.35	34.580994	37.166976
0.4	34.359198	36.586121
0.45	34.061135	35.989777
0.5	34.03341	35.494114
0.55	33.981424	35.277261
0.6	33.534345	34.967472
0.65	33.513551	34.719641
0.75	33.399182	33.495973
0.8	33.378388	33.488228

Table 7: Text BLOB Accuracy by Different Threshold Values

Different threshold values were applied to get the best accuracy. The smaller threshold value, the higher accuracy obtained, but there was a small difference after a point as it can be seen in Table 7. However, to accept the threshold as 0.0001 would't give a precise result in each application.

6.2.1.3. All Models in Sentiment Analysis

Model	Description
ProsusAI (FinBERT)	a pre-trained model by a company named prosus (ProsusAI).
Sigma (FinBERT)	a fine-tuned version of a FinBERT model by financial phrasebank dataset. It was pre-trained by TRC2-financial, Bloomberg News, Corporate Reports, and Earning Calls by Ahmed Rachid Hazourli (Hazourli, 2022).
yiyanghkust (FinBERT)	a pre-trained model by Corporate Reports, Earning Call Transcripts, Analyst Reports (Huang, Wang, & Yang, 2022)
distilbert-base-uncased-finetuned-sst-2-english	a fine-tuned DistilBERT-base-uncased model by Stanford Sentiment Treebank (Sanh, Debut, Chaumond, & Wolf, 2019).
nlptown (BERT-base-multilingual-uncased)	a bert-based model trained by 6 languages: English, Dutch, German, French, Spanish, and Italian by Yves Peirsman (Peirsman).
siebert (RoBERTa Large)	a fine-tuned based on RoBERTa-large model by 15 datasets such as reviews (Hartmann et al. 2023).
cardiffnlp (Twitter RoBERTa)	a roBERTa-base model trained by ~58M tweets (Barbieri et al., 2020).
blob	a Python library used for preprocessing text data and sentiment analysis. It is open source tool licensed by MIT.
vader	a tool used for sentiment analysis especially on social media data, under MIT license (Hutto & Gilbert, 2015).

Table 8: LLM Accuracies on both Datasets with Explanations

6.2.1.4. Combined LLM Models

Since LLMs don't perfectly perform yet, to use combined model approach could be better. In the combined model, 3 best models were used and each model's output was calculated by using each accuracy through different approaches. Since each model's output is different from each other. Different functions were created to obtain the same outputs as "negative", "neutral", and "positive". The reason why 3 model was chosen specifically is because of being and odd number and having more than 50% accuracy in any of financial and mixed datasets. If an even number had been chosen, there could be the case where the number of labelled sentiments would be equal, and a combined sentiment wouldn't have been able to assign.

Totall 3 different approaches were used as follows:

6.2.1.4.1. Combined-1: Accumulated Weight

The label found from each model was assigned by model accuracy and summed together.

Definitions

Let $S = \{s_1, s_2, \ldots, s_n\}$

Let $M = \{m_1, m_2, \ldots\ , m_k$

Each model $m_i$has an associated accuracy A_i

Each model m_i produces a sentiment prediction for each sentences_j, resulting in a sentiment labelL_{ij}.

Sentiment Labels

$$Sentiments = \{negative, neutral, positive\}$$

Sentiment Scores

For each sentences_j, sentiment scores are calculated based on the models' predictions.

Let $Score(s_j, sentiment)$ be the score for a given sentiment for sentence s_j.

Calculation of Sentiment Scores

For each sentences_j, the sentiment scores at the beginning are assigned as 0:

$$Score(s_j, negative) = 0$$

$$Score(s_j, neutral) = 0$$

$$Score(s_j, positive) = 0$$

For each model $m_i$with accuracyA_i:

Let L_{ij} be the sentiment predicted by model $m_i$for sentence s_j

Update the sentiment score based on the model's prediction and its accuracy:

$$Score(s_j, L_{ij}) = Score(s_j, L_{ij}) + A_i$$

Final Sentiment Decision

For each sentences_j:

The final sentimen $F(s_j)$ is specified by the sentiment with the highest score:

$$F(s_j) = \arg max_{sentiment \in \{negative, neutral, positive\}} Score\ (s_j, sentiment)$$

Example

Consider $n = 2$ (two sentences) and $k = 3$ (three models)

No.	Model Accuracy	Prediction1	Prediction2
1	68.58	positive	neutral
2	47.43	neutral	neutral
3	58.00	negative	positive

Table 9: Example Parameters for Combined-1: Accumulated Weight

Sentence s_1:

$$Scores_1 = \{negative{:}\ 58.00\ , neutral{:}\ 47.43, positive{:}\ 68.58\}$$

$$argmax_{scores1} = positive$$

$$Label_1 = positive$$

Sentence s_2:

$$Scores_2 = \{negative{:}\ 0\ , neutral{:}\ 68.58 + 47.43\ , positive{:}\ 58.00\}$$

$$Scores_2 = \{negative{:}\ 0\ , neutral{:}\ 116.01, positive{:}\ 58.00\}$$

$$argmax_{scores2} = neutral$$

$$Label_2 = neutral$$

6.2.1.4.2. Combined-2: Max Voting

Each sentiment was found from 3 models, the majority of the sentiments was chosen as the main sentiment for each sentence.

Definitions

Let S be the set of sentences: $S = \{s_1, s_2, \dots \ , s_n\}$

Let M be the set of models: $M = \{m_1, m_2, \dots \ , m_k\}$

Sentiment Labels

$$Sentiments = \{negative, neutral, positive\}$$

Calculation of Sentiment Scores:

-Initialization

$$Score(s_j, negative) = 0 \ , \qquad Score(s_j, neutral) = 0, \qquad Score(s_j, positive) = 0$$

-Update Scores:

Let L_{ij} be the sentiment predicted by model$m_i$for sentence s_j

Increase the sentiment score 1 based on the model’s prediction:

$$Score(s_j, L_{ij}) = Score(s_j, L_{ij}) + 1$$

Final Sentiment Decision

For each sentence s_j, the final sentiment $F(s_j)$ is determined by the sentiment with the highest accumulated score:

$$F(s_j) = argmax_{sentiment \in \{negative.neutral, positive\}} Score \ (s_j, sentiment)$$

Example

Consider $n = 2$ (two sentences) and $k = 3$ (three models)

No.	Model Accuracy	Prediction1	Prediction2
1	68.58	positive	neutral
2	47.43	positive	neutral
3	58.00	negative	positive

Table 10: Example Parameters for Combined-2: Max Voting

Sentence s_1:

$$Scores_1 = \ \{negative: 1 \ , neutral: 0, positive: 2\}$$

$$argmax_{scores1} = positive$$

$$Label_1 = positive$$

Sentence s_2:

$$Scores_2 = \{negative: 0\ , neutral: 2, positive: 1\}$$

$$argmax_{scores2} = neutral$$

$$Label_2 = neutral$$

6.2.1.4.3. Combined-3: Weighted Average

As for weighted average, the sentiment predictions were converted into binary representations as -1, 0, and 1 for negative, neutral, and positive sequentially. In order to be able to make this conversion possible, that the sentence is detected as negative can be neutral, since the neutral is the closest to the negative. Assume that the sentiment of a sentence is found as negative by 2 models, while another model finds it as positive. After being multiplied by the weights, it will be between negative and neutral due to the nature of weighted average.

Definitions

Let S be the set of sentences: $S = \{s_1, s_2, \dots\ , s_n\}$

Let M be the set of models: $M = \{m_1, m_2, \dots\ , m_k\}$

Let $A_i$be the accuracy of model m_i
Total accuracy: $\sum_{i=1}^{k} A_i$

Sentiment to Binary Conversion:

$$B = \begin{cases} -1\ if\ sentiment = "negative" \\ 0\ if\ sentiment = "neutral" \\ 1\ if\ sentiment = "positive" \end{cases}$$

Calculation of Weighted Sentiment Scores:

-Initialization:

$$Score(s_j, negative) = 0\ , \qquad Score(s_j, neutral) = 0, \qquad Score(s_j, positive) = 0$$

-Update Scores:

For each model $m_i$with accuracy $A_i$and predictionL_{ij}:
Find the binary score $b_{ij} = B(L_{ij})$.
Update the sentiment score for sentence $s_j$by weighting the binary score with the model's accuracy and normalizing by the total accuracy.

$$Weighted\ Score(s_j, L_{ij}) = Score(s_j, L_{ij}) + \frac{A_i x\ b_{ij}}{Total\ Accuracy}$$

Total Score:

$$Total\ Score = \sum_{sentiment\ \in \{negative, neutral, positive\}} Weighted\ Score_{sentiment}$$

Final Sentiment Decision:

$$F(s_j) = \begin{cases} negative\ if\ total\ score < -0.5 \\ positive\ if\ total\ score > 0.5 \\ neutral\ otherwise \end{cases}$$

Example

Consider $n = 2$ (two sentences) and $k = 3$ (three models)

No.	Model Accuracy	Prediction1	Prediction2
1	68.58	positive	neutral
2	47.43	positive	neutral
3	58.00	negative	positive

Table 11: Examples Parameters for Combined-3: Weighted Average

$$Binary\ Conversion = \{negative: -1, neutral: 0, positive: 1\}$$

$$Total\ Accuracy = 68.58 + 47.43 + 58.00 = 174.01$$

Sentence s_1:

$$Weighted\ Scores_1 = \{negative: \frac{-1\ x\ 58.00}{174.01}, neutral: 0, positive: \frac{1\ x\ (68.58 + 47.43)}{174.01}\}$$

$$Weighted\ Scores_1 = \{negative: -0.33, neutral: 0, positive: 0.66\}$$

$$Total\ Score_1 = -0.33 + 0 + 0.66 = 0.33$$

$$Label_1 = neutral$$

Sentence s_2:

$$Weighted\ Scores_2 = \{negative: 0, neutral: \frac{0\ x\ (68.58 + 47.43)}{174.01}, positive: \frac{1\ x\ (58.00)}{174.01}\}$$

$$Weighted\ Scores_2 = \{negative: 0, neutral: 0, positive: 0.33\}$$

$$Total\ Score_2 = 0 + 0 + 0.33 = 0.33$$

$$Label_2 = neutral$$

6.2.1.5. sBERT

The idea that "similar sentences are found in similar contexts" is fundamentally associated with the principles of word embedding techniques (Mikolov et al., 2013).

Therefore, the sentences which contain negative sentiment might have a kind of relationship between other negative ones because of containig words which has negative meaning, so the other sentiments do.

By using SBERT and 2 different similarity measures (cosine, and manhattan similarity), each sentence's similarity between other sentences from each label was calculated. The average similarity for each label was found, and the maximum one was accepted as the label of the sentence. This calculation was performed over 10 different datasets.

