INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Previsão Da Tendência da Bitcoin Utilizando Extração de Sentimentos do Twitter

ALEXANDRE EMANUEL MARTINS MENDES
julho de 2022

P.PORTO

Forecasting Bitcoin Trend Using Sentiment Extraction From Twitter

Alexandre Emanuel Martins Mendes
Student No.: 1200128

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Artificial Intelligence

Supervisor: Doutor João Miguel Ribeiro Carneiro, Professor Adjunto da Escola Superior de Tecnologia e Gestão do Instituto Politécnico do Porto
Co-Supervisor: Doutora Maria Goreti Carvalho Marreiros, Professora Coordenadora com Agregação do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Evaluation Committee:
President:
Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Members:
Doutor Paulo Jorge Freitas de Oliveira Novais, Professor Catedrático da Universidade do Minho
Doutor João Miguel Ribeiro Carneiro, Professor Adjunto da Escola Superior de Tecnologia e Gestão do Instituto Politécnico do Porto

Porto, June 30, 2022

Abstract

Bitcoin is the first decentralized digital currency constituting a successful alternative economic system. As a result, the Bitcoin financial market occupies an important position in society, where it has gained increasing popularity. The correct prediction of this type of market can drastically reduce losses and maximize investor profits. One of the most popular aspects of predicting the cryptocurrency market is the analysis of sentiment in posts shared publicly on social networks. Currently, the Twitter platform generates millions of posts a day, which has attracted several researchers in search of problem solving using sentimental analysis in tweets.

With this evolution, it is intended to develop, through Artificial Intelligence (AI) techniques, models capable of predicting the Bitcoin trend based on daily sentimental analysis of posts made on the Twitter platform with Bitcoin's historical data. Specifically, it is intended to assess whether sentiment positively influences the Bitcoin trend, and whether positive, neutral and negative feelings positively influence the Bitcoin trend in the same way. Finally, it is also objective to assess whether indicators such as market volume and the volume of tweets carried out within the scope of the Bitcoin theme positively influence its trend.

To validate the potential of the study, two AI models were developed. The first model was created to classify the sentiments of tweets into three typologies: positive, neutral and negative. This model focused on AI techniques based on Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (BI-LSTM) and Convolutional Neural Network (CNN). In turn, the second model was designed to classify Bitcoin's future trends into strong uptrend, uptrend, downtrend and strong downtrend. In this sense, the model focused on AI techniques based on LSTM and Random Forest Classifier.

In general, it was possible to achieve good performance in the development of sentiment classification models, achieving an accuracy value of 87 % in the LSTM and BI-LSTM models and 86% in the model based on CNN technology. Regarding the model focused on predicting the Bitcoin trend, it was possible to validate that sentiment positively influences the Bitcoin trend prediction. More interestingly, neutral sentiment volume has a more significant impact on Bitcoin trend prediction. The Random Forest Classifier technique proved to be the best, recording accuracy of 57.35% in predicting the Bitcoin trend. Removing the sentiment variable made it possible to verify a cadence of 15% to 20% in the Bitcoin trend forecast, which effectively validates that sentiment positively influences the trend forecast.

Keywords: Sentiment Analysis, LSTM, BI-LSTM, CNN, Bitcoin, Bitcoin Trend Prediction, Random Forest

Resumo

A Bitcoin é considerada a primeira moeda digital descentralizada constituindo um sistema económico alternativo de sucesso. Em resultado, o mercado financeiro da Bitcoin ocupa uma posição importante na sociedade, onde tem vindo a angariar cada vez mais popularidade. Prever acertadamente este tipo de mercado pode reduzir drasticamente as perdas e maximizar os lucros dos investidores. Um dos aspetos mais populares, quando se trata de prever o mercado de cryptomoedas, passa pela análise de sentimentos em posts partilhados publicamente em redes sociais. Atualmente, a plataforma do Twitter, gera milhões de posts todos os dias, o que tem atraído diversos investigadores na procura de resoluções de problemas com recurso à análise sentimental em tweets.

Com esta evolução, pretende-se desenvolver através de técnicas de Inteligência Artificial (IA), modelos capazes de prever a trend da Bitcoin com base numa análise sentimental diária dos posts efetuados na plataforma do Twitter com os dados históricos da Bitcoin. Em específico, tenciona-se avaliar se o sentimento influencia positivamente a trend da Bitcoin, bem como avaliar se os sentimentos positivos, neutros e negativos, de forma isolada, influenciam da mesma forma positivamente a trend da Bitcoin. Por fim, é ainda objetivo, avaliar se indicadores como o volume de mercado e o volume de tweets realizado no âmbito do tema da Bitcoin influenciam positivamente a trend da mesma.

De forma a validar o potencial do estudo, foram desenvolvidos dois modelos de IA. O primeiro modelo foi criado para efetuar a classificação de sentimentos dos tweets em três tipologias: positivos, neutros e negativos. Este modelo, focou-se em técnicas de IA basedas em LSTM, BI-LSTM e CNN. Por sua vez, o segundo modelo foi elaborado para classificar as trends futuras da Bitcoin em quatro tipologias: strong uptrend, uptrend, downtrend e strong downtrend. Neste sentido, o modelo focou-se em técnicas de IA baseadas em LSTM e Random Forest Classifier.

Em geral, foi possível atingir uma boa performance no desenvolvimento dos modelos de classificação de sentimento, atingindo um valor de accuracy de 87% nos modelos LSTM e BI-LSTM, e 86% no modelo baseado na técnica de CNN. Em relação ao modelo focado em prever a trend da Bitcoin, foi possível validar que o sentimento realmente influencia positivamente a previsão da trend da Bitcoin. Mais curiosamente, verificou-se que o volume de sentimento neutro tem um impacto mais significativo na previsão da trend da Bitcoin. A técnica Random Forest Classifier demonstrou ser a melhor, registando uma accuracy de 57,35% na previsão da trend da Bitcoin. Ao remover a variável sentimento foi possível verificar uma cadência de 15% a 20% na previsão da trend da Bitcoin, o que valida efetivamente que o sentimento influencia positivamente a previsão da trend.

Palavras-chave: Sentiment Analysis, LSTM, BI-LSTM, CNN, Bitcoin, Bitcoin Trend Prediction, Random Forest

Acknowledgement

The pursuit of a master's degree was never part of my plan. However, the field of AI has always sparked interest and curiosity. So when Instituto Superior de Engenharia do Porto (ISEP) opened an AI course, i did not think twice about applying. To this day, i do not regret it.

All my evolution was clearly due to all the teachers I had the opportunity to interact with and who managed to share knowledge in the best possible way, which enabled me to reach this final stage.

The fact that i already have professional experience and I'm surrounded by high-quality professionals has significantly improved my attitude and posture when taking this master's degree. All those who help me every day to be a better professional and make good decisions in the most adverse situations are worth mentioning. Thanks to André Duarte, who has always helped me to be mentally motivated to finish this course and offered the necessary support to achieve this goal successfully. A big thanks to Blip as a company, which helped in every possible way by providing the flexibility necessary to complete this master's degree successfully. A big thank you to all the friends who have supported me on this journey and who have always supported me in every aspect. Mentioning Maria, who helped and supported in every aspect of this study.

I want to thank João Carneiro and Goreti Marreiros for being the supervisors of this study, where they offered all the necessary support for the development of this study. Highlighting the essential help of João Carneiro in the final phase of this study. Supporting and accompanied every day in the implementation phases, and always guiding to achieve better results.

Finally, i would like to thank my family members who always offered me the necessary tools that helped to archive this goal successfully. Highlight the effort that my parents made throughout my life that enabled me to successfully complete this master's degree independently.

Contents

List of Figures

List of Tables

List of Source Code

List of Acronyms

AI	Artificial Intelligence.
ANN	Artificial Neural Network.
API	Application Programming Interface.
BI-LSTM	Bidirectional Long Short Term Memory.
CNN	Convolutional Neural Network.
CPU	Central Processing Unit.
CSV	Comma-Separated Values.
DL	Deep Learning.
DNN	Deep Neural Network.
GB	Gigabyte.
GDPR	General Data Protection Regulation.
GLM	Generalized Linear Model.
GPU	Graphics Processing Unit.
GRU	Gated Recurrent Unit.
HTTP	Hypertext Transfer Protocol.
IA	Inteligência Artificial.
IDE	Integrated Development Environment.
IMDB	Internet Movie Database.
ISEP	Instituto Superior de Engenharia do Porto.
JSON	JavaScript Object Notation.
LSTM	Long Short Term Memory.
MB	Megabyte.
MCC	Matthews Correlation Coefficient.
ML	Machine Learning.
NER	Named-entity recognition.
NLP	Natural Language Processing.
NLTK	Natural Language Toolkit.
NN	Neural Network.
OCHL	Open Close High Low.

RAM	Random Access Memory.
REST	Representational state transfer.
RMSE	Root Mean Square Error.
RNN	Recurrent Neural Network.
SMA	Simple Moving Average.
SSD	Solid-State Driver.
SST2	Stanford Sentiment Treebank v2.
SVM	Support Vector Machine.
TPU	Tensor Processing Unit.

Chapter 1

Introduction

1.1 Contextualization

Financial market systems have an important position in modern society, and have long been one of the most attractive pillars of economic investment (Hao et al. 2021). Accurately predicting market behaviour can potentially reduce unexpected risks and maximize profits (Hao et al. 2021). In recent years, the analysis of social sentiments has become widely recognized and in particular interest for researchers, companies, governments and organizations (Birjali, Kasri, and Beni-Hssane 2021). This typology of analysis is the key to the development of Artificial Intelligence (AI) and has been one of the most referenced areas of investment for large companies and institutions, such as Thomson Reuters, Bloomberg, banks and hedge funds (Birjali, Kasri, and Beni-Hssane 2021). As these major financial market players began investing in sentiment analysis to improve their trading models, researchers interest in predicting financial markets grew. In this vein, they started by providing trade sentiment analysis services and exploring investor sentiment to help make better predictions about financial markets. Most of these institutions reported using sentiment analysis on their structured transaction data, such as past prices, historical earnings and dividends, to improve their sophisticated Machine Learning (ML) models for trading (Audrino, Sigrist, and Ballinari 2020).

With the evolution of financial markets and global interest by many people, Bitcoin appeared as the first completely decentralized digital currency, created by Satoshi Nakamoto in 2008 (Nakamoto 2009), in anonymity until today. This cryptocurrency is distributed all over the world and can be bought and sold on any computer connected to the Internet (Guo et al. 2021). Bitcoin is neither controlled nor supervised by any authority, government or financial institution, but by a peer-to-peer network of users who control the creation and transfer of coins. Bitcoin's independence from third party intermediaries provides its users with a highly desired level of privacy and convenience. According to Coinmarketcap (Coinmarketcap 2022), Bitcoin represents more than 43.95% of the dominance of the entire cryptocurrency market and remains the leader. Since its creation in 2008, Bitcoin has gradually gained traction around the world. Due to its innovation, market position and price fluctuation, many researchers started developing many ML models to predict Bitcoin's price, in order to facilitate investment decisions (Guo et al. 2021). Research's focus is to assess the impact of social media data on the profitability of trading strategies that predict short-term Bitcoin price movements.

In this sequence, the growing evolution of the Internet and social networks in recent years has encouraged the sharing of thoughts, feelings, opinions, as well as the exchange of information and experiences through simple and interactive social networks such as Facebook, Instagram,

Twitter, Blogs, TikTok, etc (Sattarov et al. 2020). This sharing of opinions and feelings is of great importance in our daily lives and therefore it is necessary to analyze this data generated by users in order to automatically monitor public opinion and assist in decision making (Birjali, Kasri, and Beni-Hssane 2021). In particular, Twitter became widely used by researchers to predict sentiment as well as people movements for a wide range of events, particularly for financial markets as Bitcoin. Twitter users generates every day huge amounts of data for different topics like cryptocurrencies and tweets sentiment analysis was found to have a predictive power for Bitcoin's price. Strong evidence has indicated that the collective opinion of individuals is as reliable as that of a single expert, in the same way that the use of a large number of tweets will have a positive impact on the Bitcoin price prediction (Matta, Lunesu, and Marchesi 2015).

In this context the use of ML techniques became very powerful to extract valuable information from social media then correlate the same with historical Bitcoin data. Simultaneously, Natural Language Processing (NLP) become one of the powerful techniques to analyse sentiment expressed on tweets, then supervised techniques to correlate sentiment with historical Bitcoin's data. Thus, there is a need to develop tools that allow extracting the sentiment expressed then correlating it with historical Bitcoin data. Despite the existence of ML models to predict tweets sentiment, it becomes relevant to correlate sentiment with Bitcoin historical data dynamically and automatically, with configurable tools, interactive user interfaces to allow people to easily analyse data and visualize social movements before Bitcoin financial market moves, to better use the data.

1.2 Research Hypotheses

Given the current fame of the cryptocurrency market, especially Bitcoin, it is believed that there may be a certain correlation between social power and the financial market, especially at the level of the Bitcoin trend in the way this idea was considered and interpreted. Given the volume of data generated daily on the Twitter platform by many people worldwide, there are multiple benefits associated with processing and analyzing it. Sentiment analysis in tweets is an excellent example of extracting useful information from the thousands or millions of data generated daily on this topic. Suppose the price of Bitcoin is affected by information people share through tweets. In that case, the sentiments expressed must be significant predictors. In this follow-up, what would be the value of using only cryptocurrency market indicators if it is impossible to observe people's movement on social networks? On the other hand, how frustrating can it be in case there is an adverse movement by people that can negatively influence the price and trend? How significant would the impacts of a simple tweet made by Elon Musk be?

With this study, it is intended to answer these questions using AI techniques to benefit from the feelings expressed on Twitter's social network in conjunction with Bitcoin market values. Based on this proposed challenge, the following hypotheses were defined and formulated:

H1 - It is possible to predict the trend of Bitcoin in the short term.

H2 - There is a correlation between the sentiment expressed in the tweets and the Bitcoin trend.

H3 - The sentiment expressed in the tweets positively correlates with the historical trend of Bitcoin.

H4 – The negative sentiment expressed in the tweets positively correlates with the historical trend of Bitcoin.

H5 – The neutral sentiment expressed in the tweets positively correlates with Bitcoin's historical trend.

H6 – The volume of tweets correlates with the Bitcoin trend.

H7 - The volume of transactions carried out on Bitcoin correlates with the Bitcoin's trend.

1.3 Objectives

The main objective of this study is to understand the influence of sentiments expressed in tweets on the Bitcoin trend by developing two ML models: one classifying tweets as positive, negative or neutral, and a second model to correlate sentiment acquired in the first model with the historical trend of Bitcoin to predict future trends.

With the implementation of these two models, it is expected to acquire new knowledge previously unknown or difficult to visualize, allowing to verify new application cases for negotiation processes in cryptocurrency markets. In addition, it becomes possible to help companies, institutions, day traders, or even individuals make the best decisions about Bitcoin trends, increasing their profits.

In detail, this study is designed with the focus on the following objectives:

- Understand state of the art in the following areas: Neural Network (NN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), sentiment analysis and Bitcoin trend prediction;
- Develop of a ML a model that can classify sentiment in tweets according to user intentions;
- Develop of a ML a model capable of predicting a trend, positive or negative, for the next day according to current market values;
- Demonstrate that the sentiment expressed on the Twitter network impacts the prediction of the Bitcoin trend.

1.4 Methodologies

In order to achieve the defined goal, a methodology called Scrum (Carvalho and Mello 2011) was used. This methodology is an inherently flexible method, which can adapt to the unpredictable nature of research. To achieve this goal, it provides the tools to allow researchers in focusing on one project at a time and explore the potential of the work to increase motivation and productivity. In this way, the researcher systematically defines the steps to realise the problem, dividing it into several smaller objectives. After this division, a timeline is defined for the delivery of smaller goals, which can be weekly, fortnightly or monthly. These tasks are described below:

- Collection of relevant data to the state of the art, by gathering all articles written by other researchers who intend to solve the same problem;
- Definition of architecture to achieve a more general visibility of the problem and how it is addressed in implementation;

- Collection of datasets, namely, collection of all data necessary to carry out this study, from the collection of tweets from the Twitter platform to the collection of historical data related to Bitcoin;
- Analysis of previously collected data for detailed analysis, and research to identify possible changes and necessary adjustments;
- Development of a solution capable of classifying feelings in tweets through the exploration and implementation of AI models to compare the techniques which obtain better performance;
- Infer a new dataset with the previously implemented model;
- Development of a solution capable of predicting the Bitcoin trend for the next day, using the exploration and implementation of AI models and validation of hypotheses of this study;
- Analysis of the results and formulation of the conclusion: this phase consists of analysing and validating the developed prototype.

1.5 Document Structure

This dissertation presents a structure composed of 4 chapters, which will ensure an easy understanding for the readers, particularly the perception of all the themes addressed and developed.

The first introductory chapter is intended to introduce the history of the cryptocurrency world and the relationship between social networks and the Bitcoin trend. This chapter presents the research hypotheses, the objectives to be achieved, and the methodology selected for the development of this study.

Then the chapter of the Literature Review appears, with the intention of revealing the history, investigations and papers written about neural networks, RNN and CNN networks, focusing later on sentiment analysis and finally on the prediction of the Bitcoin trend.

In the third chapter, the methods are described, in which the technologies used for the implementation of this study are introduced. Then, subsections are presented on safety and ethics regarding the data collected for the development of the study and on the architecture where the work carried out is intended to be visualized in general. Also, in this chapter, the entire development phase of the sentiment analysis model is described, from data collection, exploration, pre-processing phase, implemented models, and finally, the results obtained through the experiments.

Finally, a fifth sub-topic addresses the implementation of the Bitcoin trend prediction model, where the phases of dataset collection, data exploration and pre-processing are performed, models implemented and results obtained through the experiments performed.

Chapter 2

Literature Review

Research and software development implies the study of similar implementations or with the same objective that has already been carried out to find aspects that can be improved or even problems that can be solved in order to find an efficient solution. The literature review chapter is structured on several sections and sub sections according to the relevant topics and methods used in this dissertation. Section 2.1 addresses the basis of NN. Section 2.2 the basis of RNN. Section 2.3 the basis of CNN. Section 2.4 what is sentiment analysis. Finally section 2.5 the bitcoin's trend prediction.

2.1 Artificial Neural Network

ML attracts a lot of interest because it provides new tools to successfully reveal patterns from complex and unstructured big data. The reason is that the assumptions in classical statistical techniques about the underlying structure are not considered necessary anymore. ML achieves such a breakthrough, as it turns the deductive problem of finding a rule to an inductive one by letting the data inform us of the best rule characterizing data.

Artificial Neural Network (ANN) is a soft computing tool that mimics the ability of the human mind to employ modes of reasoning and pattern recognition. ANN learn from the relationships between input and output provided through training data, and could generalize the output, making it suitable for non-linear problems where experience and surrounding conditions are the key features (Kulkarni, Londhe, and Deo 2017). Typically they consist of three layers. The input layer with input neurons, the hidden layer(s) with hidden neurons, and output layer with output neurons. On figure 2.1 it is represented a basic ANN architecture.

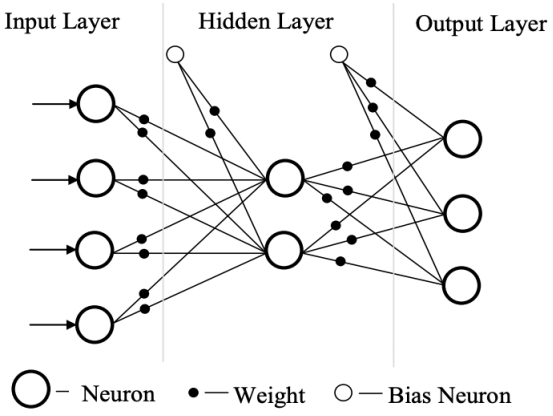

Figure 2.1: Basic Artificial Neural Network Architecture.

Each neuron in the input layer is connected to each neuron in the hidden layer, and every neuron in a hidden layer is connected to each neuron in the output layer. The number of hidden layers and the amount of neurons in each hidden layer can be one or more. Before its application, the network is trained until a very low value of the error is achieved. The network will then be tested with an unseen set of data to assess the accuracy of the developed model (Kulkarni, Londhe, and Deo 2017).

ANN resemble biological functions of a human nervous system (Abiodun et al. 2018). The human brain processes information through complex signals that easily coordinate the human to perform a task. ANN can be designed to perform certain functions like data classification and pattern recognition through learning. One of the great advantages of ANN is the ability to learn from complex and large amounts of data (Abiodun et al. 2018).

2.2 Recurrent Neural Network

Long Short Term Memory (LSTM) is a popular RNN architecture for modelling sequential data, designed to capture long-term dependencies better than the vanilla RNN models. As with other types of RNN, the LSTM network receives input from the current time-step and output from the previous time-step at each time-step, and produces an output fed to the next time step. The hidden layer from the last time-step is then used for classification. The high-level architecture of a LSTM network is shown in Figure 2.2, (Minaee, Azimi, and Abdolrashidi 2019).

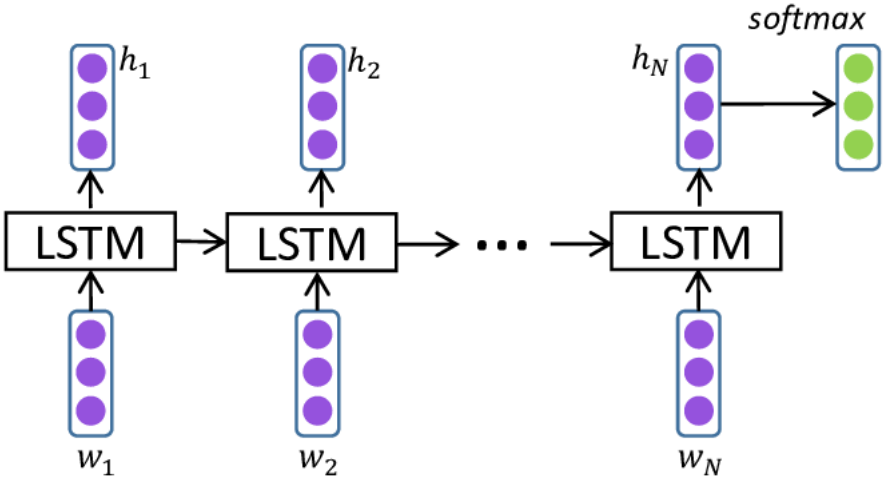

Figure 2.2: Standard LSTM Architecture.

LSTM came to fulfil the promise of vanilla RNN, which often suffers from gradient vanishing problems. The LSTM follows an architecture consisting of one memory cell and three gates, the input gate, the output gate and the forget gate. The memory cell is responsible for remembering past values, and the gates regulate the flow of information passed into and out of the cells. Figure 2.3 intends to illustrate the architecture of an LSTM module (Minaee, Azimi, and Abdolrashidi 2019).

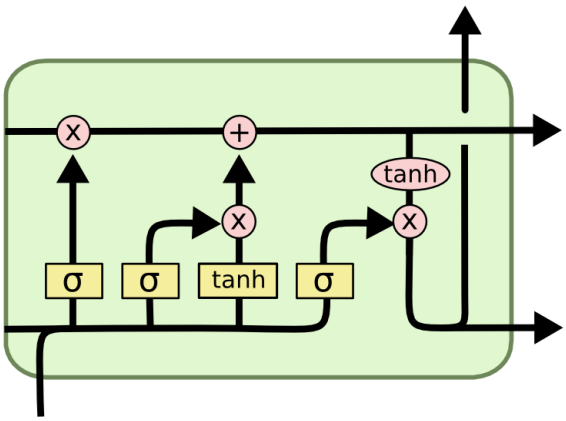

Figure 2.3: Standard LSTM Module.

Over time, the interest arose to investigate the flow of information in both directions, with this emerged a variant of LSTM capable of solving this same problem, called Bidirectional Long Short Term Memory (BI-LSTM) (Graves, Fernández, and Schmidhuber 2005; Minaee, Azimi, and Abdolrashidi 2019). BI-LSTM train two hidden layers on the input sequence. The first one on the input sequence as it is, and the second one on the reversed copy of the input sequence. This can provide additional context to the network, by looking at both past and future information, and results in faster and better learning. Figure 2.4 illustrates the high-level architecture of a BI-LSTM network.

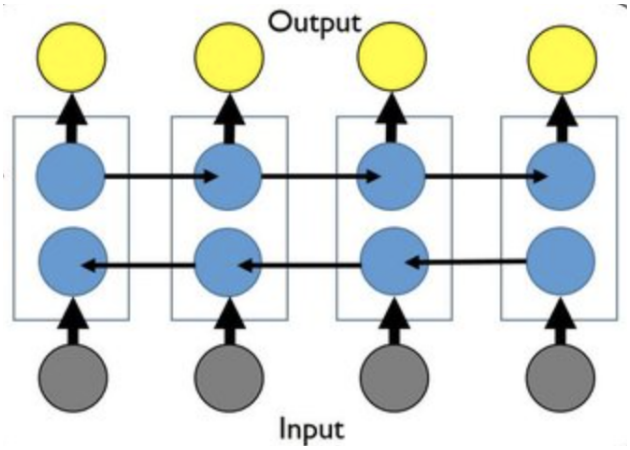

Figure 2.4: Standard BI-LSTM Architecture.

For this study, both models are used to compare their performance in classifying feelings in the text. Subsequently, an LSTM is used to classify bitcoin trends to demonstrate their efficiency in working with time series.

2.3 Convolutional Neural Network

CNN (Lecun et al. 1998) networks have been successful in various computer vision and NLP tasks recently. They are mighty in exploiting the local correlation and pattern of the data through learning through their feature maps (Minaee, Azimi, and Abdolrashidi 2019). (Kim 2014) showed great performance on several text classification tasks using CNN.

In order to classify text with CNN, the embedding of different words of a sentence is usually stacked together to form a two-dimensional array. Then convolution filters are applied to a window of h words to create a new representation of features. Then some pooling, usually max pooling, is applied to new features, and pooled features from different filters are concatenated to form the hidden representation. One fully connected layer then follows these representations to make the final prediction (Minaee, Azimi, and Abdolrashidi 2019). Figure 2.5 illustrates an high level overview of CNN network architecture.

To perform text classification with CNN, usually the embedding from different words of a sentence are stacked together to form a two-dimensional array, and then convolution filters are applied to a window of h words to produce a new feature representation. Then some pooling, usually max-pooling, is applied on new features, and the pooled features from different filters are concatenated with each other to form the hidden representation. These representations are then followed by one fully connected layer to make the final prediction (Minaee, Azimi, and Abdolrashidi 2019). Figure 2.5 illustrates a high level overview of CNN network architecture.

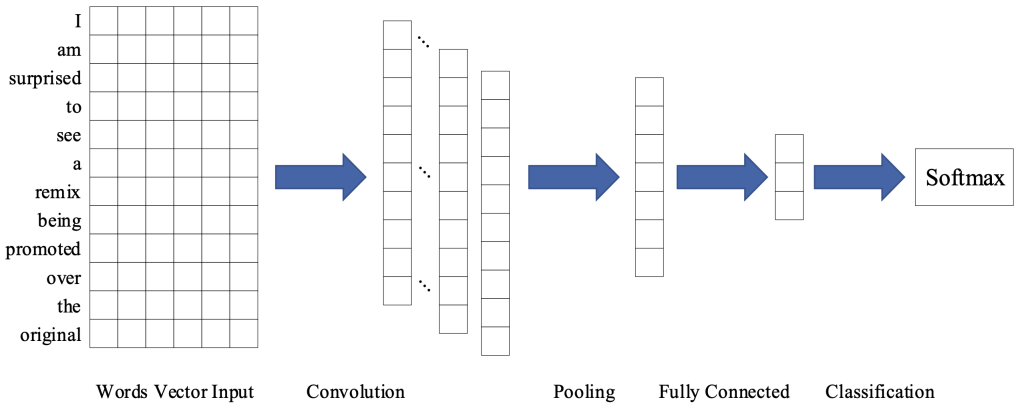

Figure 2.5: Standard CNN Architecture.

2.4 Twitter Sentiment Analysis

Sentiment Analysis is a task of NLP that aims to extract text sentiments (Chaturvedi et al. 2018). The task of sentiment analysis can be considered as a text classification problem (Choi and Lee 2017; Ji et al. 2013; Schuller, Mousa, and Vryniotis 2015) because the process includes several operations that classify whether a particular text expresses a positive, neutral or negative feeling. Sentiment analysis seems an easy task. However, it requires considering many NLP sub tasks like sarcasm and subjectivity detection (Valdivia et al. 2018). The text is not always organised as in books or newspapers (Birjali, Beni-Hssane, and Mohammed 2017; Erritali et al. 2016). It can contain many orthographic mistakes, idiomatic expressions, or abbreviations. The 280-character length limitation of Tweets makes them

extremely noisy data (Giachanou and Crestani 2016). Nowadays, researchers have widely acknowledged sentiment analysis (Ji et al. 2013), and internet growth has made the web the most important source of information, as millions of people express their opinions and feelings on social networks (Ramírez-Tinoco et al. 2018). (T. Li et al. 2017) says micro-blogs like Twitter can better provide a broad and global live stream of market information. Furthermore, micro-blogs spread generated content virally before news outlets report it and have an immediate market-changing impact on financial markets. Twitter data provides a rich source of information that can influence markets and extract emotional intelligence through sentiment analysis. Twitter posts, for example, were used to predict election results (O'connor et al. 2010). Since 2004, sentiment analysis has become the fastest growing and most active research area, as the number of papers focusing on sentiment analysis has recently increased dramatically (Mäntylä, Graziotin, and Kuutila 2018). According to Google Trends, figure 2.6 shows the rising popularity of sentiment analysis. In addition to this, the emergence of new technologies such as Big Data (Birjali, Beni-Hssane, and Erritali 2018; Yaqoob et al. 2016), Cloud Computing (Marston et al. 2011), and blockchain (Frizzo-Barker et al. 2020) has widened the area of applications providing for sentiment analysis unlimited possibilities to be applied in almost every domain.

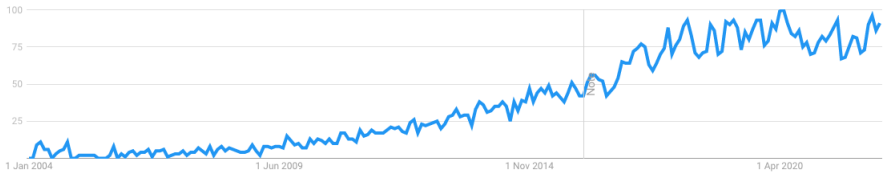

Figure 2.6: Interest in "Sentiment Analysis" since 2004 according to Google Trends.

The task of sentiment analysis was investigated on several levels. However, feelings can be detected mainly at the document, sentence or aspect level (Behdenna, Barigou, and Belalem 2016; Do et al. 2019). Figure 2.7 shows the sentiment analysis levels.