6.2.1.6. Evaluation of Sentiment Distribution

Skewness measures the symmetry of a distribution. Normal distribution has a perfectly symmetric curves on both sides. It is desired to have 0 skewness value to get the symmetric data distribution.

$$\begin{cases} If\ Skewness > 0: & right - skewed\ distribution \\ If\ Skewness < 0: & left - skewed\ distribution \\ If\ Skewness = 0: & normal\ distribution \end{cases}$$

Kurtosis measures the tailedness of a distribution. When a distribution has light tails, it has fewer outliers than usual, while it is opposite for long tails. Hence, the density of outliers can be understandable. In normal distribution, there might be few of outliers. It is desired to have a value of 3 to have a normal distributed data (NIST/SEMATECH, 2012).

$$\begin{cases} If\ Kurtosis > 0: & long\ tails \\ If\ Kurtosis < 0: & light\ tails \\ If\ Kurtosis = 3: & normal\ distribution \end{cases}$$

6.3. Time-Series Analysis with LSTM Models

6.3.1. Data Preparation for LSTM

The data was prepared for LSTM modeling by first converting sentiment information into numerical values to ensure compatibility with the model. The data was then resampled into hourly and daily intervals to maintain a consistent timeline, and cryptocurrency price data was integrated to form a comprehensive dataset that included both sentiment scores and corresponding prices for each period of time specified. To make the data suitable for modeling, normalization was applied, and a target variable representing the cryptocurrency price at the next time step was created. These steps ensured that the data was clean, well-organized, and ready for effective use in LSTM-based predictions. Also, some parameters were provided to find the best value for each as they follow:

- **period:** the time intervals were specified as either hours or days (1H, 2H, … 1D, 2D…)
- **shifted_feature_number:** the price was shifted into new columns. If 3 is provided, 3 previous average consequetive prices of the cryptocurrencies were added into 3 new columns by shifting. In other words, 3 previous prices were taken as an input to predict the 4th price in each row.
- **n_later_predict:** n-period later cryptocurrency price which was preferred to be predicted. For example, as for period is equal to 2H and n_later_predict is equal to 3, data is arranged by each 2-hour period and the prediction starts froom 3 periods further, which means 6H for these parameters.
- **sentiment_parameter:** it helps to specify which sentiment was taken into account, or all sentiments were excluded to see the effect of the sentiment in the model (**with:** with sentiment, **without:** without sentiment, **body:** body sentiment (comment & replies), **body_market_cap:** body sentiment & market capitalization)

6.3.2. LSTM Model Architecture

To explore the impact of sentiment data on the accuracy of cryptocurrency price predictions, different parameters were tried on different hyperparameters and evaluated.

Input Layer: Takes in sequences of data, which can include historical prices and sentiment scores depending on the model configuration.

LSTM Layers:

- **First LSTM Layer:** Configured to capture and remember sequences over time, building an understanding of temporal patterns.
- **Dropout Layer:** Follows the first LSTM layer to reduce overfitting by randomly deactivating some units during training.
- **Second LSTM Layer:** Further processes the data by focusing on long-term dependencies within the time series.
- **Dropout Layer:** Another dropout layer follows to ensure the model generalizes well to new data. Therefore, overfitting is prevented.

Dense Layer: Condenses the information extracted by the LSTM layers into a more refined and focused representation.

Output Layer: Produces a single value, which is the predicted cryptocurrency price for the next time step.

The model was compiled with a loss function that measures the difference between predicted and actual prices, and an optimizer that helps the model learn efficiently from the data.

On the other hand, 2 LSTM layers were used so that the model can adapt more complex data structure like cryptocurrency market with sentiments.

6.3.3. Training and Validation of LSTM Models

At first, the effect of the sentiment and the most correlated columns were examined. The data was split into training and validation sets, with the rates of 80% and 20% respectively.The following stable hyperparameters were used across all models until finding the best data preparation parameters:

- **First LSTM Layer Units:** 50
- **Second LSTM Layer Units:** 50
- **Dense Layer Units:** 50
- **Dropout Rate (First Layer):** 0.2
- **Dropout Rate (Second Layer):** 0.2
- **Optimizer:** Adam
- **Number of Epochs:** 100
- **Batch Size:** 32
- **period:** 1H, 6H, 1D
- **shifted_feature_number:** 0
- **n_later_predict:** 1

6.3.4. Evaluation Metrics

After training, the models were evaluated using several performance metrics:

Mean Absolute Error (MAE): The average magnitude of errors in the predictions, providing a direct measure of prediction accuracy. Smaller is better.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Mean Squared Error (MSE): The variance of prediction errors, emphasizing larger errors due to the squaring process. Smaller is better.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE): The square root of MSE. Smaller is better.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

R² Score: How well the model can explain the variance in the data. It gets value between 0 and 1. A value close to 1 is better.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Mean Absolute Percentage Error (MAPE): Expressed prediction accuracy as a percentage, facilitating easy comparison across different models. Smaller is better.

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

In the models, mean absolute error value was taken into account mainly, but other ones were checked to be sure about the preciese of the metric.

6.3.4.1. With vs. Without Sentiment

With Sentiment: These models included sentiment features alongside other financial indicators like price, market cap, and total volume.

Without Sentiment: These models excluded sentiment features, relying solely on traditional financial indicators.

6.3.4.2. Body Sentiment vs. Without Body Sentiment

Only Body Sentiment: Only body sentiment column was included, while others were removed from the dataframe.

Without Sentiment: All sentiment-related columns were excluded from the dataframe.

6.3.4.3. Most Correlated Columns vs. Others

The most correlated columns which were found in the data analysis part were considered as the most correlated columns. Other ones were accepted as not-correlated ones. After finding the best columns, the best parameters for data preprocessing part according to mean absolute error mainly were found.

6.3.5. Finding the Best Model

After finding the best parameters for the data preprocessing, different parameters were tried as hyperparameters. The optimization process is carried out using the fmin function from the Hyperopt library, which applies the Tree of Parzen Estimators (TPE) algorithm to search for the optimal set of hyperparameters as below.

- **First LSTM Layer Units:** 50, 100, 150
- **Second LSTM Layer Units:** 50, 100, 150
- **Dense Layer Units:** 50, 100, 150
- **Dropout Rate (First Layer):** 0.1 to 0.5 (uniform distribution)
- **Dropout Rate (Second Layer):** 0.1 to 0.5 (uniform distribution)
- **Optimizer:** Adam
- **Number of Epochs:** 50, 100, 150
- **Batch Size:** 16, 32, 64

6.4. Overall Workflow of Sentiment Analysis and Price Prediction

The following flowchart (Figure 6) illustrates the comprehensive workflow used in this study, from the preprocessing of social media data for sentiment analysis to the final price prediction using LSTM models. Each step in the process is interconnected, demonstrating how sentiment data is combined with cryptocurrency market data to produce accurate price predictions

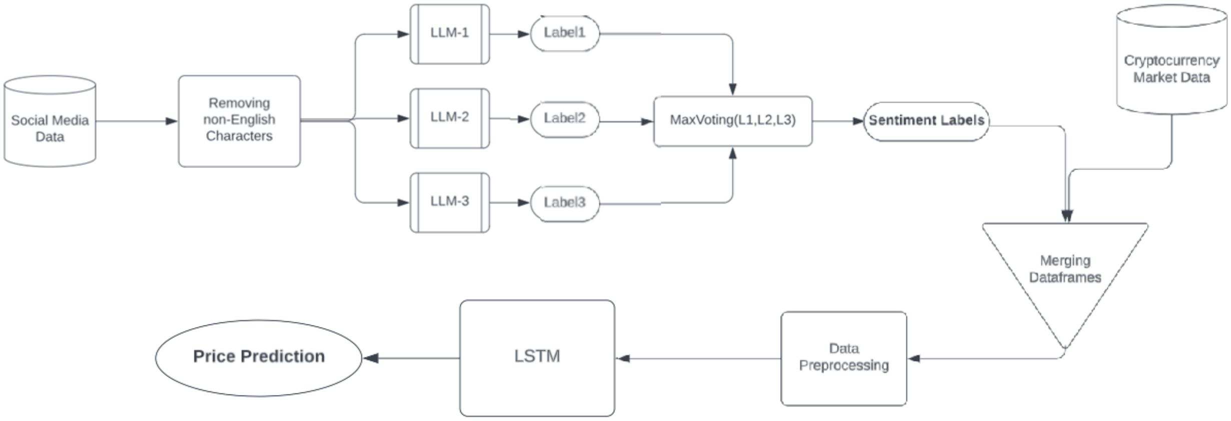


Figure 6: Overall Flowchart

7. Experimental Results

7.1. Tools and Libraries Used

Python was used as a programming languages. Python is a high-level object oriented programming language. The syntax of it makes the learning and implementation easier compared to other programming languages. Since the numerous of users, the excessive number of libraries for almost every application can be found on many platforms. Additionally, the libraries in Table 12 was used (Van Rossum & Drake, 2012).

Library	**Explanation**
pandas	Data analysis and manupilation
numpy	Computation in mathematics by matrices, arrays
requests	HTTP library to get the data from API
time	Helps to add delay
re	Text manipulation
tqdm	Progress bar
datetime	Converting UTC to datetime
pytz	Timezone calculations
seaborn	Visualization
matplotlib	Visualization
torch	GPU computation
pipeline	Combining multiple process into a single one
transformers	Using LLMs from Huggingface
os	Interaction with files on computer
sklearn	Machine learning library, automatically calculates the model parameters such as accuracy, precision, recall, and f1 score.
autotokenizer	Automatically selects the tokenizer
modelForSequenceClassification	Sequence classification tasks
tensorflow	Machine learning framework developed by Google
hyperopt	Hyperparameter optimization

Table 12: Libraries Used in the Research

7.2. Sentiment Analysis Results

7.2.1. Model Accuracies on Both Datasets

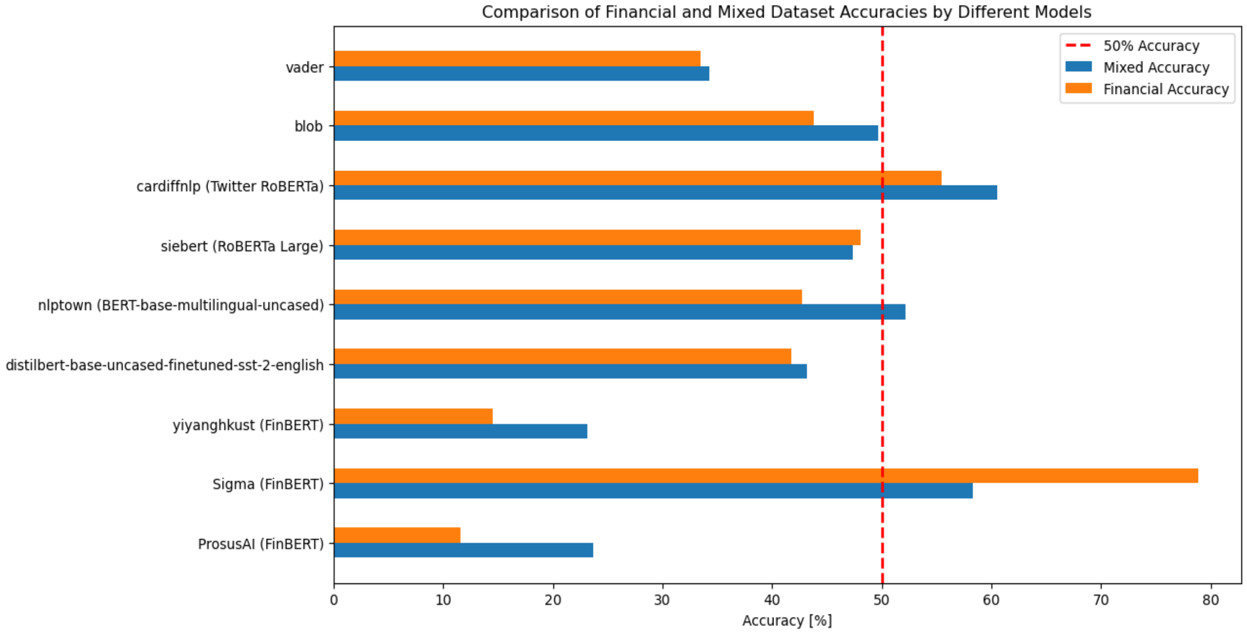


Figure 7: Comparision of Different Models in Sentiment Analysis over 2 Datasets with a Fifty-percent Accuracy Line

model	financial accuracy	mixed accuracy	average accuracy
ProsusAI (FinBERT)	11.57	23.66	17.61
Sigma (FinBERT)	**78.83**	**58.34**	**68.58**
yiyanghkust (FinBERT)	14.51	23.10	18.81
distilbert-base-uncased-finetuned-sst-2-english	41.71	43.18	42.44
nlptown (BERT-base-multilingual-uncased)	42.75	**52.12**	47.43
siebert (RoBERTa Large)	48.02	47.39	47.71
cardiffnlp (Twitter RoBERTa)	**55.48**	**60.54**	**58.01**
blob	43.75	49.67	46.71
vader	33.49	34.25	33.87

Table 13: LLM Accuracies on both Datasets with Explanations

Each model's accuracy can be seen in Figure 7 and Table 13Table 8 for both dataset. Since all models were not good, only the models which hold more than 50% accuracy in either dataset were taken into account for the combined models.