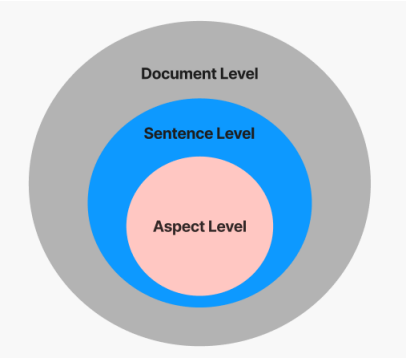

Figure 2.7: Sentiment Analysis Levels.

The aspect level performs fine-grained analysis because it aims to find sentiments concerning the specific aspects of entities. For example, consider the following sentence, "The camera of iPhone 13 is awesome." the review is on "camera" which is an aspect of the entity "iPhone 13", and the review is positive. Therefore, the task at this level helps to identify exactly what people like or do not like. It focuses on the aspects of entities like products or features

instead of discovering the sentiment of paragraphs or sentences. According to (Tubishat, Idris, and Abushariah 2018), aspect extraction is the core task for sentiment analysis. In this regard, the authors proposed a review of implicit aspect extraction techniques from different points of view. Moreover, many real-life applications require this level of detailed analysis.

The sentence level focus is on the sentence. The main goal is to determine whether the sentence expresses positive, negative, or neutral sentiments (Liu 2012). However, to achieve this goal, the sentence needs to be classified as objective, factual information, or subjective views and opinions. Several approaches tackled this level of analysis. (T. Chen et al. 2017) used a sentence type approach to improve the performance of sentence level sentiment analysis. They applied first a neural network-based sequence model to classify sentences into three types based on the number of targets included in a sentence (sentence with non target, one-target, or multi-target). For classification, they used a one-dimensional CNN, where each type of sentence is fed to the model separately. Sentiment analysis at both the sentence and document level is essential and valuable. However, it does not provide the necessary detail needed opinions on all aspects of the entity (Medhat, Hassan, and Korashy 2014), as they do not find precisely what people like or dislike.

Finally, the document level process aims to classify whether a whole document expresses a negative or positive sentiment (Alqaryouti et al. 2019). Each document is classified based on the overall sentiment of the opinion holder about a single entity (e.g., a single product). The classification at the document level works best when the document is written by one person and is not suitable for documents that evaluate or compare multiple entities. There have been many approaches proposed for document-level sentiment analysis. Sentiment analysis is instrumental for many application domains. However, sometimes the document may include opposite sentiments, which can impact the final decision.

For this study, the sentence level is used to determine whether a sentence shows positive, negative or neutral feelings. Several articles have been published in this area, demonstrating various techniques to achieve better performance when analysing text.

(Symeonidis, Effrosynidis, and Arampatzis 2018), compared different pre-processing techniques for tweets sentiment analysis. They classified tweets with four different ML algorithms, and tested 16 different pre-processing methods. They found that it is recommended to use lemmatization, to replace repeated punctuation, to replace contractions and remove numbers.

(Minaee, Azimi, and Abdolrashidi 2019), seek to improve the accuracy of sentiment analysis using an ensemble of CNN and LSTM networks and test them on popular sentiment analysis databases, such as the Internet Movie Database (IMDB) review and Stanford Sentiment Treebank v2 (SST2) datasets. The main objective of the CNN network is to extract information about the local structure of the data by applying multiple filters. In contrast, the LSTM network is better suited to extract the temporal correlation of the data dependencies in the text snippet. They show that by using ensemble model techniques, they were able to outperform the performance of both individual models. Ensemble models have been used for various problems in NLP and vision. They used word embedding based on a pre-trained LSTM model and a CNN with four filter sizes, each with 100 feature maps. Then, two fully connected layers were fed into a softmax classifier. Both models performed well, archiving good performance for sentiment analysis. LSTM achieves this mainly by looking at temporal

data information. While CNN achieves this by looking at the holistic view of local information in a text. They believe it is possible to boost performance further by combining the scores from both models.

2.5 Bitcoin's Trend Prediction

A blockchain is a system that acts as a trusted and reliable third party, not centralized, always online, to maintain a common state, mediate exchanges, and provide secure computations (Gramoli 2020; X. Li et al. 2020). Technically, it is a distributed register that stores transaction data grouped into blocks that constitute a growing and unalterable linked list. The register is managed by a large group of networked servers, each of which holds a copy of the entire blockchain. As blockchain grows, servers need to reach consensus on each new block to be included. A wallet is a software for making transactions and checking their validity. Bitcoin is the most widely used blockchain. An anonymous person introduced it in 2009 with the pseudonym of Satoshi Nakamoto (Nakamoto 2009). The Bitcoin software includes a transaction verification engine and connects to the network as a full node. Bitcoin coins are created at a predictable and decreasing rate, which means the demand must follow this level of inflation to keep the price stable. Many factors influence highly volatile bitcoin prices, including the supply of bitcoin and market demand for it, the mining process cost, the number of competing cryptocurrencies, and the number of transactions based on the Bitcoin platform. Blockchain technology accelerated when the bitcoin price reached its all-time high value of 19k$ in 2017. From that moment on, the bitcoin time series became an object of study for the research community. Since Bitcoin is a new phenomenon. The use of machine learning and deep learning with greater accuracy and speed with social media data is still new. Although there is much research on various machine learning techniques for forecasting time series, there is a lack of research directly related to Bitcoin in this area.

Several attempts have been made to predict early market movements of cryptocurrencies using tweet sentiments (Kraaijeveld and Smedt 2020; Lyu et al. 2021). Researchers are recognizing Twitter's power to predict a wide range of event, particularly for financial markets. Even the numbers of tweets correlates with Bitcoin's trading volume (Sattarov et al. 2020).

The volume of posts or messages also correlates with Bitcoin's trading volume (Mai et al. 2015). In addition, (Karalevicius 2018) confirm what was suggested earlier; cryptocurrency investors appear to overreact to news leading to a price pattern where the price initially moves with the sentiment and is then slightly corrected.

Twitter sentiment analysis has been used in various studies to predict Bitcoin's price fluctuations. In a study by (Georgoula et al. 2015), a Support Vector Machine (SVM) and various regression models were used to predict Bitcoin's price fluctuations using Twitter sentiment analysis. The authors obtained an accuracy of 89.6% and only found a short-term correlation between positive Twitter sentiment and Bitcoin's price.

Some researchers investigated the impact of news sentiment on Bitcoin and traditional currency returns, volume, and volatility (Rognone, Hyde, and S. S. Zhang 2020). Where a high-frequency intra-day data (15 min) for a dataset with seven years of data was analyzed to identify the sentiment of non-scheduled news around Bitcoin and six traditional currencies. The authors found that traditional currencies react immediately and significantly to news wire messages coming from the economy. For bitcoin, the results were different from those on traditional markets which means that Bitcoin does not react similarly to news arrivals as traditional currencies (Rognone, Hyde, and S. S. Zhang 2020).

The studies that apply sentiment analysis to the field of blockchain technology still scarce and the existing work generally use sentiment analysis to forecast digital currencies value as in the work of (Kraaijeveld and Smedt 2020). The authors used a cryptocurrency-specific lexicon-based approach to perform tweeter sentiment analysis in order to predict the price returns of some well-known cryptocurrencies. (Jing and Murugesan 2019) proposed a theoretical framework to detect fake news automatically on social media using the principals and methods of blockchain technology. Although the effectiveness and the performance of this framework need to be validated, it promises that a combination of sentiment analysis and blockchain technology can be useful.

The use of strengh and polarization of opinions displayed on Twitter have been considered to predict Bitcoin's trend (Garcia and Schweitzer 2015). They show that an increase in the polarization of sentiment anticipates a rise in the price of Bitcoin.

(Huang et al. 2021) proposed a RNN with LSTM by utilizing the sentiment analysis of social media to predict the real time price movement of the digital currency. (Pimprikar, Ramachandran, and Senthilkumar 2017), found the LSTM combined with a Twitter sentiment analysis outperforms other machine learning models such as SVM in predicting the stock price.

(Kraaijeveld and Smedt 2020) argue that the cryptocurrency market is driven by news disseminated via social media, such as Twitter, as traditional media lacks coverage of this newly emerged asset class. Utilising a lexicon sentiment analysis approach to measure investor sentiment, they find that investor sentiment measured using tweets has predictive power on bitcoin returns.

(McNally, Roche, and Caton 2018) tried to predict with the highest possible accuracy, achieving 52% and a Root Mean Square Error (RMSE) of 8%, the directions of Bitcoin prices in USD using machine learning algorithms like LSTM and RNN.

(Matta, Lunesu, and Marchesi 2015) use 'SentiStrength', a lexical-based sentiment analysis approach, to measure sentiment using tweets. They find that positive tweets predict bitcoin price movements over the sample period of January 2015 to March 2015.

(Abraham, Dowling, and Florentine 2018) find that tweet volume and Google Trends from March 2018 to June 2018 predict price changes of bitcoin. Nonetheless, they find that tweets sentiment obtained from the VADER sentiment analysis method fails to predict price changes of bitcoin.

(Xu and Keselj 2019) investigation, showed using Twitter mood to predict stock market did improved enhancement compared with non-sentiment approaches. (Schumaker and H. Chen 2009) also refers classic methods are mostly based on feature engineering. With Deep Neural Network (DNN) drawing much more attention in past years, CNN based method and LSTM based models were able to use larger datasets from text and history stock price to produce better outcomes.

(Mao, Counts, and Bollen 2011) show that although traditional investor sentiment does not have predictive power for financial markets, Twitter sentiment is able to have strong predictive power for the next 1–2 day(s) returns. (Bollen, Mao, and Zeng 2011; T. Li et al. 2017; Sprenger and Welpe 2010; X. Zhang, Fuehres, and Gloor 2011), also confirm that the predictive power of Twitter sentiment for financial markets is generally observed to be the strongest between 1–4 days.

(Kaminski 2014) studied correlations and causalities between Bitcoin market indicators and Twitter posts. The considered dataset spans 104 days (from November 2013 to March 2014) and contains 161 200 tweets, as well as different features extracted from different exchanges, such as BitStamp, Bitfinex, BTC-e and BTC China. The results of the data analysis led the author to the interpretation that emotional sentiments rather mirror the market than that they make it predictable. However, the considered timeframe is too short and characterized by an unusual bitcoin trend (exponential growth) to draw any general conclusion.

(Matta, Lunesu, and Marchesi 2015) analyzed the bitcoin trend using Google Trends data and 1924891 tweets, in a timeframe of 60 days (from January to March 2015). They observed a correlation between the bitcoin price and the tweets that express a positive sentiment. Remarkably, the tweets appear to anticipate by 3–4 days the bitcoin trend. Correlation results between the bitcoin price and Google Trends data are less convincing. Google Trends data are not easy to handle, as they are always normalized with respect to the considered timeframe, such that the period with the highest relative search intensity corresponds to an arbitrary reference value set to 100.

(Stenqvist and Lönnö 2017) considered a 31-days timeframe from May to June 2017, in which they collected 2 271 815 tweets. Their sentiment analysis was carried out by means of a powerful tool, namely VADER (Hutto and Gilbert 2014). A careful cleaning process allowed the authors to exclude more than 50% of the tweets, i.e., those produced by bots and those carrying duplicated content. Despite the sound approach, the resulting accuracy of the predicted bitcoin value shows too much variability, depending on the chosen timeframe.

(Madan, Saluja, and Zhao 2014) proposed a bitcoin forecasting approach based on machine learning algorithms. In particular, they predicted the sign of the future change in price using a binomial Generalized Linear Model (GLM), leveraging both SVM and random forest. The considered dataset has 26 features relating to the bitcoin price and payment network over the course of five years (from 2009 to 2014). The proposed solution achieves 50%–55% accuracy in predicting the sign of future price change using 10 min time intervals. The same result was obtained by (Greaves and Au 2015), with a reduced timeframe for the dataset (1 year) and 12 features. In this case, the authors solved the prediction problem by means of a feed-forward NN with two hidden layers.

Using a GPU-enhanced deep learning approach, with a LSTM network, (McNally, Roche, and Caton 2018) achieved a 52% accuracy. The dataset takes into account the bitcoin value, the hash power, the mining difficulty and other information extracted from the blockchain. For the first time, a financial index was used, namely the Simple Moving Average (SMA). The considered timeframe spans 3 years (from August 2013 to July 2016).

(Mittal et al. 2019) studied the correlation between bitcoin price, Twitter and Google search patterns. Using different machine learning techniques, they concluded that there is a relevant degree of correlation of Google Trends and Tweet volume data with the bitcoin price, and no significant relation with the sentiments of tweets. In particular, the authors achieved a 62.4% accuracy in predicting bitcoin price fluctuations based on Google Trends and Tweet Volume using a RNN model.

(Linardatos and Kotsiantis 2020) analyzed 7M tweets, Google Trends data, the bitcoin price and other features, over a 2-years timeframe (from January 2017 to December 2018). They used VADER (Hutto and Gilbert 2014) for the sentiment analysis of the tweets, and an LSTM network for the prediction task. The resulting accuracy was 52%.

(Cavalli and Amoretti 2021) developed a bitcoin trend prediction implemented on a cloud-based system characterized by a highly efficient distributed architecture. They extracted specific data from CoinMarketCap, Twitter and the Bitcoin blockchain, respectively. The data was collected between April 2013 to February 2020. They compared the accuracy between a CNN model against an LSTM model. The results accuracy shows CNN have better accuracy that LSTM. They were able to archieve 74.24% using CNN and 54.31% using LSTM.

(Georgoula et al. 2015) utilise a machine learning approach to perform sentiment analysis on Twitter data from October 2014 to January 2015. They find that the Twitter sentiment ratio for bitcoin has a positive short-run impact on bitcoin prices.

(Huang et al. 2021) use LSTM as the NN learning layer and combine it with the sentiment analysis method to develop a cryptocurrency sentiment analyzer that can predict the price movement of cryptocurrency. In the pre-processing phase, the first tokenized each social media post according to the cryptocurrency vocabulary and then fed it into an embedding layer, thus converting the word token into the cryptocurrency word embedding. All post labels used in training were manually labelled and encoded with positive, neutral and negative. Then a RNN were trained by taking the embedding feature vector sequence. They used a fully connected layer to transform the LSTM output and then used a sigmoid function to output the prediction. They used precision and recall metrics to measure the model performance in predicting sentiment. The precision measures the model ability to return only relevant instances. In contrast, the recall measures the model ability to classify all relevant instances. Finally, they compared their method with the time series autoregression approach to evaluate the model performance, and they found that the LSTM approach outperforms over 18% in precision and over 15% on recall. They prove the effectiveness and power of the LSTM in predicting sentiment analysis on social media content.

(Xu and Keselj 2019) built a dataset with tweets sentiment and technical indicators, then tested an attention-based LSTM model to predict future stock price movements. Their goal was to study the attention-based LSTM variant and test the combination of tweets sentiment with a technical stock indicator to verify if a modified LSTM could gain better performance than traditional approaches. They studied how posted tweets could affect or impact the stock rise and fall prediction in the next trading days. They found that tweets posted during intraday, after hours, and the entire day could directly affect the model's performance. For the time period of finance tweets, they defined three categories: full day, intraday and after hours. Intraday tweets refer to tweets posted during the trading hours. After-market tweets refer to the tweets that are posted from market closes till before market opens in the next trading day. Full-day tweets are tweets that are posted in the past 24 hours before the market closes on a target trading day. They mentioned that RNN performance tends to decrease when the input sequence increases, and the attention model could maintain a good performance. The use of the attention layer can review the input sequence and extract useful information that has more connection to the target. For the data collection they used StockTwists, a platform to acquire data from Twitter, and collected finance tweets between 2016 and 2018. To evaluate model's performance, they adopted a standard accuracy measure and Matthews Correlation Coefficient (MCC). MCC is used to measure the quality of binary classifications. As outcome, the use of attention based LSTM model improves over traditional LSTM on aggregated datasets.