7.2.2. Combined Models' Results

	financial negative	financial neutral	financial positive	financial accuracy	financial macro avg	financial weighted avg
weighted_average_precision	0.3828	0	0.8089	0.4412	0.3972	0.3972
weighted_average_recall	0.9910	0	0.333	0.4412	0.4412	0.4412
weighted_average_f1-score	0.5523	0	0.471	0.4412	0.3413	0.3413
max_voting_precision	0.8454	0.6579	0.7862	0.7512	0.7631	0.7631
max_voting_recall	0.7122	0.7934	0.7480	0.7512	0.7512	0.7512
max_voting_f1-score	**0.7731**	**0.7193**	**0.7666**	**0.7512**	**0.7530**	**0.7530**
accumulated_weight_precision	0.8392	0.651	0.8258	0.7574	0.7720	0.7720
accumulated_weight_recall	0.7408	0.8081	0.7234	0.7574	0.7574	0.7574
accumulated_weight_f1-score	**0.7869**	**0.7211**	**0.7712**	**0.7574**	**0.7598**	**0.7598**

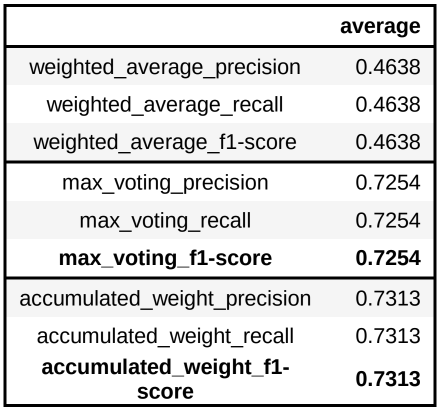
Table 14: Combined Models' Results for 3 Approaches on Financial Data

Weighted Average**,** Max Voting**,** and Accumulated Weight were tested in a homogenic financial dataset. The metrics displayed are precision, recall, and F1-score for three classes (negative, neutral, positive) as well as their averages (accuracy, macro average, and weighted average). These metrics provide insight into the effectiveness and biases of each method. Accumulated weight and max-voting accuracy gave the best results among 3 of these approaches as it can be seen in Table 14. There was only less than 1% difference between each other.

	mixed negative	mixed neutral	mixed positive	mixed accuracy	mixed macro avg	mixed weighted avg
weighted_average_precision	0.4059	0	0.8202	0.4864	0.4087	0.4087
weighted_average_recall	0.9811	0	0.4780	0.4864	0.4864	0.4864
weighted_average_f1-score	0.5742	0	0.6039	0.4864	0.3927	0.3927
max_voting_precision	0.7971	0.5962	0.7444	0.6996	0.7126	0.7126
max_voting_recall	0.6645	0.7177	0.7165	0.6996	0.6996	0.6996
max_voting_f1-score	**0.7248**	**0.6513**	**0.7302**	**0.6996**	**0.7021**	**0.7021**
accumulated_weight_precision	0.7924	0.5804	0.8104	0.7052	0.7277	0.7277
accumulated_weight_recall	0.6801	0.7572	0.6785	0.7052	0.7052	0.7052
accumulated_weight_f1-score	**0.7319**	**0.6571**	**0.7386**	**0.7052**	**0.7092**	**0.7092**

Table 15: Combined Models' Results for 3 Approaches on Mixed Data

In the mixed financial dataset, the results summarized in Table 15. Both Max Voting and Accumulated Weight demonstrated superior performance, similar to the previous dataset, with only a marginal difference of less than 1% between them.



	average
weighted_average_precision	0.4638
weighted_average_recall	0.4638
weighted_average_f1-score	0.4638
max_voting_precision	0.7254
max_voting_recall	0.7254
max_voting_f1-score	**0.7254**
accumulated_weight_precision	0.7313
accumulated_weight_recall	0.7313
accumulated_weight_f1-score	**0.7313**

Table 16: Average accuracies for combined models

By taking the average scores from both datasets, an average table was created as it can be seen in Table 16.

The Weighted Average method consistently showed the lowest scores in precision, recall, and F1-score. This suggests that while it may have some applicability, it is less effective in capturing the nuances of financial sentiment compared to the other two methods.

The Max Voting method performed significantly better, with all metrics averaging above 0.72. This method demonstrated a balanced approach, effectively identifying and categorizing financial sentiments across different datasets.

The Accumulated Weight method achieved the highest average scores across all metrics, marginally outperforming the Max Voting method. With an average precision, recall, and F1-score of approximately 0.73, this method appears to be the most reliable for sentiment analysis in financial datasets.

7.2.3. Combined Model Conclusion

As a result, by using only FinBERT model, maximum accuracy was obtained 78.83% for financial, while that of mixed was obtained by Twitter RoBERTa model with the accuracy of 60.54%.

When it comes to the combined model, the best accuracy for financial data was found around 75%, while mixed data was 70%.

Using combined model for mixed data increases the model performace, but it decreases the accuracy by around 3% for financial data.

On the other hand, since the social media data is generated by people- who mostly share their ideas by casual language- to use mixed model with either max voting or accumulated weight can be resulted in a better result.

However, to use the accumulated weight method is computational expensive, because there is a need of labelled dataset and model accuracy on each dataset.

Besides that, a model can give a wrong decision, but when it is supported by multiple models, the result which will be gathered from the models will be more precise.

Consequently, to use max voting method is easier to implement and less expensive in terms of computation and more reliable under these circumstances.

7.2.4. sBERT Sentiment Analysis Result

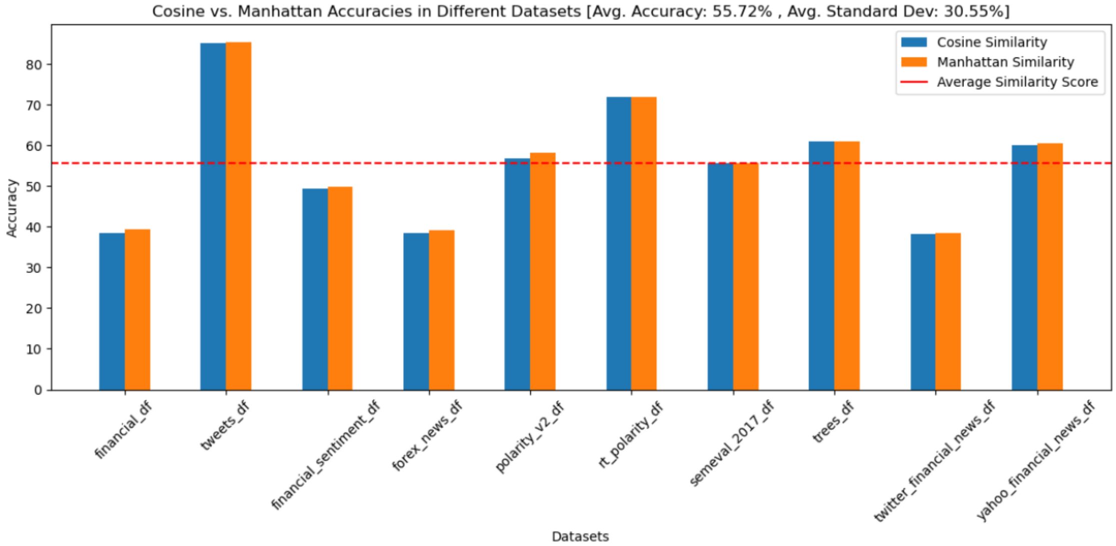


Figure 8: sBERT accuracies over different datasets

According to Figure 8, Tweets overperfomed, but the average accuracy was found as 55.72%. Additionally, average standard deviation was found as 30.55%, which meant that it cannot be accepted as a propriate model to assing the sentiment of a sentence. The reason why the best accuracy was obtained on tweets can be that the tweets were taken from similar hashtags. On the other hand, it poorly performed on the datasets relevant to finance.

As a conclusion, combined LLM was chosen as a method to assign the sentiments.

7.3. Dataset, Labels, Correlation, and Data Analysis

The dataframes for posts and comments were collected seperately and they were combined together to get a single dataframe to work.

At the end, there were 206724 data instances, where there was no null value, in the combined dataframe.

subreddit	count
Bitcoin	60790
CoinBase	13132
CryptoCurrency	49906
CryptoCurrencyClassic	124
CryptoMarkets	6930
CryptoMars	3410
CryptoMoonShots	3777
Crypto_General	2270
ethtrader	51208
memecoins	15177

Table 17: The count of data instances of each subreddit

Table 17 shows the number of instances for each subreddit. The majority of data was hold by Bitcoin, ethtrader, and CryptoCurrency. Since CryptoCurrencyClassic had only 124 data instances, it was excluded in the further analysis. After that, the total number of data decreased to 206600.

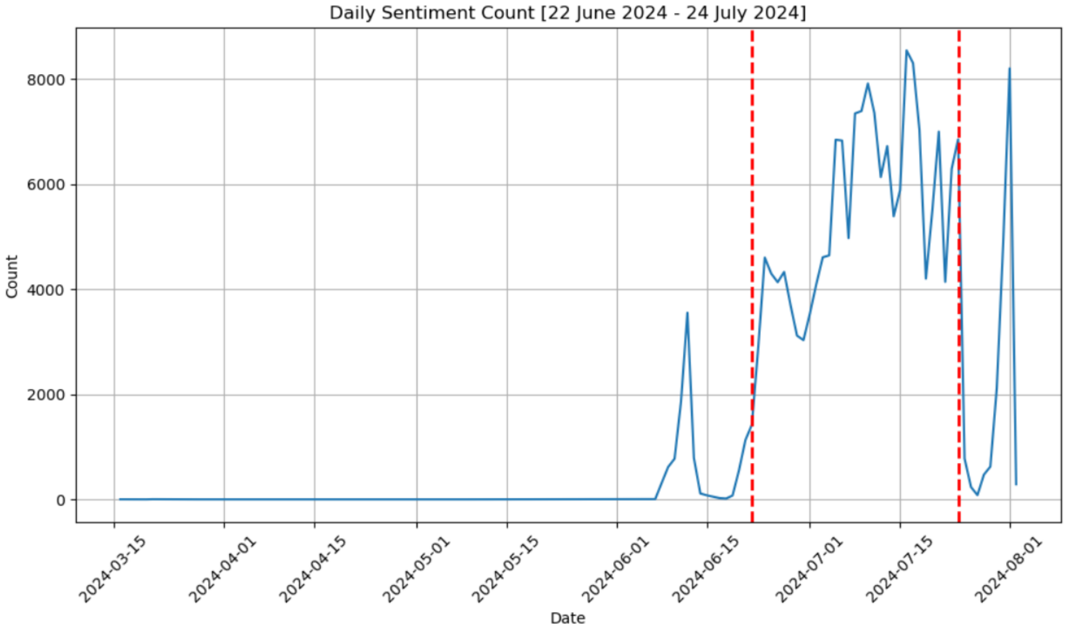


Figure 9: Daily sentiment count between the 15th of March and the 1st of August in 2024

The daily number of instances named as sentiment in Figure 9. According to the graph, it can be seen that there are less data or nearly no data for some dates due to some reasons. Because of that, nearly 1 month period, between 22 June 2024 and 24 July 2024, was taken into account in further parts. These dates were showed by red dashed lines. After focusing on these short period of time, the graph was obtained like in Figure 9, which had 178960 data instances.

After that, the max-voting method by using 3 LLMs which had the highest accuracies was used to assign the sentiment of each textual data. Each of their sentiments can be seen in Table 18.

	selftext_sentiment	title_sentiment	body_sentiment	average_sentiment
negative	38012	28422	41351	23424
neutral	124735	151875	114387	157666
positive	43853	26303	50862	25510

Table 18: Assigned sentiments for each column (**selftext:** post, **title:** post title, **body:** comments & replies, **average:** average of 3 sentiments)

Since the sentiment of post, title, and comments were different from each other for some of them, the average sentiment was calculated by using the max voting method as it was used in the part where the sentiments were found (Combined-2: Max Voting). Each row was assigned by the sentiment which repeated more than others in the row. If each of them was appeared once, it was specified as neutral.The number of neutral sentiment outraced the others, while the positive sentiment covered 12.35% of all sentiments according to average sentiment values.