(Georgoula et al. 2015) studied the dynamics governing the formation of Bitcoin prices by focusing on Twitter sentiment as an explanatory factor, along with other economic and

technological variables. They collected over 2 million tweets during 78 days. The dataset includes eleven variables, like Bitcoin's historical data, the daily number of tweets, the daily sentiment ratio associated with Twitter posts, and the daily number of Bitcoin searches on Google and Wikipedia. They found that other variables, such as the number of searches on Wikipedia and the hash rate of Bitcoin, positively impact the price of Bitcoin. In contrast, the impact of the exchange rate between the USD and the euro has a negative impact. The sentiment ratio of Twitter users has a positive effect on the price of Bitcoins. They also studied the increase in the stock of Bitcoins, which led to an increase in the Bitcoin price. In contrast, an increase in the % Standard and Poor's 500 stock market index negatively affects the price of Bitcoins in the long run. This also reflects that investments in traditional stocks and Bitcoins are treated as substitutes. For the model development, they used linear regression techniques. However, since the dataset was not large enough, they cannot conclude if the reached conclusions remain valid or not.

(Pant et al. 2018) used a RNN model along with the Bitcoin's historical price and sentiment, extracted from over 4000 tweets, to predict new price for the next time frame. They used different technique to correlate tweets sentiment score with Bitcoin's historical price to predict future Bitcoin price. For this they built a sentiment analyser that gives a daily percentage sentiment which they feed into a RNN predictor along with the historical Bitcoin's price. The output of this RNN model is the future price prediction. From the collected tweets they removed irrelevant tweets like promotional and advertising using FuzzyWuzzy method, then processed to word tokenization, filtered out stop words, they removed hyperlinks and emojis. Named-entity recognition (NER) along with Regex is used to extract the names of persons, organizations and country present in tweet and it is later used for giving double weight to its sentiment if the extracted names are listed in impactful groups index. The RNN network is based on LSTM and Gated Recurrent Unit (GRU) techniques to predict future prices. They obtained an accuracy of 81.39% for tweets sentiment classification and 77.62% accuracy using RNN for overall bitcoin price prediction Finally they found that Word2Vector does not perform so well as Bag of word technique, and that there's a moderate correlation between Bitcoin's price and social sentiment when there is a rise of negative sentiment and consequent fall in Bitcoin's price but related with the increase of positive sentiment there's a strong increase in price.

(Xu and Keselj 2019) used Open Close High Low (OCHL) data, collective sentiment and technical indicators to feed the NN. The classification model applied is based on the LSTM and attention mechanism. The goal was to predict the direction of stock price movement for the next trading day. They attempted to predict whether the target stock would rise or fall on the next trading day. They trained the model on historical data from 2017 to 2018. In addition, daily stock price data was collected from Yahoo Finance. The result shows excellent potential to use financial tweet sentiment and technical indicators compared to non-sentiment and non-technical datasets. They found that the tweets posted from the closing of the market until the opening of the market the next day have more predictive power over the stock movement the next day. Their model was able to archive 65% accuracy.

(Sattarov et al. 2020) used Random Forest Regression binary classification model with diverse inputs and evaluated the model output. They used sentiment analyzing score and history price of Bitcoin as an input data and implemented a random forest algorithm by using Random Forest Classifier from sklearn.ensemble provided by scikit-learn. They experimented 10 different estimators using both presence and frequency features. Presence features performed better than frequency though the improvement was not substantial.

Their findings confirms the presence of a correlation between them. They observed 62.48% accuracy when making predictions based on bitcoins-related tweet sentiment and historical bitcoin price. They found that Random Forest Regression is quite effective with working a different kind of inputs that has not relationships with each other. The algorithm has advantages in predicting future outputs as well. The use of Bitcoin sentiment lexicon could help to improve their model. As well taking in consideration other features like Twitter users, tweets volume and emotions could help correlating sentiment with Bitcoin price.

(Matta, Lunesu, and Marchesi 2015) result seems to confirm that volumes of exchanged tweets may predict the fluctuations of Bitcoin's price. In order to compare tweet sentiment with Bitcoin's price they calculated the cross-correlation between them, and they found that tweets volume is related to price with a maximum cross correlation value of 0.15 at a lag of 1 day. They were able to see that there are peaks in tweets trend that precede peaks in price, suggesting a relationship between the two time series. They also analyzed tweets with positive mood and they noticed a two-fold increase in cross-correlation value. And confirmed that positive mood could predict the Bitcoin's price almost 3-4 days in advance. All patent peaks in the positive tweets plot precede a significant change in the Bitcoin's price after some days. Applying cross correlation between Google Trend data and Bitcoin's price also looks significant. This result is shown also by a little significant relationship that exists between positive tweets and Google Trends data.

(Bollen, Mao, and Zeng 2011) demonstrated that tweets can predict the market trend 3-4 days in advance, with a good chance of success. They analyzed the Bitcoin price's behavior comparing its variations with the number of tweets, with the number of tweets with positive mood, and with Google Trends results.

Chapter 3

Methods

3.1 Technology & Tools

3.1.1 Hardware & Software

For the development of the present study, hardware was used, specifically, a fixed computer equipped with an Intel Core i9 Central Processing Unit (CPU), 32Gigabyte (GB) of Random Access Memory (RAM), an M.2 Solid-State Driver (SSD) disk with a storage capacity of 500GB and a 3080 TI Graphics Processing Unit (GPU) with 12GB from memory. This equipment allows to achieve good data processing and simultaneously allows the training of neural networks faster and more efficiently.

Regarding software, Ubuntu 20.0 was used and freely available with community and professional support. The selection of Ubuntu is based on the fact that it is recognized as a complete operating system, given its typology and functioning, namely, being predictable, stable and secure. In addition, this operating system can expose the machine as a server so it can be accessed remotely and trained at any time, as long as it is connected to the Internet. To convert this machine into a server, it was initially necessary to install the OpenSSH server, facilitating the remote connection of other clients to this local machine.

It is important to mention that the described hardware and software are fundamentally used to train the neural networks developed in this study to obtain a machine with greater potential and computational power characteristics. Despite its capabilities, another machine was also used to create the code produced during the development of this study and to communicate remotely with the first machine described above. This equipment is a Mackbook Pro 2019, equipped with a 6-Core Intel Core i7 processor, 16GB of RAM, a 250GB disk and an Intel UHD Graphics 630 with 1536Megabyte (MB) of memory.

3.1.2 Development Environment

For the development of the models present in this study, the Python language version, 3.9 was used, given its ease of learning, understanding, code maintenance, structuring and interaction with libraries. In detail, Python is the high-level programming language created by Guido Van Rosum in 1991. This language is developed under an OSI-approved open source license, making it freely usable and distributable for commercial use. It is also worth noting that its language constructs and object-oriented approach aim to help programmers write logical code, integrated into small and large-scale projects.

Despite its advantages, managing Python dependencies and controlling their versions becomes crucial. The Virtual Environment was used in an initial phase to solve this problem.

However, due to its complexity in managing some dependencies, it was necessary to use Conda. One of the features that Conda has and was particularly useful is the ability to manage and resolve versions when installing new dependencies that it was impossible to verify in an easy and attainable way using the virtual environment.

Conda is an open-source package and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on a local computer. It was created for Python programs but can package and distribute software for any language. Due to its management capacity, it helps find and install packages. For example, let us say there is a need for a package that requires a different version of Python. In that case, it does not need to change to a different environment management system since Conda also allows to manage the environment. With a few commands, it is possible to set up an entirely different environment to run a different version of Python.

After selecting the language and the environment management, we defined the Integrated Development Environment (IDE) to be used to develop all the code. Considering its advantages and features of an intuitive and easy-to-manage interface, the IDE selected for this study was Jupyter Notebook.

Jupyter Notebook is a classic notebook interface widely used in AI. Specifically, it is the original web application for creating and sharing computational documents. This instrument offers a simple, streamlined and document-centric experience, supporting over 40 programming languages. Notebooks can be shared with others in various ways, such as email, Dropbox, Github and the Jupyter Notebook Viewer. In addition, the code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types. JupyterLab, the next-generation notebook interface, is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and ML.

For this study's development, these were the main tools used for the development environment. This makes the development faster and more stable, allowing to easly manage dependencies if needed.

3.1.3 Libraries

TensorFlow is an end-to-end open source platform for ML. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. It is easy model building, TensorFlow offers multiple levels of abstraction. Build and train models by using the high-level Keras Application Programming Interface (API), which makes getting started with TensorFlow and ML easy. TensorFlow provides a direct path to production. Whether it is on servers, edge devices, or the web, TensorFlow lets train and deploy models easily. Allow to build and train state-of-the-art models without sacrificing speed or performance. TensorFlow gives the flexibility and control with features like the Keras Functional API and Model Subclassing API for creation of complex topologies. TensorFlow also supports an ecosystem of powerful add-on libraries and models to experiment with.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent and simple API, it minimizes the number of user actions required for common use cases, and it provides clear and actionable error messages. It also has extensive documentation and developer guides. Keras is the most

used Deep Learning (DL) framework among top-5 winning teams on Kaggle. Built on top of TensorFlow 2, Keras is an industry-strength framework that can scale to large clusters of GPU or an entire Tensor Processing Unit (TPU) pod. They have advantage of the full deployment capabilities of the TensorFlow platform. Keras is also a central part of the tightly-connected TensorFlow 2 ecosystem, covering every step of the ML workflow, from data management to hyperparameter training to deployment solutions. Because of its ease-of-use and focus on user experience, Keras is the DL solution of choice for many university courses. It is widely recommended as one of the best ways to learn DL.

Scikit-learn is a Python module for ML built on top of SciPy and is distributed under the 3-Clause BSD license. Is an indispensable part of the Python ML toolkit. It is very widely used across all parts of the bank for classification, predictive analytics, and very many other ML tasks. Its straightforward API, its breadth of algorithms, and the quality of its documentation combine to make scikit-learn simultaneously very approachable and very powerful. Also provides a toolbox with solid implementations of a bunch of state-of-the-art models and makes it easy to plug them into existing applications.

Finally Natural Language Toolkit (NLTK), a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

3.2 Safety & Ethics

3.2.1 Public vs Private Spaces

In the world of the internet, social communication platforms have shown an active role in online interaction on an ongoing basis. However, despite its importance, it has been found in the literature that there is no consensus on the distinction between public and private spaces. Thus, given the ethical and privacy implications, it is essential to clarify these concepts for the present study (Ahmed, Bath, and Demartini 2017).

According to the British Psychological Society, much of Internet communication takes place simultaneously in a private place, for example, at home, and in public, for example, in an open discussion forum. However, in this medium, it is difficult to determine which spaces people identify as "private" or "public" accessible (Ahmed, Bath, and Demartini 2017).

Some social media platforms are considered inherently private spaces, such as Facebook. Others, on the other hand, are seen as public spaces for online communication, such as Twitter. In this sense, the difference between these platforms is that most of the content shared on Twitter is publicly accessible through the Twitter API and data resellers. At the same time, on Facebook, the data is only available at an aggregated level. On the other hand, Twitter profiles and tweets are configured for public visibility. Thus Twitter can be seen as more of a public space than other social platforms. However, not all Twitter users know that all their posts are public or available for review and scrutiny (Ahmed, Bath, and Demartini 2017).

In general, Twitter has become a platform with more popularity in academic research, taking into account the ease of accessing data, its collection, classification and expansion. Therefore, Twitter data is more accessible to retrieve, as significant incidents, news and events on Twitter tend to be hashtag-centric (Ahmed, Bath, and Demartini 2017).

3.2.2 Terms of Service & Privacy

Twitter was designed to spread information that users share publicly widely and instantly. In this sense, Twitter has created its Terms of Service and Privacy. These essential documents govern what users can access and use on the platform to make informed decisions and ensure understanding and control of the information they collect, how they are used and when they are shared (Ahmed, Bath, and Demartini 2017).

By accepting Twitter's Terms and Service Agreement, users consent to their information being collected and used by third parties (Twitter 2022b).

In accordance with the Privacy Policy established by *"You are responsible for your Tweets and other information you provide through our services, and you should think carefully about what you make public, especially if it is sensitive information."* (Twitter 2022a).

In addition, the Terms of Service states that *"You agree that this license includes the right for Twitter to provide, promote, and improve the Services and to make Content submitted to or through the Services available to other companies, organizations or individuals for the syndication, broadcast, distribution, Retweet, promotion or publication of such Content on other media and services, subject to our terms and conditions for such Content use."* (Twitter 2022b).

In this sequence, when publicly sharing content, the user instructs the dissemination of this information as widely as possible, including through the API, and directs those who access the information to do the same. To facilitate the rapid global spread of tweets worldwide, technologies such as API and Twitter have been incorporated to make some of this information available to websites, applications and news sites. In these situations, Twitter has standard terms that govern how that data can be used and a compliance program to enforce those terms. One justification often provided by scientists with Twitter data on the ethical and legal implications of using data without informed consent is that the reuse of data is permitted by Twitter's terms and services and privacy policies.

However, it is important to note that discarding tweets or downloading tweets from Twitter's Advanced Search will violate Twitter's Terms and Conditions and void any protection these policies may provide. In addition, this procedure would bypass data retrieval from the Twitter API and allow Twitter to see who retrieved data from the platform. As a result, Twitter expressly discourages this practice *"...scraping the Services without the prior consent of Twitter is expressly prohibited"* (Twitter 2022b). Reproducing but removing IDs or altering tweets will contravene Twitter's User Development Policy, which requires tweets to be published in full. The Twitter platform not only controlls the access to data, but also dictate how results of research projects are presented. For that reason, there is a definitive need for researchers to engage with Twitter company for academic use of data.

3.2.3 Academic Perspective

In an academic setting, it is widely considered a cornerstone of research integrity and research quality to have considered the ethical implications of research, especially within the fields of social research.

Research may need to pass through a research ethics committee, whose role it is to protect research participants from potential harm, institutions from potential negative attention and reputational risk, as well as the researchers themselves.

All of these principles apply to social media research because, essentially, the majority of content on online spaces such as Twitter is created by people, with the exception of organisational, news, and automated Twitter accounts.

This research project, took the ethical standpoint of not quoting tweets or disclosing non-public usernames, unless with the permission of the user. The main reason of taking this decision is that those users, although they may be doing in a public space, may not be aware that their tweets are being used for academic research. Although Twitter's term and condition states that user data may be redistributed or used for other purposes. A 2017 Deloitte survey of 2,000 consumers found that 91% of people consent to legal terms and services conditions without reading them. For younger people, ages 18-34 the rate is even higher, with 97% agreeing to conditions before reading (Deloitte 2017).

Twitter developer portal, enables to submit projects for researches purposes and allows to use public data. This project is registered and approved by Twitter Developer Portal as a valid project for research with a licence for non-commercial use. This account allow to make use of Twitter's data, were is used to extract Bitcoin's related tweets. Complaining with all the legal requirements imposed by Twitter.

In this study, the researcher is the only one with access to the extracted data from Twitter. The analysis of the data was conducted by the researcher and would take place in the researcher's place of study and home. The data is not analysed in places deemed as public and would be stored on a two password protected laptop. The ethics application also noted that certain might need to be shared with the supervisor and co-supervisor of this project for administrative use.