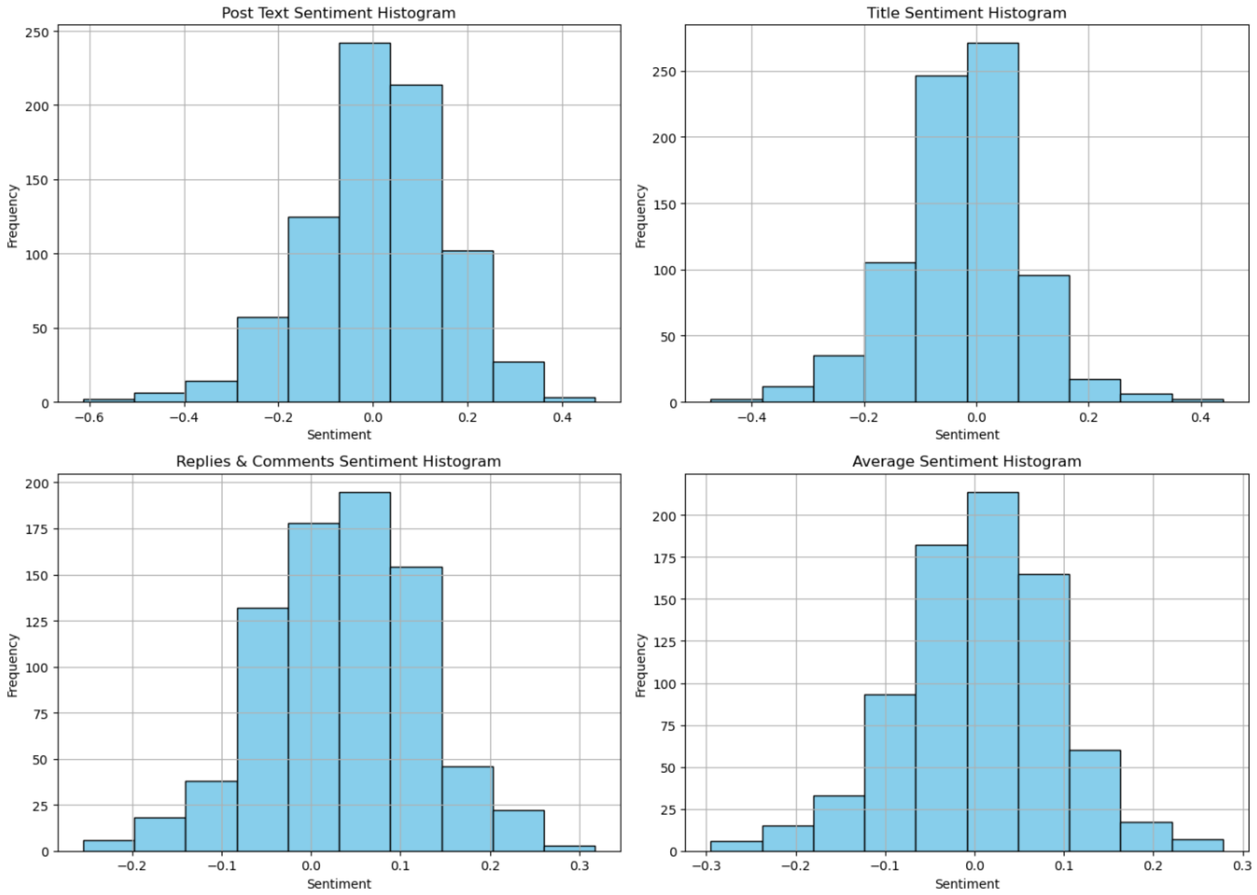


Figure 10: The distribution of each sentiment

Sentiment generally remains neutral, with just little variations into positive or negative domains, across all sentiment types in Figure 10. While replies and comments exhibit a minor positive movement, posts and titles tend to lean slightly in the direction of negativity. When the sentiment is averaged over the various categories of content, it is generally neutral with a tiny positive skew. These findings indicate that, while comments and replies tend to be slightly more positive than original posts and headlines, conversations around cryptocurrencies on this platform are typically neutral and do not strongly trend towards either extreme of positivity or negativity.

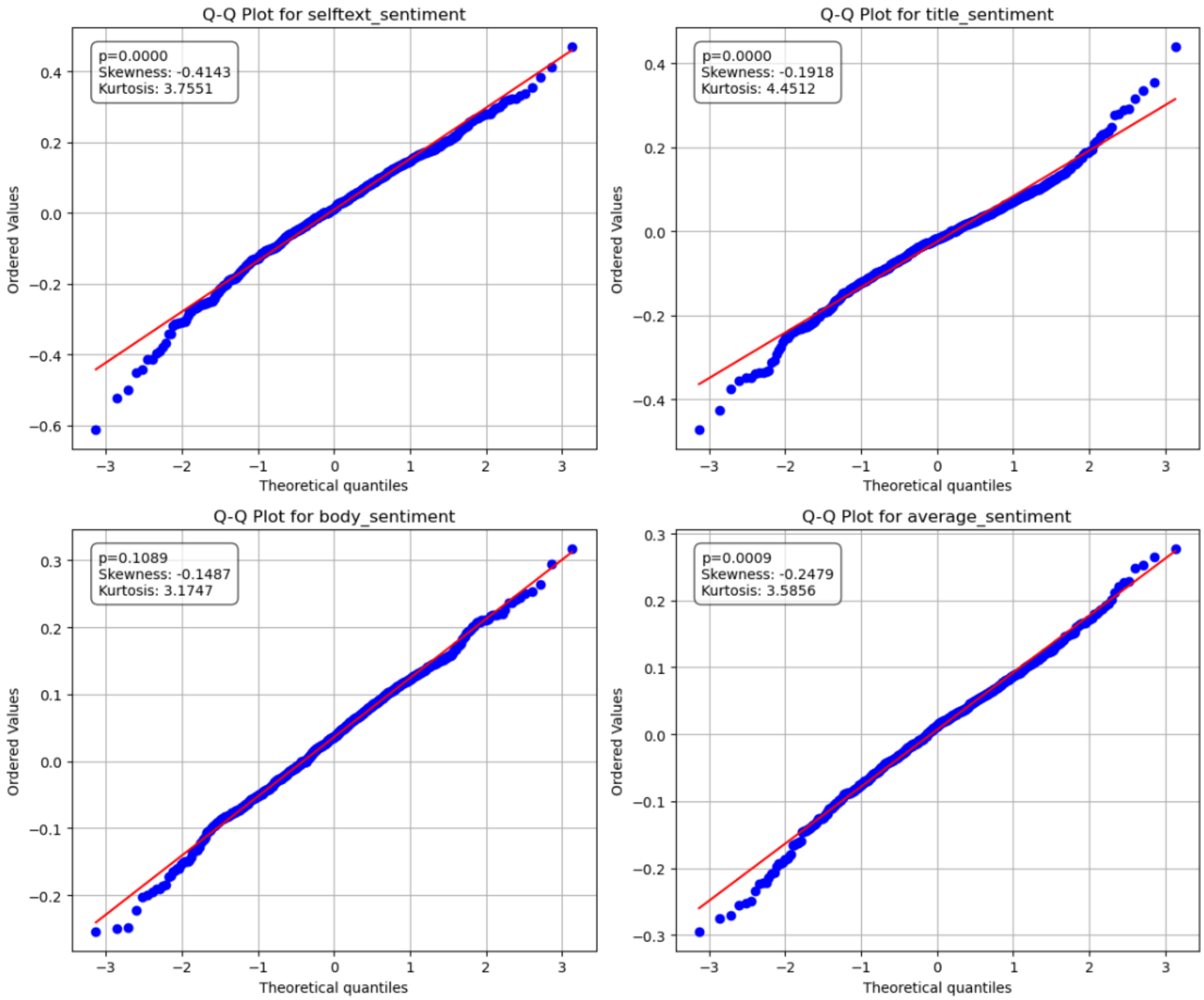


Figure 11: Q-Q plot for each sentiment type

In Figure 11, Q-Q plots can be seen. Body sentiment was the closest to a normal distribution with a non-significant Shapiro-Wilk p-value (0.1089), the lowest skewness (-0.1487), and a kurtosis value (3.1747) closest to normal. Title sentiment had the highest kurtosis (4.4512) with the longest tails. On the other hand, all distributions were left-skewed, but the highest skewed one was selftext sentiment distribution, which was related to the text shared in the post holder.

All distributions except for body sentiment showed significant deviations from normality based on the Shapiro-Wilk test (p-values < 0.05).

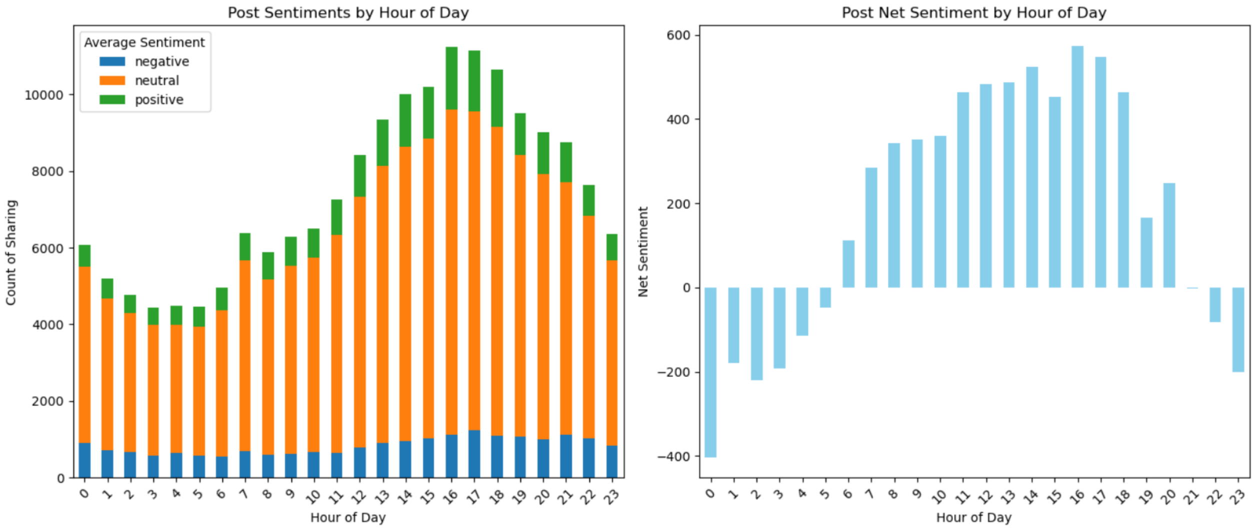


Figure 12: The count of post sentiment (left) and post net sentiment (right) across the hour of the day

In Figure 12, it can be seen that the volume of posts gradually increased from around 04:00, peaked about at 16.00 followed by a gradual decrease. As for the net sentiment, the positive sentiment surpassed between 6 AM and 8PM. There is a clear pattern where sentiment tends to be more positive during the active hours of the day and more negative during the early morning and late evening. Additionally, the time was set to GMT +0 in the analysis as it was indicated in Social Media Data. If the cryptocurrency price had had a negative trend in the dates when the data was collected, this distribution would have been dominated in the negative part, but the number of sharing still could be instensive around the active times.

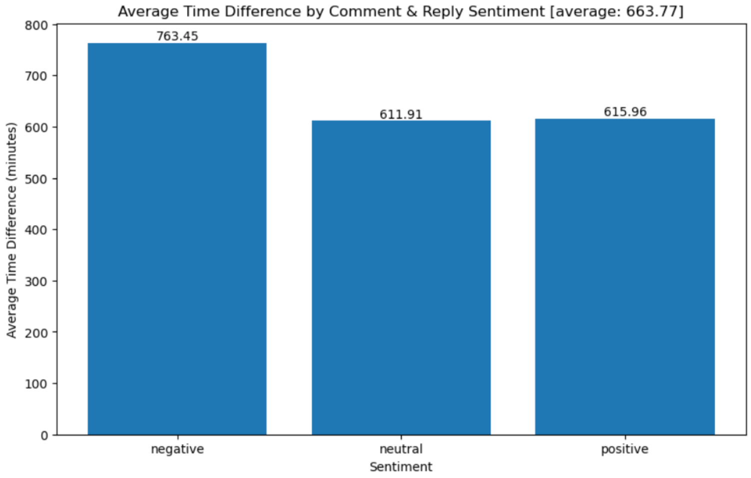


Figure 13: Average time difference

The difference between when the post is shared and that of comment was found for each one. After that, average was taken groupping by each sentiment and showed in Figure 13. According to the graph, people were tend to make comment on posts much later than the post is created, when they thought something negative compared to other sentiments. Neutral-sentimented comments took place earliest by holding nearly 4 minutes difference between positive ones (1 hour 11 minutes). The required average time to people make comment something negative was found as 12 hours and 42 minutes.

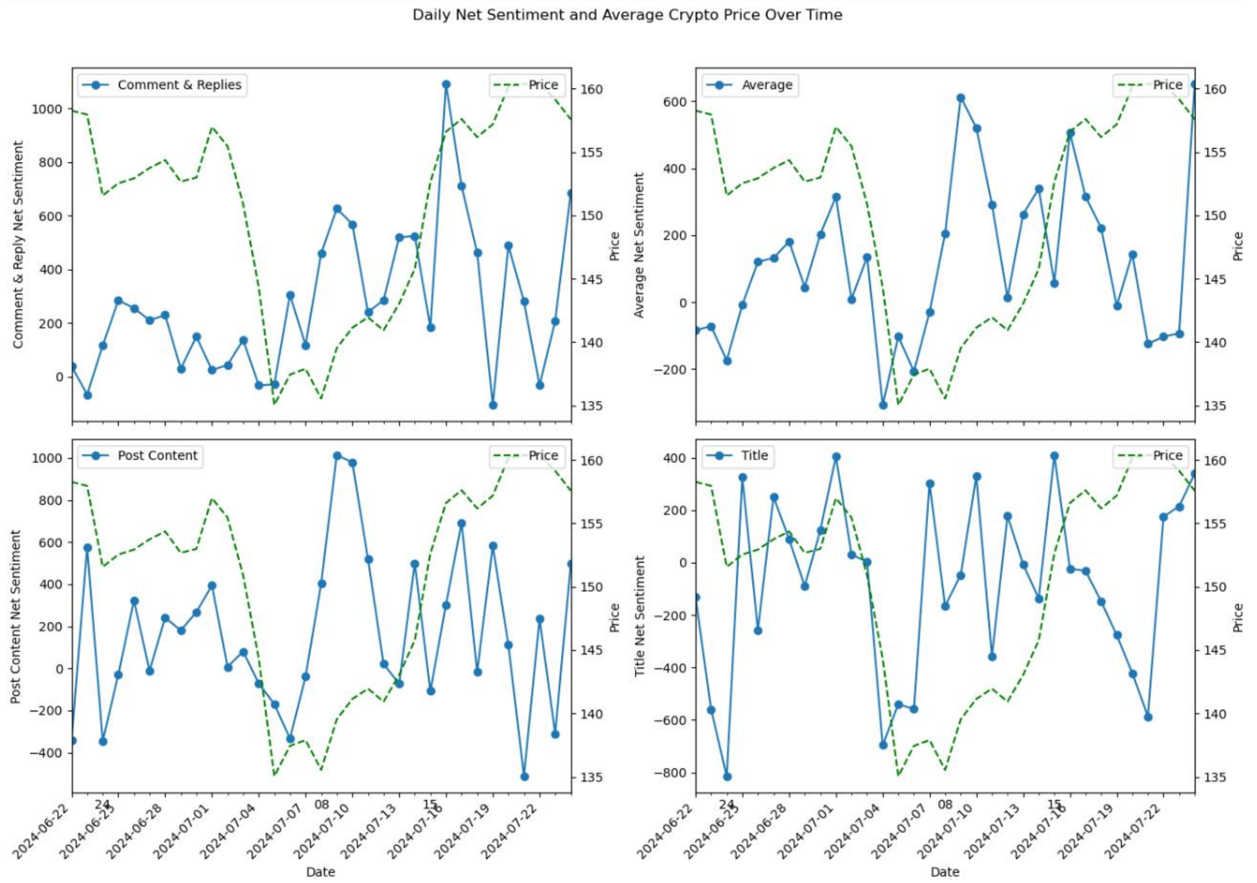


Figure 14: Daily net sentiment vs.average cryptocurrency price between 22 June 2024 and 24 July 2024

In Figure 14, the daily net aggregated sentiment with and the the total number of comments can be seen day by day. In all, the net sentiment started with negative followed by slightly positive trend until around the 4th of July. Just after then, the net sentiment jumped in the positive side suddenly. This difference was more for post content, title, and average in those dates, while it was more obvious for comment & reply on 16th of July, when the average price had already been on increase. This phenomenon might have caused that the price increased gradually from 9th to 20th of July. From this point of view, it can be told that there might be a kind of correlation between social media sentiment and the price somehow.

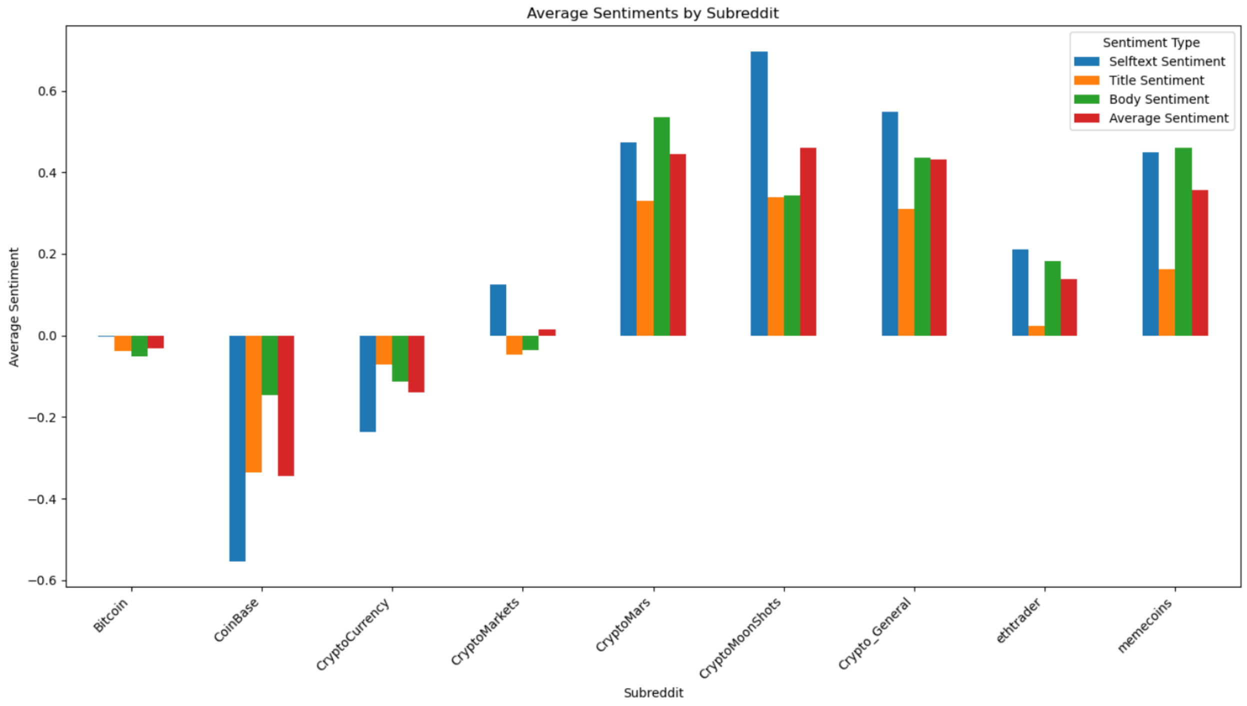


Figure 15: Subreddit average sentiment

In Figure 15, it can be seen that **CoinBase** had the most negative sentimented posts. **CryptoCurrency** was the second one. In contrast, the other subreddits, except of **Bitcoin, and CryptoMarkets**, showed generally positive sentiments, reflecting a more optimistic or bullish tone in those communities. Body (comments&replies) sentiment was the highest in **CryptoMars**. The sentiment in title was lower than selftext (text in post), even though both were shared by the same person. A question might come up at this point that how correctly each sentiment was assigned. Moreover, people may not indicate their sentiments in the title, but their intents are more obvious in the text they share.

7.3.1. Average Correlation with Price

Since the aim of the project is to uncover the key relations between sentiment on social media and cryptocurrency market, different periods of times and n period shifting of the average cryptocurrency price was taken into account. At first, the average cryptocurrency prices were found. Then, both sentiment data and average cryptocurrency price was set to specific periods of times from 1 hour to 15 days seperately. Also, the cryptocurrency average price was shift over the specified period in order to understand when the sentiments affect the price. Then, the correlation for each of them was found. For example period is equal to 1D (1 day) and n is equal to 3, both the average sentiment and cryptocurrency prices were found per day and the cryptocurrency price data was shifted 3 periods to see the effect of sentiments after 3 days. Finally, the correlation between the price and other columns seperately was calculated.

Besides that, to find the average optimum day for hourly periods, the period was multiplied by n to find how many hours difference they had between each other. Then, the value was divided by 24 to get the average optimum day value.

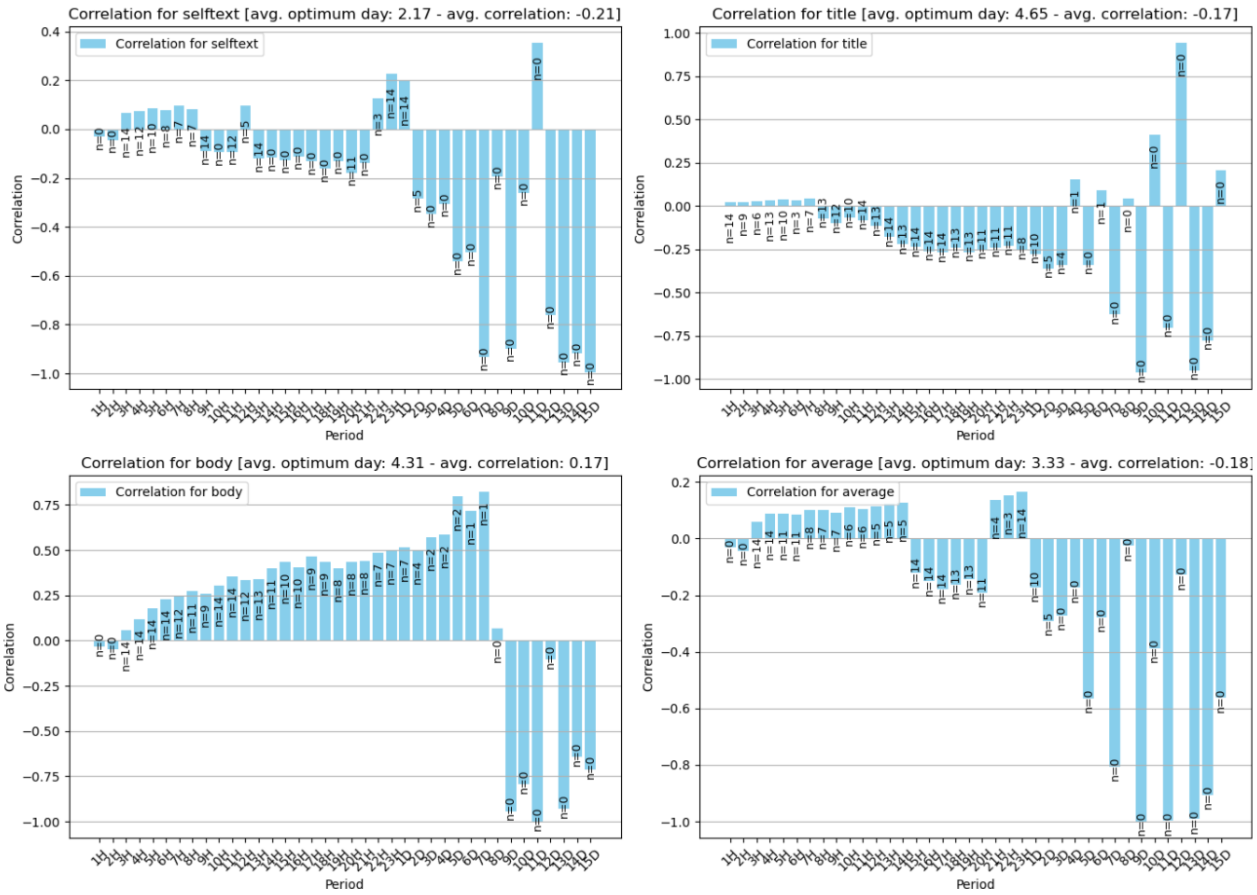


Figure 16: Average correlation between sentiment and n-period shifted cryptocurrency average price over different periods (without market data)

In Figure 16, correlation of sentiments can be seen. Normally, it is expected to get positive correlation between social media textual data and crpytocurrency market, since the sentiments affect the market in the same way as they are. Body (comments & replies) was the only one mostly correlated positive with the cryptocurrency average price. From this case, it can be thought that taking only body sentiment can provide a better model. 3.62 days after the sentiment is the average value which was found through the average of all.