3.3 Architecture

This section describes all the steps for realizing and idealizing the architecture developed in this study. The architecture intends to be a means of visualization to guide all the steps of this study. The architecture is divided into two sub-architectures that aim to focus on the steps necessary to carry out the construction of sentiment analysis 3.4 and Bitcoin trend forecasting models 3.5. Implementing a higher-level architecture is necessary to better visualise all components of the construction of these two models and understand how they were interconnected and what parts they have in common.

In Figure 3.1, it is possible to visualize the architecture at the highest level, where it is possible to see the sub-architecture of the sentiment analysis model on the left and on the right, the sub-architecture related to the Bitcoin trend forecasting model.

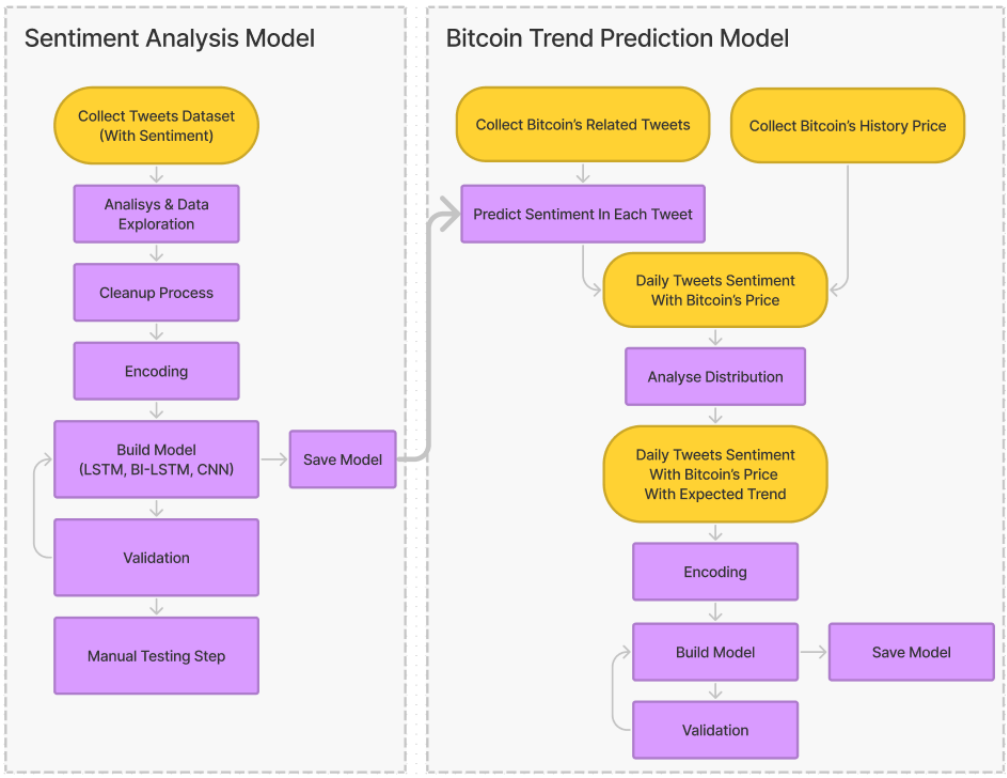

Figure 3.1: Overall Implementation Architecture.

The figure above represents all the steps taken during the research and planning phase and how the two models intertwine to achieve an ultimate goal. This architecture was designed and thought to evolve the models independently, allowing it to optimize the sentiment analysis model without affecting the performance of the Bitcoin trend forecasting model and vice versa. In this way, it is possible to collect more data over time, improve the performance of both models individually, and reach a better end goal.

Analyzing the sub-architecture represented on the left of figure 3.1, it is possible to visualize all the steps necessary to carry out the sentiment analysis model. Likewise, in figure 3.2, it is possible to envision all phases of developing these first AI models.

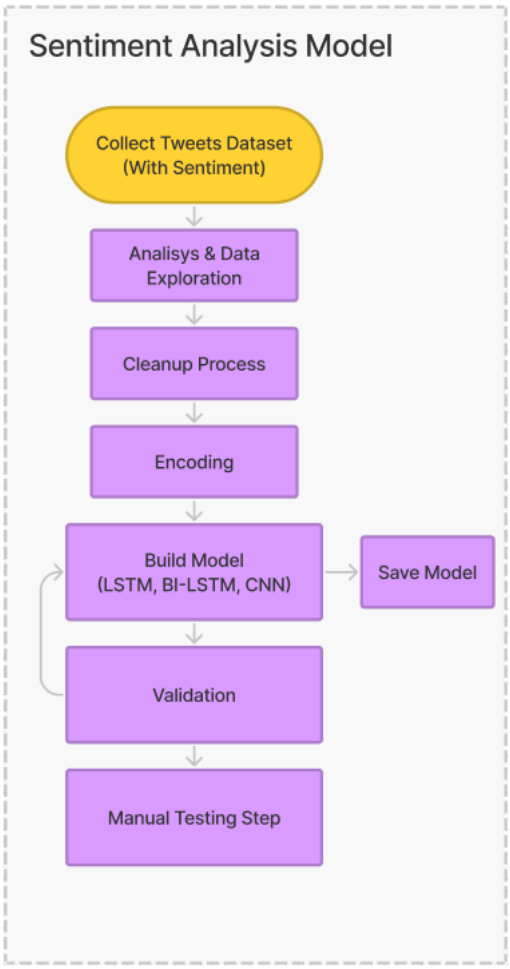

Figure 3.2: Sentiment Analysis Model Architecture.

Initially, the focus is on collecting categorized tweets with the respective sentiment expressed in them. Their collection is detailed in the 3.4.1 section. An analysis and exploration phase is carried out to visualize possible correlations and what data are important for this model. After this analysis, the data cleanup process is carried out, and finally, an encoding phase, since the model needs these data in a numerical format, and the tweets are presented as textual data. Then the focus is on building the AI model that will go through an interactive training and validation phase until obtaining the desired results. In this training and evaluation phase, the model will change, the so called tuning process, until it can optimize this model to the maximum. After each training, an evaluation will be carried out to analyze the model's performance. During this training phase, the model with the best performance is also persisted so that, in the end, it is possible to persist it on disk or in a database for future use. At the end of the training phase, a code block is still added, where it is possible to manually test and interact with the model quickly to obtain faster feedback on the model's performance and what can be optimized.

Analysing the second AI model focused on Bitcoin trend forecasting represented in the figure 3.3, it follows the same structure of the sentiment analysis model mentioned above 3.2 but with a few more specifics.

This architecture starts with constructing a dataset, which will be a junction of two datasets.

The first one focused on daily tweets that will be categorized with the sentiment analysis model mentioned above in the 3.2 architecture, which will then be joined with a second dataset of Bitcoin historical prices, thus giving rise to the final dataset containing the daily sentiment with bitcoin price for the respective day. The construction of this dataset is explained in detail in the section of the 3.5.1. Then, an analysis phase of the dataset is carried out to understand the distribution of trends over time. With this, a new column will be added with the expected trend for the next day. Then a data encoding phase is carried out, which is later sent to the model. This encoding phase is again necessary to ensure that all data is sent in numeric format and at the same scale. Finally comes the model construction, which contains an interactive phase of training and validation, where the tuning phase is carried out until reaching the desired performance. At the same time that this is performed, a new model version is interactively saved during the training phase if it performs better than the previous one.

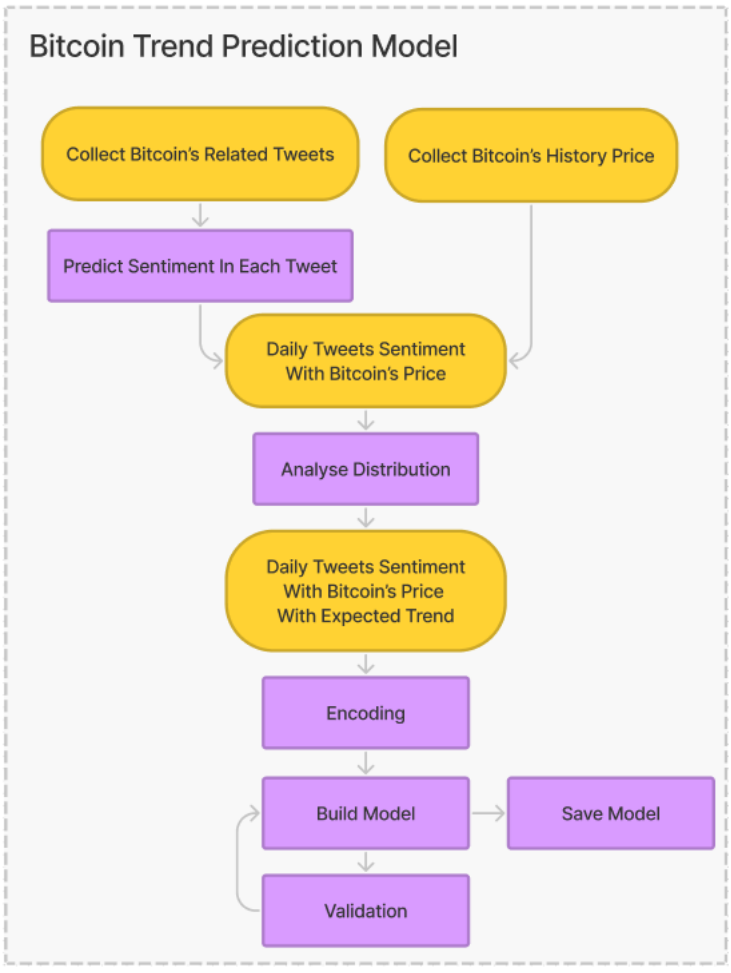

Figure 3.3: Bitcoin Trend Forecasting Architecture.

In conclusion, the architectures followed for the development of this thesis were detailed. It is possible to analyze the construction of the two models individually and see how they are interconnected. This phase is crucial to know which parts are connected to which, and it is vital to keep the separation between models to avoid increasing their complexity. The separation of the models has already been designed to work individually on them, without having to change the architecture every time it was necessary to modify one of the models.

Since the sentiment analysis model deals with more data and data that take up more space, this separation made perfect sense, optimizing the time spent on each model.

3.4 Sentiment Analysis Model

This section describes all phases carried out for the construction of the sentiment analysis model. It is intended to describe all the steps taken to build a dataset of categorized tweets with the respective sentiment, analysis and exploration of the same dataset, exciting metrics related to tweets, the construction of three different AI models from RNN to the use of a CNN network, the explanation of the tuning process carried out, and finally a demonstration of the results obtained in the evaluation and validation.

3.4.1 Dataset

This subsection will describe the constitution of the first dataset used to train different AI models to perform sentiment classification in tweets. In this way, two main options emerged to carry out the same collection. A first solution would be to manually collect tweets that contain hashtags related to bitcoin, where it would later be necessary to carry out a manual classification process. This option would not be feasible due to the time required to carry out this study. However, as a second option, an internet search could be carried out on datasets already classified and highly rated in terms of quality. It was decided to continue with the second option. After a search, several datasets were found already classified with the classifications intended for this study. The datasets found are not directed to the bitcoin topic. However, despite containing general topics, it is a large dataset that will give the model room to train based on more text variety. This prevents the model from learning too much about people already in the cryptocurrency world and not about new people just starting. This way, it is possible to better detect trends when many new people join cryptocurrencies. Since tweets are small texts and people express themselves differently, it was decided to mix Bitcoin's related tweets with non Bitcoin's related tweets to get a more diversified dataset that does not focus only on Bitcoin, but on the opinions of the general public. The dataset in question was collected from the Kaggle (Kaggle 2022) platform, a well-known and recognized platform for AI and researchers. Kaggle enables data scientists and other developers to engage in running ML contests, write and share code, and to host datasets. Which made it possible to find several datasets in which one related to sentiment analysis. After analyzing the dataset, it was in the structure represented in 3.1, which was in the desired format for the implementation of this model. In the JSON 3.1, it is possible to see that the dataset has only the properties related to the respective tweet text and the classified sentiment, where the positive sentiment is ranked as 1, the negative sentiment as 2 and the neutral sentiment as 0. Then the quality of the dataset was analyzed through the Kaggle platform, where it was possible to see that it was well rated despite showing an imbalance.

```
{
  "text": "Tweet content",
  "sentiment": 0
}
```

Listing 3.1: Sentiment Analysis Dataset JSON Representation.

After obtaining the data in a Comma-Separated Values (CSV) format, a dataset with a dimension of 860452 tweets was obtained, in which 309056 tweets are classified with positive sentiment, 266239 tweets with neutral sentiment and 285157 tweets with negative sentiment. With this, we obtained a dataset of considerable size that will be used to train the NN described below. One of the big challenges in collecting this dataset was not the quantity but the quality of the data that is exposed on the internet on platforms like Kaggle. Since it is millions of tweets, it would not be possible in useful time to categorize them correctly in order to obtain a quality dataset. Thus assuming a degradation on the part of the model, due to the use of tweets collected on the internet without having a second manual validation.

3.4.2 Data Exploration

This section details the steps to explore the data from the dataset described above. This phase is critical and crucial for the performance of the model. First, the data must be well analyzed to understand which information is important and which should be removed. With this, some decisions were taken and will be described below.

As a first analysis, an analysis of the balance of the dataset was carried out to understand how many feelings there are for each of the categories. This is important, because it directly impacts the performance of the model. With this, the distribution of feelings in the present dataset was analyzed, represented in the figure 3.4, where it was validated that it was imbalanced. It contained 285157 negative tweets, 309056 positive tweets and 266239 neutral ones, which in the future would make our model tend more towards the categories that contain more tweets.

Figure 3.4: Dataset Sentiment Distribution.

There are three possibilities to solve this situation. The first is to search for more tweets to increase the categories with fewer tweets to equalize the distribution between feelings. The second solution consists of generating data based on existing data, and finally, a third solution, in which tweets are removed to equalize the distribution of tweets across the three categories.

The first solution would be the most appropriate. However, the difficulty of finding more publicly categorised tweets made it difficult to implement this solution, which was eventually discarded. The second solution is not ideal, the goal is to obtain a diversity of tweets, and if more tweets were generated based on existing tweets, this diversity would not be achieved, and the model would focus a lot on the most repetitive tweets. The third solution proved to be the best solution to this problem, where it is unnecessary to search for more tweets or even pay for them. Given the size of the dataset, it was assumed that some data was lost to balance the data. The final solution was to remove tweets until each of the three categories

equalled 260000 tweets. With this, a large dataset is maintained, totalling 780000 tweets and maintaining much diversity between tweets.

Followed, the size of the tweets was analysed, which proved to have randoms sizes. There are small tweets that are only one word. Twitter allows only tweets of 280 characters, which means most tweets are medium-sized. For this specific case, no decision was made regarding the dimensions of the tweets sice most of them are small to medium size.

Then the content of the tweets was analysed, which words they contained and which other types of characters are included in the tweets. As a result, several words and characters have been identified that make no sense to include in the model training, such as hyperlinks, symbols and pictographs, flags, emails, characters that represent lines, single quotes, signs, emojis, hashtags, html tags. These words or characters can be turned into essential information if needed. It should be noted that all personal information that directly identifies a user must be deleted for General Data Protection Regulation (GDPR) reasons and that it would not make sense to add to the model. Emojis can be a useful source of information, but they were eventually discarded in the development of this present study since the focus is to forecast Bitcoin's trend. Several stop words have been detected that do not add value to the model, such as "I," "me," "my," "we," "he," "you," etc. This type of information will be treated so as not to pollute the model with false or unnecessary information for this study.

As the last step, an analysis was performed of how many tweets were null. Although no evidence of null tweets was detected in this dataset, null tweets may be filled with the text "No Content" to avoid unbalancing the dataset. The same was analysed for feelings in which no data without feeling were detected. With this, the analysis and exploration phase is done. Several possible problems were detected in advance and then dealt in the pre-processing phase in the subsection 3.4.3.