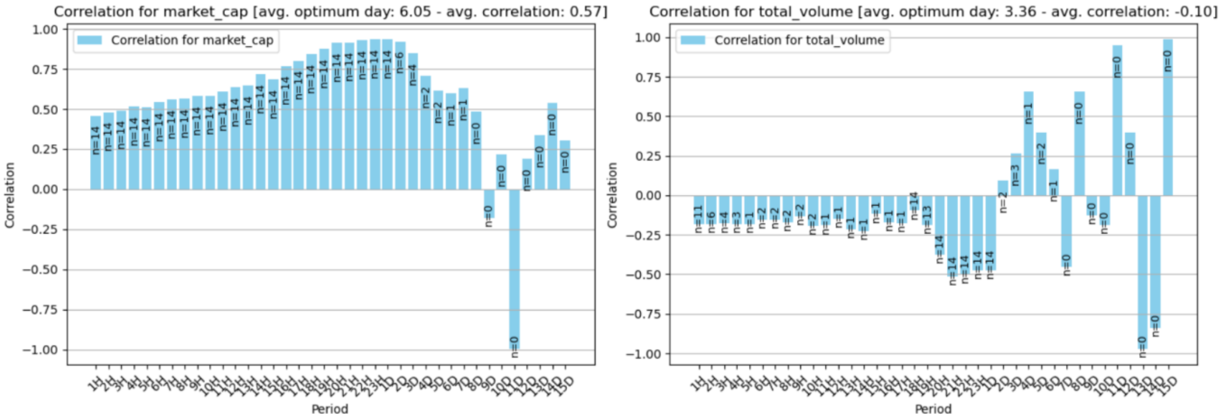


Figure 17: Average correlation between market data and n-period shifted cryptocurrency average price over different periods

In Figure 17, cryptocurrency-market-relevant data was taken into account to calculate the same values as Figure 16. Market cap was correlated generally strong positive (0.57), while total volume was not correlated well. Also, the average day value for market data was found as 4.7 days.

7.3.2. Weighted Correlation with Price

The weighted version of the previous implementation was performed by multiplying the count of the period with the summed net sentiment for each period

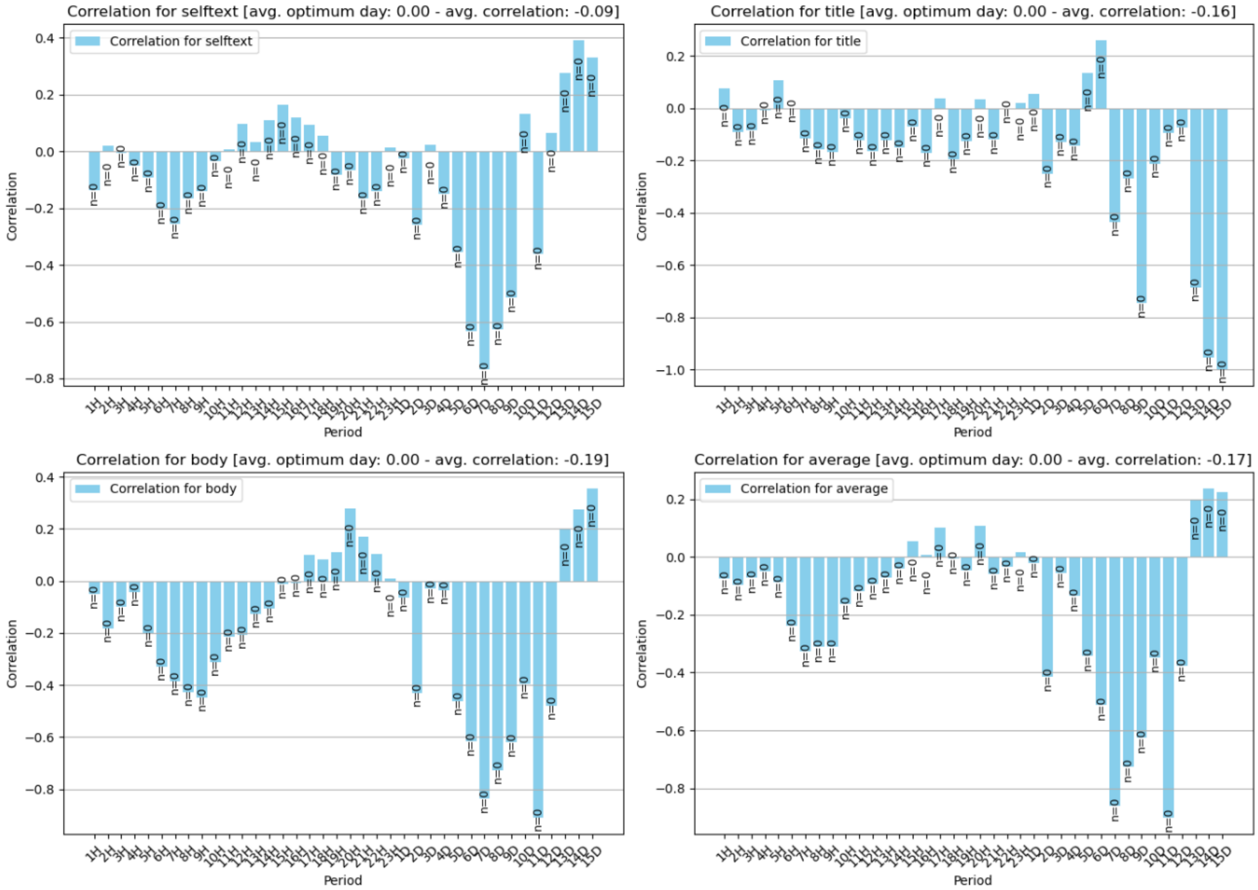


Figure 18: Weighted correlation between sentiment and n-period shifted cryptocurrency average price over different periods (without market data)

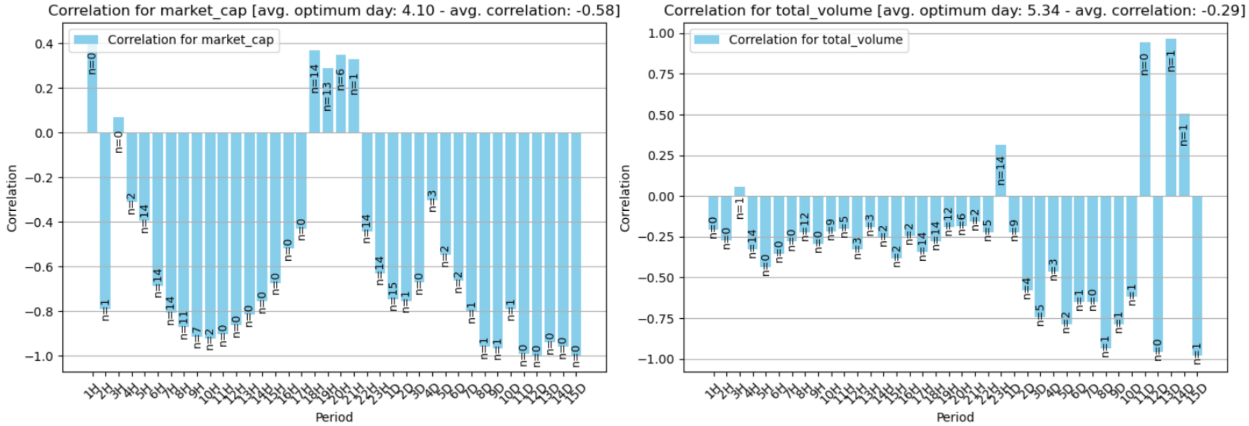


Figure 19: Weighted correlation between market data and n-period shifted cryptocurrency average price over different periods

They performed much worse, and results can be seen in Figure 18 and Figure 19 for without market data and with market data respectively. It can be seen that they don't have any proper relation between each other. That's why weighted method shouldn't be used in creating another metric from the sentiment scores when the collected data are taken into account. It could be related to social media data, but the similar graphs were obtained from cryptocurrency market data. Because of that, it cannot be associated with only the Reddit data and can be accepted as a general situation.

7.3.3. Average Trend Correlation with Price

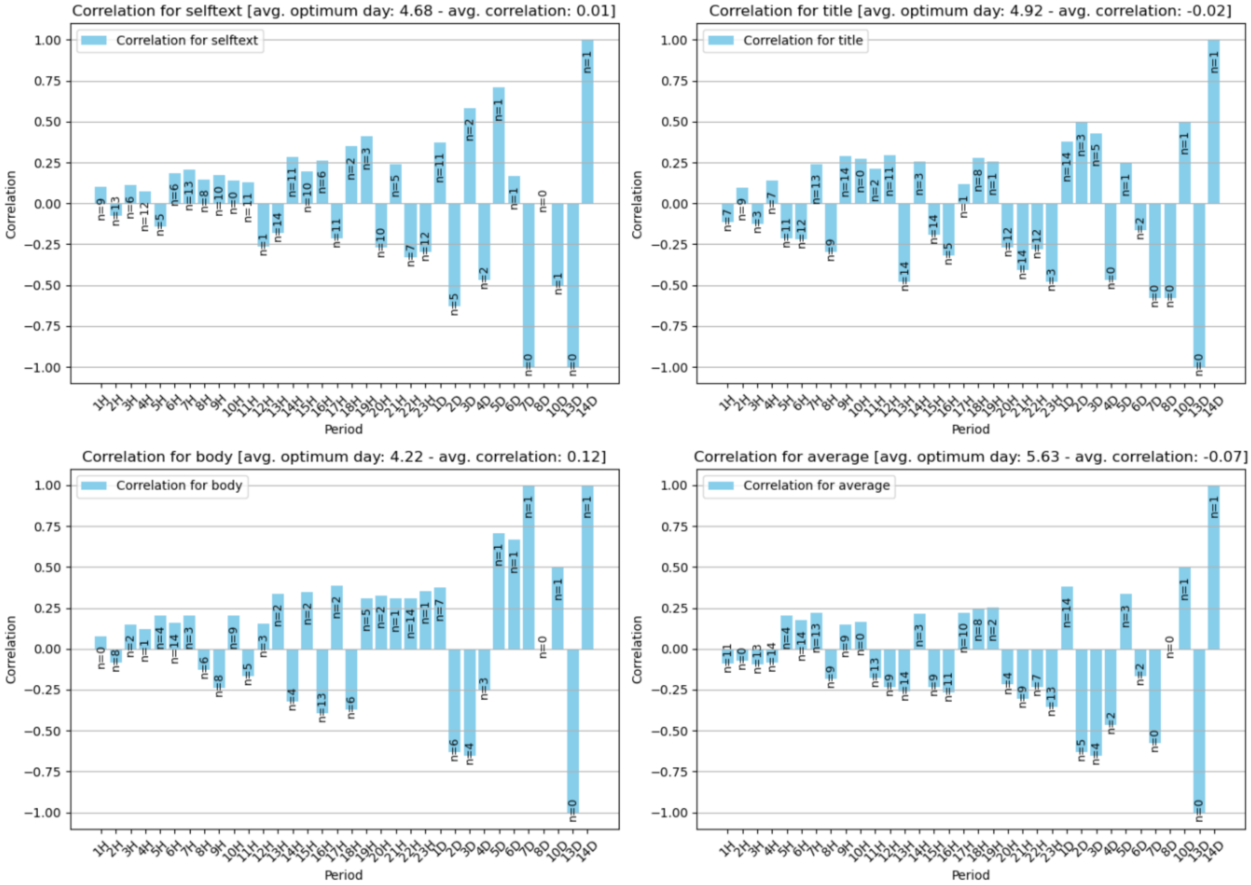


Figure 20: Average correlation between sentiment trend and n-period shifted cryptocurrency trend over different periods (without market data)

Across all in Figure 20, the average correlations were found generally weak, indicating that sentiment trends didn’t have a consistent relationship with cryptocurrency price trends over the periods analyzed, in spite of reaching the magnitude of 1 roughly after the long period of time.

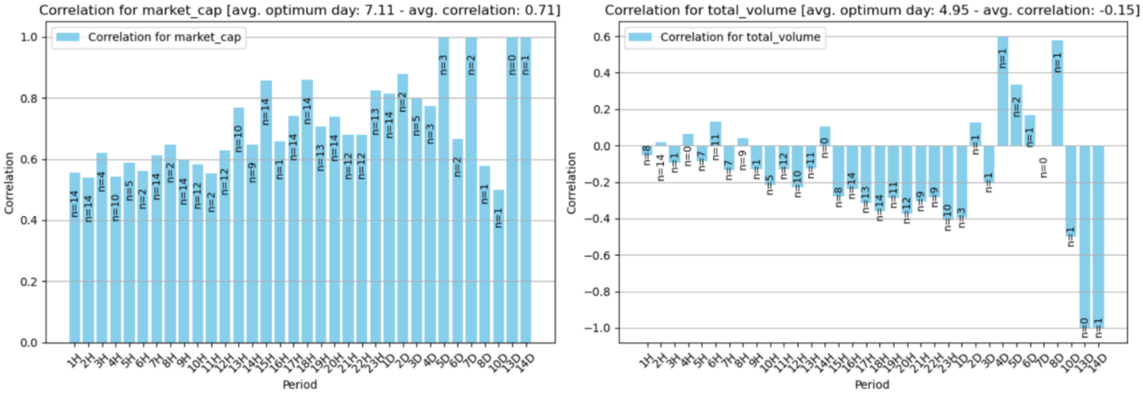


Figure 21: Average correlation between market trend and n-period shifted cryptocurrency trend over different periods

In Figure 21, the correlation between market capitalization and cryptocurrency price was generally strong and positive. The correlation values steadily increased as the period shifted except of inter periods at the end (6D, 8D, 10D), indicating that market capitalization might have a strong relationship with price movements. As for total volume, it didn't give any useful insight which can be used in either analysis or the model.

7.3.4. Subreddit-Cryptocurrency Relation

crypto_name	subreddit_name	period	correlation	source
tron	ethtrader	1H	0.3454	selftext
tron	CryptoMarkets	2H	0.396953	title
tron	CryptoMarkets	3H	0.437587	title
tron	CryptoMarkets	6H	0.478525	title
tron	CryptoMarkets	12H	0.573777	title
tron	CryptoMarkets	1D	0.621885	title
tron	ethtrader	2D	0.837522	average
tron	ethtrader	3D	0.920276	average
tron	ethtrader	5D	0.901892	average
solana	CryptoMoonShots	1W	0.971345	average

Table 19: Most correlated cryptocurrency-subreddit-sentiment with average cryptocurrency price over different periods

In Table 19, the most correlated cryptocurrency, subreddit, time of period, and which sentiment column can be seen. Tron was the most correlated cryptocurrency except of 1 week of period. The longer period of time the more correlated sentiments. In a week, the correlation went nearly 1, but it was long period of time to take decision in today's world. People can lose all wealth even in an hour. As for an hour, the cryptocurrency of tron and the subreddit of ethtrader were correlated in selftext column, in other words the texts in the post. Title was more correlated in 50% of all periods indicated in the table. Also, that there was no negative correlation can be seen.

subreddit_name	period	correlation	source
Bitcoin	1H	0.214991	selftext
Bitcoin	2H	0.22691	selftext
Bitcoin	3H	0.23118	selftext
Bitcoin	6H	0.243905	selftext
memecoins	12H	-0.296978	average
ethtrader	1D	0.491891	title
ethtrader	2D	0.640333	title
ethtrader	3D	0.595404	title
memecoins	5D	-0.791233	selftext
CryptoMoonShots	1W	0.871172	selftext

Table 20: The most correlated subreddit-sentiment with average cryptocurrency price over different periods

When the average cryptocurrency prices are taken into account in Table 20, the first 6 hours were more correlated with subreddit named Bitcoin followed by memecoins for 12 hours and 5 day. As for the highest correlation CryptoMoonShots hold with the rate of 0.87 for 1 week in. Unlike Table 19, Table 20 had negative correlated values in 2 out of 10. Selftext was the most popular source which had the most correlated sentiments.

In both cases, the longer period of time was required to have a higher correlation.

7.3.5. N-days After Sentiments

The posts shared on the internet are thought that they affect the cryptocurrency market. In order to figure out how many days later the market is affected, a simple experiment was conducted by nearly a month data. By using the same method in Average Correlation with Price, the most correlated n days were found for each cryptocurrency.

crypto	n_days	correlation
df_bitcoin	12	0.338746
df_ethereum	12	0.422758
df_tether	4	0.406752
df_binancecoin	12	0.44864
df_solana	12	0.283196
df_usd-coin	10	0.275638
df_ripple	12	0.393563
df_staked-ether	12	0.429521
df_dogecoin	12	0.464533
df_the-open-network	12	0.412601
df_cardano	**12**	**0.550696**
df_tron	12	0.241531

Table 21: Correlation between the best n-days for each cryptocurrency and sentiment difference rate

To find the best correlation for n days, the correlation between cryptocurrency price change and the net sentiment of the day was calculated for all collected cryptocurrencies. According to Table 21, 12 was found the optimal value in 10 out of 12 cryptocurrencies, which means that their prices are affected from the 12 previous days’ sentiments mostly. The highest correlation was found as 0.55 for cardano.

7.4. LSTM Models Results’

This model was designated to find the optimum values of the dataset for the preparation to the LSTM model for the further steps.

period	shifted_feature_number	n_later_predict	sentiment_par	mae	mse	rmse	r2	mape
1H	0	1	with	0.588678	0.624793	0.790439	0.848532	0.370698
1H	0	1	without	0.625252	0.679568	0.824359	0.835253	0.393722
1H	0	1	body	0.780846	0.906634	0.952173	0.780205	0.4906
1H	**0**	**1**	**body_market_cap**	**0.040661**	**0.002757**	**0.052511**	**0.826246**	**5.268417**
6H	0	1	with	1.401285	3.591258	1.895062	0.217584	0.883929
6H	0	1	without	1.380545	3.391184	1.841517	0.261173	0.871203
6H	0	1	body	1.463754	3.838999	1.959336	0.163609	0.922602
6H	**0**	**1**	**body_market_cap**	**0.018804**	**0.000628**	**0.025061**	**0.31088**	**8.835168**
1D	0	1	with	3.258559	13.609447	3.689098	-0.69693	2.053654
1D	0	1	without	2.6709	8.891045	2.981786	0.108604	1.689236
1D	0	1	body	2.420911	8.200837	2.86371	0.022543	1.534717
1D	**0**	**1**	**body_market_cap**	**0.032337**	**0.001324**	**0.036391**	**0.803834**	**21.25354**

Table 22: Performance Comparison of LSTM Models with and without Sentiment Data across Different Prediction Periods

The analysis in Table 22 showed that incorporating sentiment data improved short-term (1-hour) cryptocurrency predictions, with better accuracy than models without sentiment. However, as the prediction period extended to 6 hours and 1 day, the advantage of using the all sentiment data diminished. A model using only body sentiment and market cap had low absolute errors but struggled with percentage errors in the longer period of times. Overall, sentiment data helps in short-term predictions but can introduce noise in longer-term forecasts. From this result, 6H of period, 0 shifted feature, 1 period later prediction with body sentiment (comments & replies) and market capitalization were chosen as the best parameters due to having pretty less error values except of mape.

period	shifted_feature_number	n_later_predict	with_sentiment_par	mae	mse	rmse	r2	mape
1H	2	2	body_market_cap	0.057059	0.005765	0.075929	**0.651926**	**7.87136**
1H	3	1	body_market_cap	0.059202	0.004751	0.068926	**0.706102**	**7.535085**
1H	3	2	body_market_cap	0.056395	0.005126	0.071593	**0.692466**	**7.433393**
3H	0	1	body_market_cap	**0.02406**	**0.000911**	**0.030175**	0.544716	11.697807
6H	0	1	body_market_cap	**0.018799**	**0.000622**	**0.024944**	0.31728	8.869136
8H	0	1	body_market_cap	**0.01838**	**0.000524**	**0.02289**	0.35019	8.794255

Table 23: the best parameters for body sentiment & market capitalisation by each period

To be sure about the consistency of the parameters which were found in the previous section, another parameter experiment was conducted. Totally 47 different combination of period, shifted feature number, and n later predict was tried by taking body sentiment & market capitalisation into account. Additionally, the best models from each period were shown in Table 23. Even though mea, mae, and rmse are lower for higher periods, they couldn’t explain the variance better than the first 3 models according to R^2 value. Also, the mean absolute percentage error was lower in the first 3 models. Moreover, the period was 1 hour in all, which can conclude that making prediction in the shorter period of time can be more possible than long term.

period	shifted_feature_number	n_later_predict	with_sentiment_par	mae	mse	rmse	r2	mape
1H	**2**	**1**	**body_market_cap**	**0.033423**	**0.002169**	**0.04657**	**0.864992**	**4.518771**
1H	2	2	body_market_cap	0.065718	0.006983	0.083562	0.578421	9.10184
1H	**3**	**1**	**body_market_cap**	**0.036681**	**0.002497**	**0.049972**	**0.845514**	**4.934996**
1H	3	2	body_market_cap	0.050828	0.004621	0.067978	0.722733	6.889151

Table 24: the best 3 models’ parameters’ with metrics

The model parameters were found as can be seen in Table 24, where 2 of them outperformed shown with bold characters. These models provide a balanced approach by capturing enough historical data and focusing on a short period of time (1 hour), which makes them beneficial for a market where the prices can change drastically suddenly.

After this point, 2 of them were taken into account and conducted the last experiment for data processing parameters.

period	shifted_feature_number	n_later_predict	with_sentiment_par	mae	mse	rmse	r2	mape
1H	2	1	body_market_cap	0.040574	0.002719	0.052141	0.83076	5.33018
1H	**3**	**1**	**body_market_cap**	**0.034972**	**0.002317**	**0.048136**	**0.856657**	**4.701509**

Table 25: the last 2 best model parameters’ comparison

As it can be seen in Table 25, the model which had 3 of shifted feature number, and 1 period later prediction was slightly better than the other model. This finding can be important for traders or analysts looking to optimize prediction models by small changes.

mae	mse	rmse	r2	mape	unit 1	unit 2	dense Unit	dropout 1	dropout 2	optimizer	epochs	batch size
0.03497	0.00232	0.04814	0.85666	4.70151	50	50	150	0.20656	0.24432	adam	100	32

Table 26: Final model parameters with the model performance metrics

The final model's performance metrics are quite strong (Table 26), with an MAE of 0.03497 and an R² of 0.85666. These values suggest the model is highly effective at predicting hourly cryptocurrency prices when both sentiment and market data are considered.