3.4.3 Pre Processing

This section details the steps and transformations taken to the data set due to the data exploration carried out in the above section. This phase aims to clean up unnecessary information, normalize the data, perform the correct data encoding, and split the dataset into a training and test subsets.

As previously analyzed, the data set is unbalanced, leading to the first phase of data normalization, with the aim of reducing the number of tweets in each sentiment category until 260000 tweets per category are received. For this reason, it was only necessary to remove the excess tweets and ensure they all have the same volume.

As the tweets arrives with many special characters, hashtags, links, urls, tags, among other types of information that are not relevant to the model. There's a need to perform a second pre-processing phase where it's removed this type of information that does not bring much value to the model. For this, a method is developed that receives a tweet in text format, removes all the unnecessary information for this case study and returns the same tweet with the information needed for training. The transformations applied to the tweets include removing URLs, emails, new lines represented by "/n", removing distracting single quotes, removing references or quotes to other twitter accounts that are represented as "@username", removing emojis, removing html tags and finally applying a transformation to the hashtags, thus transforming a hashtag into a word, such as e.g. "#Bitcoin" "Bitcoin".

After analyzing the tweets with the above mentioned cleaning layers, it was possible to verify that there were words that do not bring value at the semantic level, which are called stop words. Since the dataset contains tweets in English, the library NLTK was used, which contains the function "stopwords" that allows removing any type of stop word related to the English vocabulary. Some examples of stopwords removed from the dataset are: "the", "a", "an", "in". This process of removing stop words optimizes the space that the dataset occupies as well as reducing valuable processing time. An example of a transformation applied to a tweet would be the following text "Can listening be exchausting?", which after this processing would be in the format "Listening, Exchausting".

Following, after the cleanup made above, it was detected that there were tweets without text in the dataset. As mentioned on subsection above, these null tweets have been filled with the text "No Content" to prevent to imbalance the dataset.

After the pre-processing steps described above, it is necessary to apply one more final transformation to the content of the tweets. The ML models understand numeric values and the actual dataset, presents categorical values. So there is a need to transform the categorical values into numerical values. This process goes through two phases, a first phase in which the text is transformed into a sequence of words, called the tokenization process, where the method "Tokenizer" of the Keras library is used, and a second phase that transforms this same sequence of tokens generated previously into a list of integers. This second transformation is possible with the use of "pad_sequences" method offered again by the Keras library. Finally, to transform the sentiment column into a numeric value, a label encoding technique is used, where, using the method "to_categorical" from the Tensorflow library, it is possible to transform the sentiment from neutral, positive or negative into a numeric value between 0 and 1.

Finally, a dataset prepared for sending as input to the models is obtained. Since the models needs a training and testing phase, there is a need to split the dataset into two distinct datasets, one to perform the training process and a second one to perform the model evaluation. Given this, the function "train_test_split" from the Scikit Learn library is used, where the dataset is splited into two sub datasets, one for training and one for testing. For this case study and given the size of the dataset, it was splited 80% for the training phase and 20% for the test and validation phase.

The pre-processing phase is a crucial phase in defining the model's performance. It is a phase that requires a lot of research and exploration of the existing data to define the steps described above. Several problems and challenges faced were described, such as balancing an imbalanced dataset, as well as identifying and removing characters, symbols and words that do not bring much value to the model. Finally, three models will be described in sections 3.4.4, 3.4.5 and 3.4.6, which will use this dataset as input to perform sentiment prediction using tweets.

3.4.4 Long Short Term Memory Model

This subsection describes the steps taken to implement the LSTM neural network, how its architecture is designed, and the hyperparameters necessary to carry out the training.

As already mentioned, LSTM consists of a cell, an input gate, an output gate and a forget cell. The cells allow the storage of values over certain time intervals, and the three gates have the function of regulating the flow of information in and out of a cell. LSTM networks are helpful for classification, processing, and making predictions based on time series. Given

the network's memory capacity, the goal is to find dependencies between words and to persist some context in memory to make better predictions of sentiment in the future.

For this purpose, an LSTM model was built with a simple architecture, which includes an Embedding layer as an input layer, an LSTM layer as an intermediate layer and finally a Dense layer as an output layer. In order to group these layers, a Keras Sequential class was used, which aims to group a linear stack of layers into a model which offers us functionalities such as training and inference. In the figure 3.5 it is possible to visualize the implemented LSTM architecture.

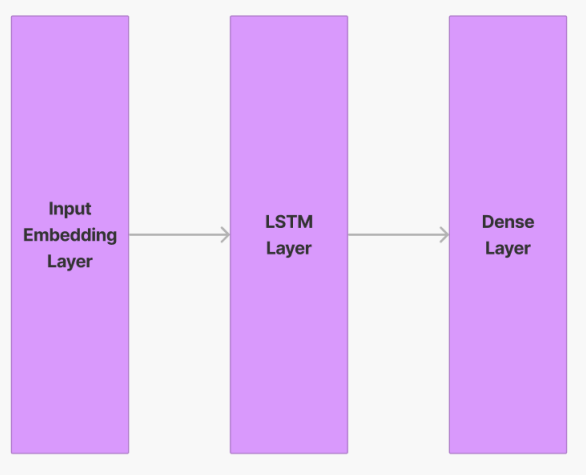

Figure 3.5: LSTM Model Base Architecture.

The primary function of the Embedding layer is to transform the list of integer values representing the words in the dataset into a dense vector of fixed size. This layer can only be used as an input layer in ML models. The embedded layer initially receives as parameters the dimension of the input, in this case the dimension of words consumed by the model, and a second parameter that defines the dimension of the output, which in turn defines the dimension of the dense embedding.

The second layer is an LSTM layer, which belongs to the family of RNN, it is a type of layer that allows to learn order dependency in sequence prediction problems. In this case study it is intended to analyze and learn about the dependency of words to make better sentiment predictions. In the present architecture the LSTM layer receives with parameter the units that define the dimensionality of the output space.

Finally an output layer called Dense layer is included, which aims to reduce the dimensionality of this network down to 3 outputs that represent each feeling, the positive, neutral and negative feelings. This Dense layer receives as parameters the units that represent the dimensionality of the output space, which in this case is always three, and receives an activation function initially defined as softmax. The code 3.2 represents the architecture described above.

```
model.add(layers.Embedding(max_words, 20))
model.add(layers.LSTM(units=10, dropout=0.5))
model.add(layers.Dense(3, activation='softmax'))
```

Listing 3.2: LSTM Model Layers Implementation.

After building the layers, it is necessary to compile them so that training can be carried out afterwards. Then, the compilation process is performed again by using the Keras library methods. In the code 3.3 it is possible to verify the compilation method.

```
model.compile(
  optimizer='adam',
  loss='categorical_crossentropy',
  metrics=['accuracy']
  )
```

Listing 3.3: BI-LSTM Model Layers Implementation.

In the compilation process, it is necessary to define some important parameters related to the optimizer selected, the loss function used, and other necessary metrics for visualization purposes during training. With regard to the optimizer, the optimizer adam was selected to measure the loss value during the training, the categorical_crossentropy method is defined to determine the loss, and finally a metric relative to the accuracy was added to visualize the accuracy of the present model. In the figure 3.6, it is possible to visualize a summary of the structure of the neural network after its compilation. The final network has 21273 parameters.

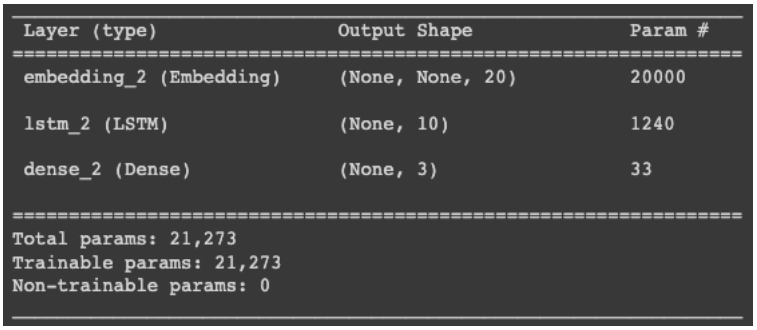

Figure 3.6: Sentiment Analysis LSTM Network Summary.

Concluding, this was the LSTM network used for this study. It should be noted that the training times of this network, using the physical machine mentioned initially, take about 30 to 40 minutes to complete. Although it is a simple network, the fact that the data set is of considerable size has led to a significant increase in training times. In subsection 3.4.7, the results obtained from this LSTM network are shown, and respective comparisons with the networks implemented in subsections 3.4.5 and 3.4.6 are detailed to confirm their performance.

3.4.5 Bidirectional Long Short Term Memory Model

This section describes the steps taken to build a BI-LSTM model, a slightly more complex model than the one mentioned in the 3.4.4 section. This model is known to get good metrics regarding text classification. As mentioned in 2.2 a BI-LSTM is the placement of two RNN together. This structure helps the network have backward and forward information about the sequence at every time step. This way, we can memorize information about the past and future for better precision of the feeling expressed at the moment. In the figure 3.7 it is possible to visualize the architecture used for the construction of this BI-LSTM.

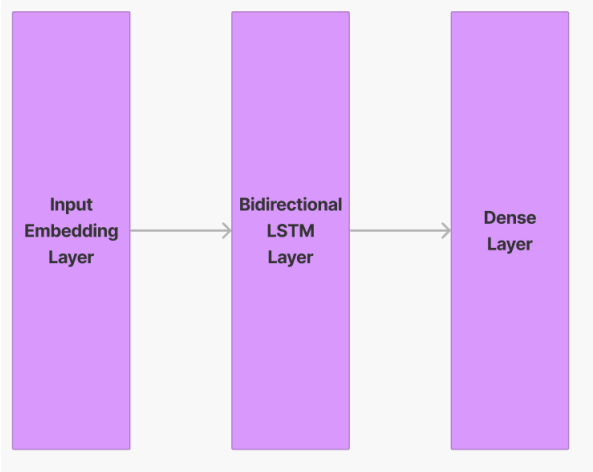

Figure 3.7: BI-LSTM Model Base Architecture.

The construction of this model follows the same structure as the LSTM model described above, with only one change in the hidden layer. With this, it is placed an Embedding layer as an input layer, then a bidirectional layer is placed that receives a LSTM layer as a parameter, and finally a Dense layer as the output layer of the model. It is possible to visualize in 3.4 the code referring to the architecture demonstrated above.

```
model.add(layers.Embedding(max_words, 20))

model.add(layers.Bidirectional(layers.LSTM(10, dropout=0.5)))

model.add(layers.Dense(3, activation='softmax'))
```

Listing 3.4: BI-LSTM Model Layers Implementation.

After building the layers, it is necessary to compile the model so that training can be performed afterwards. The compilation process is repeated using Keras methods. In the code 3.5 it is possible to observe the compilation method.

```
model.compile(
  optimizer='adam',
  loss='categorical_crossentropy',
  metrics=['accuracy']
  )
```

Listing 3.5: BI-LSTM Model Layers Implementation.

In the compilation process, it is necessary to define some important parameters related to the selected optimizer, the loss function used and finally other necessary metrics for visualization purposes during training. In terms of optimizer, the optimizer adam was selected to measure the loss value of our model, the categorical_crossentropy method was defined to determine the loss, and finally a metric relative to the accuracy was added to visualize the accuracy of this model. In the figure 3.8, it is possible to visualize a summary of the structure of the neural network after its compilation. This summary is provided by the existing methods in the Keras library. The network has a total of 222543 parameters.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 20)          20000

bidirectional_1 (Bidirectio  (None, 20)                2480
nal)

dense_1 (Dense)              (None, 3)                 63

=================================================================
Total params: 22,543
Trainable params: 22,543
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.8: Sentiment Analysis BI-LSTM Network Summary.

This subsection described the BI-LSTM network used for this study. It should be noted that the training times of this network, using the first mentioned physical machine, take about 30 to 40 minutes to complete. In subsection 3.4.7, the results obtained from this LSTM network are shown, and respective comparisons with the networks implemented in subsections 3.4.4 and 3.4.6 are detailed to confirm their performance.

3.4.6 Convolutional Neural Network Model

This section describes the steps to build a CNN model to classify feelings in tweets. When defining a structure in neural networks, it is always good to mention that there is no exact formula for building a good configuration. The best way to get a good configuration is through trial and error, where the different layers are explored and tested to evaluate their performance. In the figure 3.9 it is possible to visualize the architecture used to build the CNN.

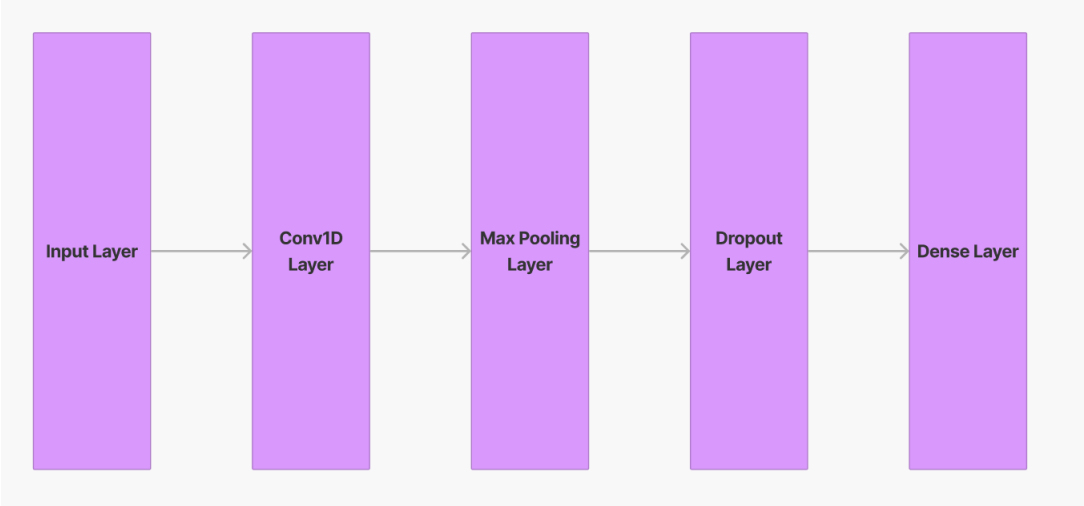

Figure 3.9: CNN Model Base Architecture.

Since this study aims to demonstrate the use of a CNN to classify feelings expressed in text, it was decided to implement a simple CNN to demonstrate its use and interpretation. Another reason to keep these networks simple is that they are complex networks and they consume a lot of computing power which would take a long time to train if it were a complex network.

In the code represented in 3.6 it is possible to visualize the code associated with the model represented above. Where an Embedding layer is used as an input layer, a Conv1D layer is followed by a MaxPooling layer, a Dropout layer, and finally, a Dense layer. The Keras library provides all these layers.

```
model.add(layers.Embedding(max_words, 20))
model.add(Conv1D(100, 5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dropout(0.2))
model.add(Dense(3, activation='sigmoid'))
```

Listing 3.6: CNN Model Layers Implementation.

After building the layers, it is necessary to compile them so that training can be carried out afterwards. The compilation process is performed again by the Keras methods. In the code 3.7 it is possible to observe the compilation method.

```
model.compile(
  optimizer='adam',
  loss='binary_crossentropy',
  metrics=['accuracy']
)
```

Listing 3.7: CNN Model Layers Implementation..

In the compilation process, it is necessary to define some important parameters related to the optimizer selected, the loss function used, and finally other necessary metrics for visualization purposes during training. In terms of optimizer, the optimizer Adam was selected, in order to measure the loss value of our model, the binary_crossentropy method was defined to determine the loss, and finally a metric relative to the accuracy was added to visualize the accuracy of the present model. In the figure 3.10, it is possible to visualize a summary of the structure of the neural network after its compilation. This summary is provided by the existing methods in the keras library. You can see that the network has a total of 30403 parameters.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 20)          20000

conv1d_1 (Conv1D)            (None, None, 100)         10100

global_max_pooling1d_1 (Glo  (None, 100)               0
balMaxPooling1D)

dropout_1 (Dropout)          (None, 100)               0

dense_1 (Dense)              (None, 3)                 303

=================================================================
Total params: 30,403
Trainable params: 30,403
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.10: Sentiment Analysis CNN Network Summary.

In conclusion, this was the CNN used for this study. It should be noted that the training times of this network, using the physical machine mentioned initially, take about 30 to 40 minutes to complete. Although it is a simple network, the fact that the data set is of considerable

size has led to a significant increase in training times. In subsection 3.4.7, the results obtained from this LSTM network are shown, and respective comparisons with the networks implemented in subsections 3.4.4 and 3.4.5 are detailed to confirm their performance.

3.4.7 Evaluation

This subsection compares the performances of the previous three models, and several metrics will be evaluated to debug the best model. It should be noted that each model was tested with the same development environment in order to obtain accurate metrics about each one. In table 3.1, it is possible to visualize the accuracy obtained in each of the three models mentioned above.

Table 3.1: Sentiment Analysis LSTM, BI-LSTM And CNN Models Accuracy.

Model	**Accuracy (%)**
LSTM	86.98
BI-LSTM	87.13
CNN	86.08

Based on the data in table 3.1, it is possible to see that the three models have good performance, where the BI-LSTM is achieving an accuracy slightly above the others. However, the accuracy metric alone cannot provide enough information about the models, so it was decided to analyze each model's precision, recall and F1 score. The precision is a ratio of correctly predicted values to the total of optimistic predictions. This metric intends to answer: how many observations were classified, for example, as positive sentiment, and how many were true? High precision is relative to how many false positives there are. Subsequently, recall is the ratio of correct predictions to the number of observations of the same class. Then, the F1 score consists of the weighted average of precision and recall.

Table 3.2: LSTM Sentiment Analysis Precision, Recall and F1 Score Measures.

LSTM	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
Negative Sentiment	85	88	86
Neutral Sentiment	86	84	85
Positive Sentiment	90	89	89
Accuracy			87
Macro Avg	87	87	87
Weighted Avg	87	87	87

In the table 3.2 it is possible to visualize the metrics related to the model based on the LSTM technique. The LSTM model obtains slightly higher accuracy when classifying positive sentiments, achieving an average accuracy of 87%. In terms of recall and F1 Score, the model obtained good results, around 87% on average, concluding that this model remains consistent and accurate for classifying feelings in tweets.

Table 3.3: BI-LSTM Sentiment Analysis Precision, Recall and F1 Score Measures.

BI-LSTM	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
Negative Sentiment	85	87	86
Neutral Sentiment	86	84	85
Positive Sentiment	90	89	89
Accuracy			87
Macro Avg	87	87	87
Weighted Avg	87	87	87

In the table 3.2 it is possible to visualize the metrics related to the model based on the BI-LSTM technique. The BI-LSTM compared to the LSTM are mostly identical, with a slight variance in recall when it comes to negative rating sentiment. It is another model with high accuracy that can be used to draw conclusions with a satisfactory level of confidence.

Table 3.4: CNN Sentiment Analysis Precision, Recall and F1 Score Measures.

CNN	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
Negative Sentiment	79	88	83
Neutral Sentiment	86	81	83
Positive Sentiment	90	86	88
Accuracy			85
Macro Avg	85	85	85
Weighted Avg	85	85	85

Ultimately, in the table 3.4, it is possible to visualize the metrics related to the model based on the CNN technique. CNN obtained good results but remained below the LSTM and BI-LSTM models. In terms of accuracy, CNN underperforms when it comes to negative feelings. However, this study proves that CNN can achieve a good accuracy when classifying text compared to RNN techniques.

Finally, a confusion matrix is used to find out in which cases the models fail and in which they succeed more often. The confusion matrix is important to detect possible flaws in the model, such as false positives or negatives. The figures 3.11, 3.12 and 3.13 shows the graphs related to the confusion matrix of the LSTM, BI-LSTM and CNN models.

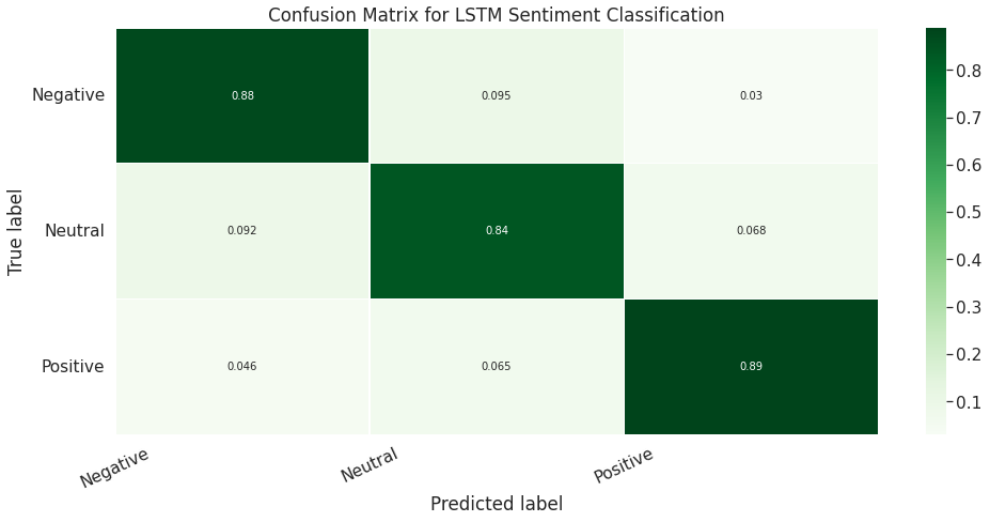

Figure 3.11: Sentiment Analysis LSTM Confusion Matrix.

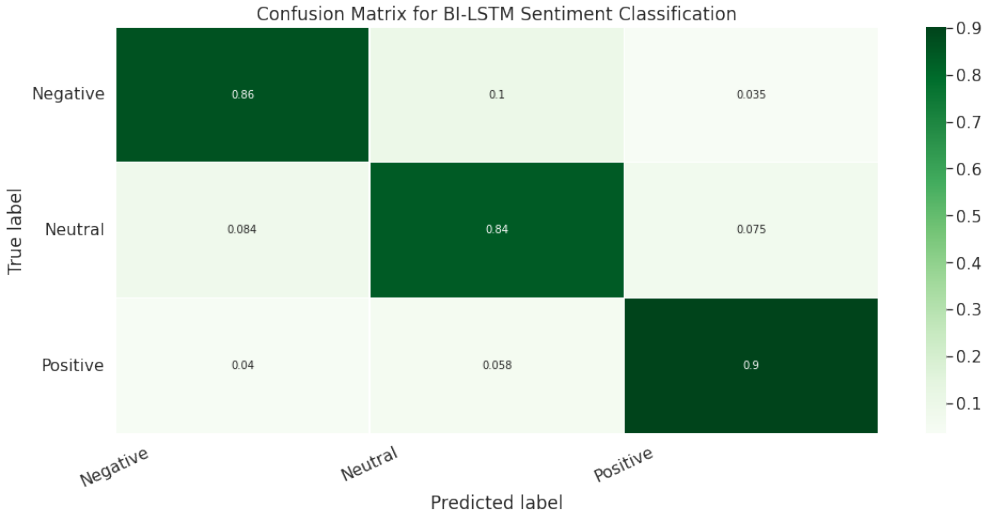

Figure 3.12: Sentiment Analysis BI-LSTM Confusion Matrix.

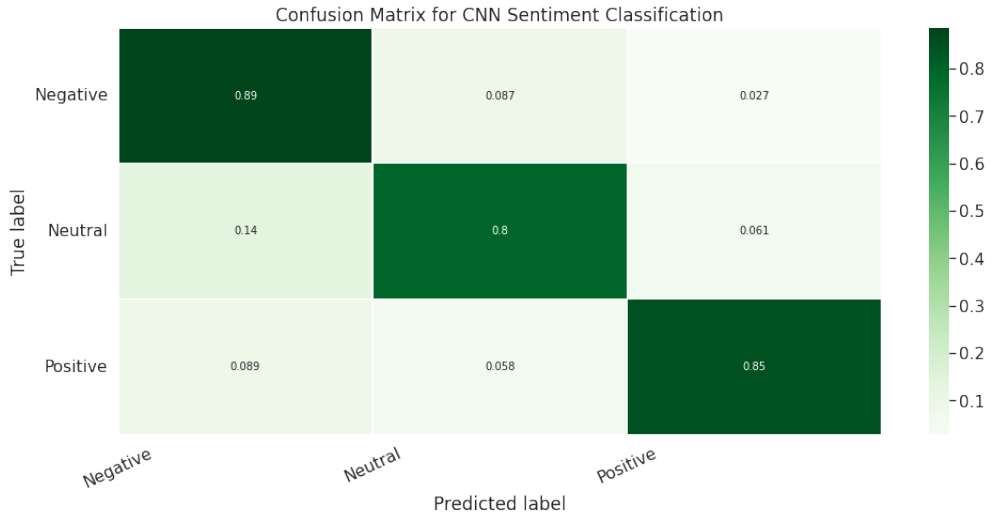

Figure 3.13: Sentiment Analysis CNN Confusion Matrix.

Comparing the three confusion matrices, it is possible to verify that the LSTM and the BI-LSTM are similar, indicating that the LSTM can be more consistent in the classification of negative feelings. On CNN, it is possible to verify a slight performance decline, but it remains consistent with the classification of the three feelings. It should be noted that both three models have more difficulty categorizing negative feelings and sometimes categorizing neutral feelings as negative. In general, the three models had high performance, with CNN slightly lower than the RNN.

In conclusion of the model evaluation process, the RNN proved superior when it came to greater consistency of the LSTM network in classifying the three different types of feelings. Since the LSTM is more consistent and does not contain many false positives or negatives, this will be the network used to draw the conclusion in the model described below to determine whether sentiment can positively influence the forecast of Bitcoin trends.

3.5 Bitcoin Trend Forecasting Model

This section describes all the phases carried out for the construction of the Bitcoin's trend forecasting model, where it is intended to describe all the steps taken to build a dataset, analysis and exploration of the same dataset, important metrics related to trend classification, construction of AI models, explanation of the tuning process carried out, and finally a demonstration of results obtained in the evaluation and validation phase of the model.

3.5.1 Dataset

This section describes in detail how the dataset is constructed. In order to predict the trend of Bitcoin for the next day, two datasets were collected, one of which contained daily tweets about Bitcoin and another with the historical price of Bitcoin. The figure 3.14 represents the structure implemented for the construction of this second dataset.

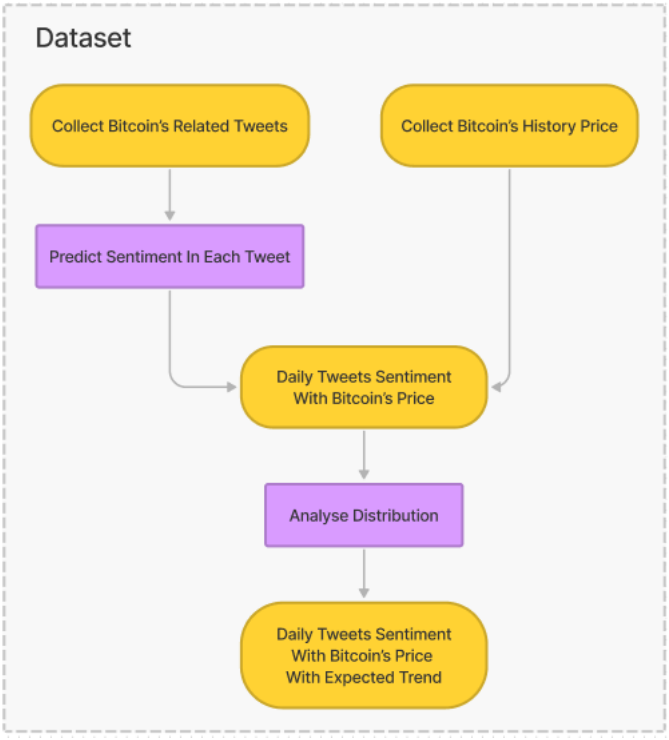

Figure 3.14: Bitcoin Trend Forecasting Dataset Collection Structure.

In a first attempt, tweets were collected for a whole day, which resulted in around 800 thousand collected tweets. However, this solution proved unfeasible from the point of view of duration. In the best scenario, this collection would allow the extraction of millions or possibly billions of tweets in 1 year. For this reason, daily tweets for seven months were collected using the Twitter's API, using an academic account on this platform that guarantees the extraction in large quantities, namely up to 10 million monthly tweets.

In this sequence, it was decided to collect 50 thousand daily tweets over a seven month period to demonstrate that it is possible to predict the trend of Bitcoin, guaranteeing the efficiency of the process in good time. It should be noted that if it were possible to obtain all existing tweets, the model would achieve better and more precise results.

In this way, a script was developed, whose main function is to extract 55 thousand daily tweets over the defined periods. To ensure the possibility of making calls to the Twitter's API, two functions have been developed, as shown in the code 3.8.

```
query_params = {
        'query': '(BTC OR Bitcoin OR #BTC OR #Bitcoin -RT) lang:en -is:
    retweet -is:reply -is:nullcast',
        'tweet.fields': 'created_at',
        'max_results': '500',
        'start_time': '',
        'end_time': ''
    }

def bearer_oauth(r):
    r.headers["Authorization"] = f"Bearer {bearer_token}"
    r.headers["User-Agent"] = "v2FullArchiveSearchPython"
    return r

def connect_to_endpoint(query_params):
    response = requests.request(
      "GET",
      "https://api.twitter.com/2/tweets/search/all",
      auth=bearer_oauth,
      params=query_params
    )

    if response.status_code != 200:
        raise Exception(response.status_code, response.text)

    return response.json()
```

Listing 3.8: Twitter Historical Data Collection.

According to the code snippet shown in 3.8, it is possible to check one of the ways to interact with the Twitter API to extract historical tweets. Through an Hypertext Transfer Protocol (HTTP) call to the Representational state transfer (REST) API, it is possible to perform queries based on some factors defined by the variable "query_params". This query has the ability to filter tweets, taking into account the information that is useful for our dataset. In detailing the "query_params" variable, the first property defined has "query" defines the type of content to be searched for in the tweets. Using the tweet filter based on the hashtags they contain allows the extraction of tweets on the subject in question and with a certain degree of relevance. For the selection of tweets, we opted for the inclusion of the English language and consequently the exclusion of retweets, replies, null or empty.

Subsequently, the "tweet.fields" property is defined, where possible extra columns intended to be extracted are assigned. In this particular case, Twitter only returns information regarding the text property. In this situation, the "create_at" property was added to extract the tweet's creation date, and then the "max_results" property to define the number of tweets to extract for each REST API call. In the present case, Twitter sets the maximum value of 500 tweets per request, so it was necessary to define this value. Finally, two more properties were defined, namely, "start_time" and "end_time", which can define dates in "YYYY-MM-DDTHH:mm:ss.SSSZ" format, e.g. 2016-06-23T09:07:21.205-07:00, allowing the collection of historical data.