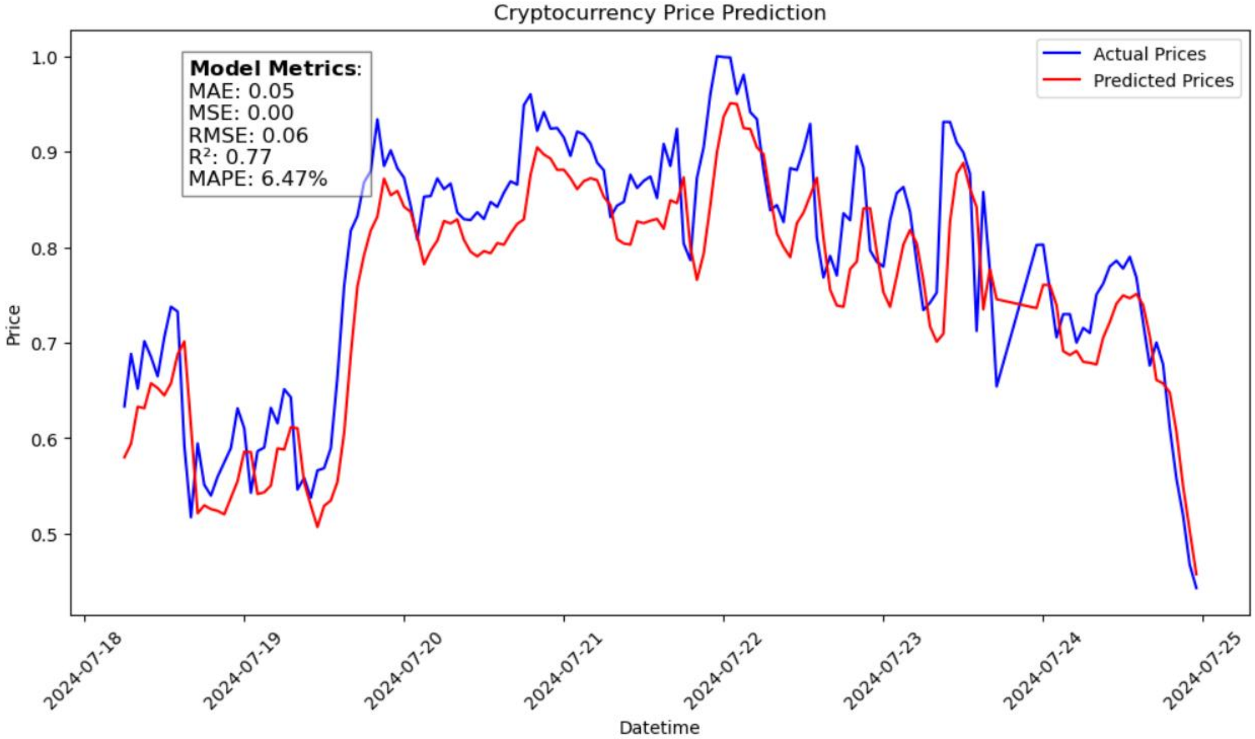


Figure 22: Price prediction and actual value as scaled data

After using the best data preprocessing parameters and hyperparameters, the predictions and the model metrics were obtained like in Figure 22. It shows how close the predicted cryptocurrency prices are to the actual prices. The model used did a good job, with small differences between what it predicted and the real prices. Most of the time, the predicted line follows the actual prices closely, which means the model is effective. There are a few times when the predictions are slightly off, especially when prices change quickly. But overall, this model seems to work well and could help traders make better decisions.

8. Discussion

This study shows some promising results, but it's important to remember that it’s based on only about a month’s worth of data. To get more accurate and reliable results, it would be necessary to analyze a larger dataset that includes more sources. Limiting the research to just a short period like this doesn’t give us the full picture, so expanding the data range is crucial.

On the other hand, as LLMs, only BERT based models were used. To use other state-of-art models such as GPT new versions, and Llama can make the sentiment analysis better.

The findings suggest that using an LSTM model combined with sentiment analysis can be effective in predicting cryptocurrency prices in the short term. The model did a good job of following actual price trends, showing that market sentiment plays a significant role in price movements. However, there were a few moments when the predictions were a bit off, especially during rapid price changes, which means there's still room for improvement.

These results can be very useful for real-world trading. Traders can use this model when quick decisions are needed. By including sentiment analysis, which captures how people feel about the market, this approach offers more accurate forecasts and potentially lead to better trading decisions.

8.1. Limitations of the Study

There are some limitations to this study that need to be emphasized. The biggest one is the limited dataset, which might not capture all the complexities of the cryptocurrency market. Also, the model might have some biases because it mainly relies on certain data sources, like specific subreddits. These factors could make the model less effective in different market conditions or over longer periods. On the other hand, these subreddits, or the website will may be closed in the future. Then, other metrics will might be needed.

8.2. Suggestion for Future Research

There is no doubt that the future research should make the gathering more data from more sources over a longer period possible. These sources can be other social media platforms as well as more reliable sources such as financial news. Additionally, considering non-textual data like images and videos could open up new possibilities for understanding market sentiment and improving predictions. Because many people share their ideas through the most used memes on the internet.

On the other hand, different types of LSTM models can be used, or they can be combined with other machine learning algorithms.

9. References

A New Forecasting Framework for Bitcoin Price with LSTM. (2018, November 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8637486

Arslan, S. (2024). Bitcoin Price Prediction Using Sentiment Analysis and Empirical Mode Decomposition. *Computational Economics*. https://doi.org/10.1007/s10614-024-10588-3

Barbieri, F., Camacho-Collados, J., Espinosa-Anke, L., & Neves, L. (2020). TweetEval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*. Retrieved from https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Bhatt, S., Ghazanfar, M., & Amirhosseini, M. H. (2023). *Machine Learning based Cryptocurrency Price Prediction using Historical Data and Social Media Sentiment*. https://doi.org/10.5121/csit.2023.131001

Chaum, D., Fiat, A., & Naor, M. (1990). Untraceable Electronic Cash. In *Lecture notes in computer science* (pp. 319–327). https://doi.org/10.1007/0-387-34799-2_25

Cho, K., Bart, V. M., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. arXiv.org. https://arxiv.org/abs/1406.1078

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018, October 11). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv.org. https://arxiv.org/abs/1810.04805v1

Farimani, S. A., Jahan, M. V., & Fard, A. M. (2022). From Text Representation to Financial Market Prediction: A Literature Review. *Information*, *13*(10), 466. https://doi.org/10.3390/info13100466

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*(10), 2222-2232. https://doi.org/10.1109/TNNLS.2016.2582924

Hartmann, J., et al. (2023). *Sentiment-roberta-large-english [Machine Learning Model].* Retrieved from https://huggingface.co/siebert/sentiment-roberta-large-english

Hazourli, A. (2022). FinancialBERT - A pretrained language model for financial text mining. *ResearchGate*. https://doi.org/10.13140/RG.2.2.34032.12803. Retrieved from https://huggingface.co/Sigma/financial-sentiment-analysis

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, Allen H., Hui Wang, and Yi Yang. "FinBERT: A Large Language Model for Extracting Information from Financial Text." *Contemporary Accounting Research* (2022). Retrieved from https://huggingface.co/yiyanghkust/finbert-tone

Husbands, D., Perach, R., & Buchanan, T. (2024, January 17). *Some people who share fake news on social media actually think they're helping the world*. The Conversation. https://theconversation.com/some-people-who-share-fake-news-on-social-media-actually-think-theyre-helping-the-world-215623

Hutto, C. J., & Gilbert, E. (2015). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*.

Hutto, C. J., & Gilbert, E. (2015). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*

Kalchbrenner, N., Danihelka, I., & Graves, A. (2015). Grid Long Short-Term Memory. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1507.01526

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019, July 26). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv.org. https://arxiv.org/abs/1907.11692

Loria, S. (2018). *TextBlob: Simplified Text Processing*. TextBlob Documentation. Retrieved from https://textblob.readthedocs.io/en/dev/

Maheshwari, T., Bharadwaj, S., Sharma, N. K., Mudegol, G., & Srivastav, S. (2024). Bitcoin Price Prediction Using Lstm. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4482735

Miah, M. S. U., Kabir, M. M., Sarwar, T. B., Safran, M., Alfarhood, S., & Mridha, M. F. (2024). A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and LLM. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-60210-7

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January 16). *Efficient Estimation of Word Representations in Vector Space*. arXiv.org. https://arxiv.org/abs/1301.3781v1

Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L. T., & Trajanov, D. (2020). Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers. *IEEE Access*, *8*, 131662–131682. https://doi.org/10.1109/access.2020.3009626

Nakamoto, S. (2008) Bitcoin A Peer-to-Peer Electronic Cash System. - References - Scientific Research Publishing. https://www.scirp.org/reference/ReferencesPapers?ReferenceID=1522950

NIST/SEMATECH. (2012). *e-Handbook of statistical methods* (Section 1.3.5.11, Measures of skewness and kurtosis). National Institute of Standards and Technology. https://doi.org/10.18434/M32189

Peirsman, Y. (n.d.). *BERT-base-multilingual-uncased-sentiment [Machine Learning Model].* Retrieved August 8, 2024, from https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment

Pekin, A., Ulusoy, I. E., & Takçı, H. (2024). Machine learning based cryptocurrency price prediction using financial news. *Science and Technology Publishing (SCI & TECH), 8*(6). ISSN: 2632-1017

Predicting the Price of Bitcoin Using Machine Learning. (2018, March 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8374483

ProsusAI. (n.d.). *FinBERT [Machine Learning Model].* Retrieved August 8, 2024, from https://huggingface.co/ProsusAI/finbert

Reimers, N., & Gurevych, I. (2019, August 27). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. arXiv.org. https://arxiv.org/abs/1908.10084

Roumeliotis, K. I., Tselikas, N. D., & Nasiopoulos, D. K. (2024). LLMs and NLP Models in Cryptocurrency Sentiment Analysis: A Comparative Classification Study. *Big Data and Cognitive Computing*, *8*(6), 63. https://doi.org/10.3390/bdcc8060063

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper, and lighter. *arXiv preprint arXiv:1910.01108*. Retrieved from https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019, October 2). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv.org. https://arxiv.org/abs/1910.01108v1

Seabe, P. L., Moutsinga, C. R. B., & Pindza, E. (2023). Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. *Fractal and Fractional*, *7*(2), 203. https://doi.org/10.3390/fractalfract7020203

Sharma, N. A., Ali, A. B. M. S., & Kabir, M. A. (2024). A review of sentiment analysis: tasks, applications, and deep learning techniques. *International Journal of Data Science and Analytics*. https://doi.org/10.1007/s41060-024-00594-x

Tweet Sentiment Analysis for Cryptocurrencies. (2021, September 15). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9558914

Van Rossum, G., & Drake, F. L. (2012). *Python: The Python programming language* (Python Documentation). Python Software Foundation. Retrieved August 8, 2024, from https://www.python.org/doc/essays/blurb/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June 12). *Attention Is All You Need*. arXiv.org. https://arxiv.org/abs/1706.03762v1

Vlahavas, G., & Vakali, A. (2024). *Dynamics Between Bitcoin Market Trends and Social Media Activity*. https://doi.org/10.20944/preprints202407.0112.v1

Yang, Y., Uy, M. C. S., & Huang, A. (2020, June 15). *FinBERT: A Pretrained Language Model for Financial Communications*. arXiv.org. https://arxiv.org/abs/2006.08097v1

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021, June 21). *Dive into Deep Learning*. arXiv.org. https://arxiv.org/abs/2106.11342