Despite the benefits, there are some limitations at the level of the Twitter platform, mainly in terms of the number of tweets extracted for each request made to the REST API and the respective limitation of the number of requests per minute that can be made. Taking into account the above, a second script was developed, which is present in the 3.9 code, which iteratively extracts 50 thousand tweets every day.

```
YEAR = 2021
MONTH = 4
HOURS_PER_DAY = 24
MAX_TWEETS_PER_DAY = 55000
ITERACTIONS_PER_HOUR = int(math.ceil(MAX_TWEETS_PER_DAY / HOURS_PER_DAY
    / 500))

def main():
    query_params['start_time'] = '{0}-{1}-01T00:00:00Z'.format(YEAR,
    MONTH)

    for day in cal.itermonthdays(YEAR, MONTH):
        if(day < 10):
            day = "0{}".format(day)

        for hour in range(HOURS_PER_DAY):
            if(hour < 10):
                hour = "0{}".format(hour)

            for i in range(ITERACTIONS_PER_HOUR):

                minutes = randint(0, 59)

                if(minutes < 10):
                    minutes = "0{}".format(minutes)

                query_params['end_time'] = "{0}-{1}-{2}T{3}:{4}:00.000Z"
                .format(YEAR, MONTH, day, hour, minutes)

                json_response = connect_to_endpoint(query_params)

                filename = './data/year_{0}/month_{1}/day_{2}/{3}:{4}.
    json'
                .format(YEAR, MONTH, day, hour, minutes)

                os.makedirs(os.path.dirname(filename), exist_ok=True)

                with open(filename, 'w') as outfile:
                    json.dump(json_response, outfile)

                print("Retrieving data from {} to {}"
                .format(query_params['start_time'], query_params['
    end_time']))
                time.sleep(2)
```

Listing 3.9: Interactive Twitter Data Extraction.

Since it is only possible to extract 50 thousand tweets, they must be collected at different times of the day. The script exemplified in 3.9 has been developed for this purpose, extracting 50 thousand daily tweets in different hours and minutes to acquire tweets that can influence a certain period of the day. If the tweets were collected continuously, the focus was only on a specific period of the day, and it was impossible to detect events that could occur during the rest of the day.

At the end of the collection, the data was presented in a JavaScript Object Notation (JSON) format. Since working with data in CSV format makes it easier to read and process them. In this sense, an additional script was developed to perform the transformation from JSON to CSV, as represented in 3.10. This script also removes some fields that Twitter returns in

response to requests made during the collection of tweets, leaving the data clean and ready to be later analyzed and processed.

```
import pandas as pd
import json
import glob
from pathlib import Path

data = []

files_per_day = glob.glob('data/*/*/*', recursive=True)

def main():
  for file_per_day in files_per_day:
    print("loading file ./{}".format(file_per_day))

    all_day_files = glob.glob('{}/**/*.json'
              .format(file_per_day), recursive=True)

    result_data = pd.DataFrame()

    for day_file in all_day_files:
      data = json.load(open("./{}".format(day_file)))

      df = pd.DataFrame(data["data"])

      result_data = result_data.append(df, ignore_index=True)

    output_dir = Path('result/{}'.format(file_per_day))
    output_dir.mkdir(parents=True, exist_ok=True)

    result_data.to_csv("./result/{}/data.csv".format(file_per_day),
    index=None)


if __name__ == "__main__":
    main()
```

Listing 3.10: JSON to CSV Transformation.

After performing this first data collection, a CSV was obtained with the columns related to the creation date and the text referring to the tweet's content. Then, it was necessary to apply the model developed in 3.4 and perform the inference on the collected data. To this end, the model was loaded and iteratively ran through all the CSV for all the collected days, starting to perform sentiment inference in each of them.

One of the difficulties experienced at this stage was the time it took to predict the feelings in these new tweets, given that 50 thousand tweets were collected daily over seven months, generating a total of 20 million tweets. On the other hand, it was found in the sentiment inference phase of the feeling that the processing time to draw feelings was considerably high. Based on the hardware and software described in 3.1.1, it was possible to process and predict the tweets at a speed of 25 tweets per second, which took a long time to complete this operation. For reference, inferring over 50,000 tweets took about 37 minutes, which took a few days to infer all the data collected.

Later, the collection of historical Bitcoin data was conducted to group the two datasets. These data were extracted using the free online platform called Yahoo! Finance (Yahoo

2022), which is available to the public. The platform provides financial news, data and commentary, including stock quotes, press releases, financial reports and original content. Given its applicability, it was possible to extract historical Bitcoin data from 2014 to the present year of 2022. Such data was extracted in CSV format, which enabled a daily view of Bitcoin with the opening price, minimum and maximum value of the day, closing price and the amount of money that flowed that day. The data structure is represented in JSON format in the 3.11 version. The data was used in the second model to predict the currency trend. In this dataset, a statistical distribution manually categorizes the trend forecast, analysing whether the trend is uptrend, strong uptrend, downtrend or strong downtrend.

```
{
  "Date": "2022-02-05",
  "Open": "41441.921875",
  "High": "41825.601563",
  "Low": "41079.910156",
  "Close": "41679.984375",
  "Adj Close": "41679.984375",
  "Volume": "21720416256"
}
```

Listing 3.11: BTC-USD Dataset JSON Format Representation.

Given that this data is publicly available, there is no obstacle to collecting it. At the moment, there are several free platforms that can extract them. Since these are daily forecasts, it is easier to collect data for daily periods than in a time interval of less than 24 hours, as there may be some adversity, as these data are only exposed daily. Finally, these data are correlated with the data from the predictions made with the first dataset described in 3.5.1, thus summing the sentiment expressed daily, as well as the volume of tweets shared daily.

Then it is necessary to combine this sentiment collected in daily tweets with the historical price of Bitcoin. With this in the following code 3.12, it is possible to visualize the steps taken to join these two datasets.

```
import glob
import time
import pandas as pd
from tqdm import tqdm

files = glob.glob('../../bitcoin_prediction/data/*/*/*/data.csv',
    recursive=True)
btc_file = glob.glob('../../bitcoin_prediction/data/BTC-USD.csv',
    recursive=True)[0]

btc_dataset = pd.read_csv(
          btc_file,
          lineterminator='\n'
        ).drop(columns=[
          "High",
          "Low",
          "Close",
          "Adj Close"], axis=1)

historical_sentiment = pd.DataFrame(columns=[
                  "Date",
                  "Negative",
                  "Neutral",
                  "Positive"
                ])

for file in tqdm(files):
    dataset = pd.read_csv(file, lineterminator='\n').drop(columns=["text
    ", "id"], axis=1)

    dataset["created_at"] = pd.to_datetime(dataset["created_at"])
    dataset["created_at"] = dataset["created_at"].dt.strftime('%Y-%m-%d'
    )

    dataset["sentiment_count"] = dataset.groupby("Sentiment")["Sentiment
    "].transform('count')

    dataset = dataset.drop_duplicates(subset="Sentiment", keep="last")

    dataset = dataset.pivot('created_at', 'Sentiment').stack(0).
    reset_index()
    dataset = dataset.rename_axis(None, axis=1).reset_index(drop=True)

    dataset = dataset.drop(columns=["level_1"], axis=0).rename(
    columns={
      'created_at': 'Date',
      0: "Negative",
      1: "Neutral",
      2: "Positive"
    })

    final_dataset = pd.concat([final_dataset, dataset], axis=0)

historical_sentiment = final_dataset.sort_values(by='Date')

merged_datasets = pd.merge(final_dataset, btc_dataset, how="outer", on="
    Date")
          .sort_values(by='Date')
          .drop_duplicates(subset="Date", keep="last")

merged_datasets.to_csv('../../bitcoin_prediction/data/
    historical_btc_sentiment.csv', index=False)
```


Listing 3.12: Bitcoin's Trend Prediction Join Sentiment Dataset With Historical Bitcoin's Price.

This way, a data set is obtained that contains the number of positive, negative and neutral daily tweets, together with the bitcoin price for that day.

Table 3.5: Historical Bitcoin And Tweet Related Sentiment.

Date	Positive	Neutral	Negative	Open Price ($)	Volume
2021-07-08	17517	22363	10120	33889.60	29910396946
2021-07-09	15785	24271	9944	32861.67	27436021028
2021-07-10	15528	23659	10813	33811.24	22971873468
2021-07-11	17457	22751	9792	33509.07	20108729370

In the 3.5 table, you can see a sample of the structure of the final dataset in terms of sentiment typology (positive, neutral and negative), opening price and volume of money transacted. After completion of this process, a complete dataset with the respective data history associated with sentiment and Bitcoin was obtained. In this way, we already have all the information necessary for the beginning of the data exploration phase, which can be found in the next subsection. It should be noted that the collection and inference phase was the development phase that took the longest time in the development of this study. The collection of data, the inference of sentiment in the data, the integration of the datasets and the manual categorization of the trend were also a slow and time consuming processes, which required much computational power to speed up the process. Fortunately, it was possible to carry out this entire process in good time, highlighting the feasibility of giving up some advantages, namely limiting the collection of tweets to 50 thousand daily tweets, rather than carrying out a total collection of tweets.

3.5.2 Data Exploration

In this subsection, the phases covered in exploring data from the dataset described in the previous subsection are detailed. Again, this phase is crucial for the model's performance where the data will be later analyzed to understand useful information and possible correlations.

Taking into account the objective of this study and the dataset presented in the table 3.5, it was necessary to find a way to calculate and classify the trend in order to create a new column in the present dataset that can identify the trend as strong uptrend, uptrend, downtrend or strong downtrend. For this, a technique called SMA was used, consisting of a moving average calculated by adding the most recent Bitcoin prices and dividing the same value by the number of periods in the calculation average. The SMA is expressed using the following formula:

$$\text{SMA} = \frac{A_1 + A_2 + \ldots + A_n}{n} \tag{3.1}$$

Where A_n is the price of Bitcoin in a given period n and n is the total number of periods. After applying the SMA to the opening price of Bitcoin with a period of 7 days, the following diagram was obtained, represented in the figure 3.15.

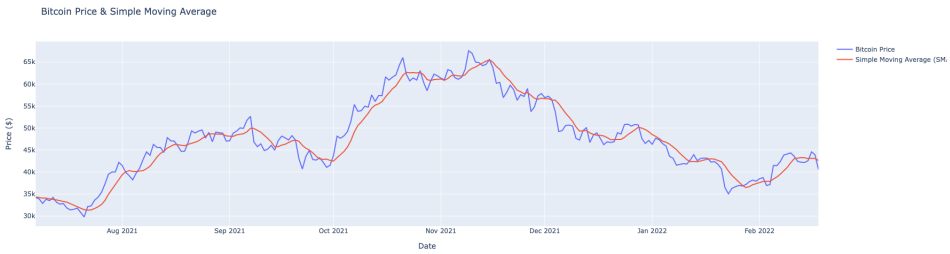

Figure 3.15: Bitcoin Price With Simple Moving Average.

According to the figure 3.15, it is possible to visualize, taking into account the seven days of the Bitcoin price, the help of the SMA curve is smaller, allowing it to detect more assertively when trend inversions occur. Also preventing drastic price variation to induct in false trends classification.

After this analysis, it was necessary to observe the distribution relative to the daily price difference of the SMA. For this reason, the difference between the day $n+1$ and the day n, represented in the graph of the figure 3.16, was performed.

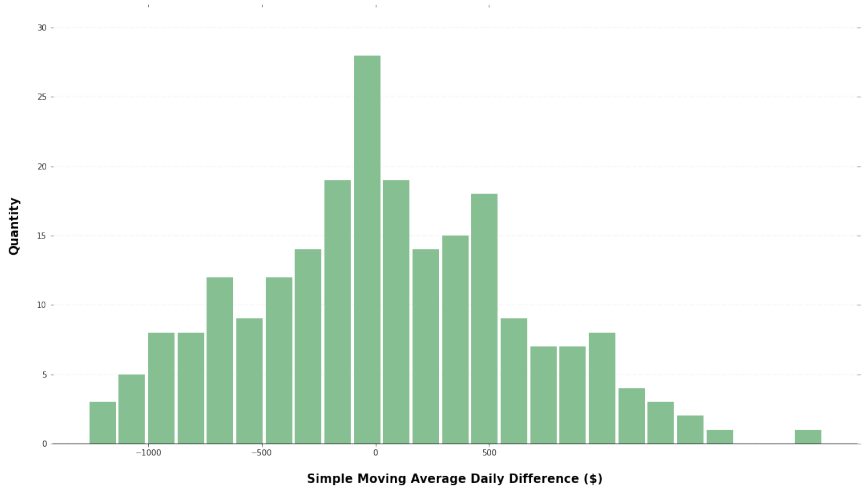

Figure 3.16: Simple Moving Average Daily Difference Analysis.

Analyzing the 3.16 graph, it was possible to verify the existence of a uniform distribution of the values of the SMA, which made it possible to divide it into 4 different categories that correspond to the four intended trends. The table 3.6 shows the division performed and the price range where each trend fits. Due to AI models needing numerical values, a value between -2 to 2 will be associated, identifying the respective trend.

Table 3.6: Bitcoin's Trend Classification.

	Price Interval ($)	Classification Number
Strong Uptrend	500 $ +	2
Uptrend	0 $ - 500 $	1
Downtrend	-500 $ - 0 $	-1
Strong Downtrend	-500 $ +	-2

With the trend classification defined, a graphic comparison was made between the columns referring to the sentiment and the trend to validate whether there is any common pattern or that identifies any correlation.

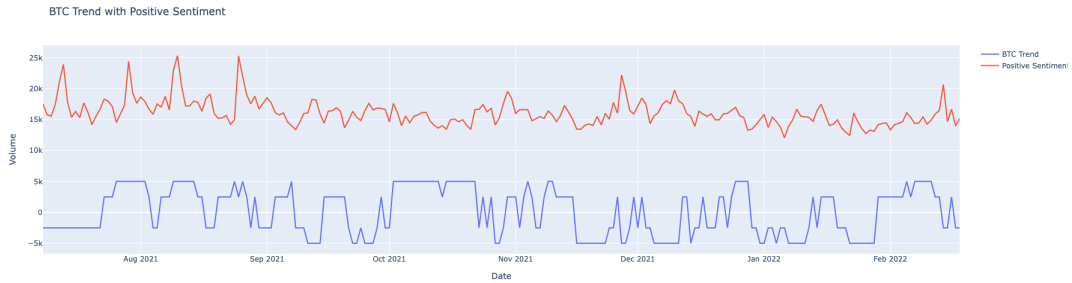

Figure 3.17: Bitcoin's Positive Sentiment with Trend.

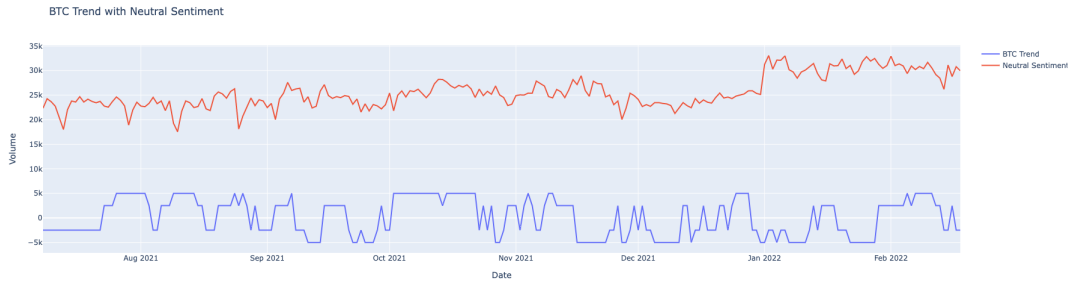

Figure 3.18: Bitcoin's Neutral Sentiment with Trend.

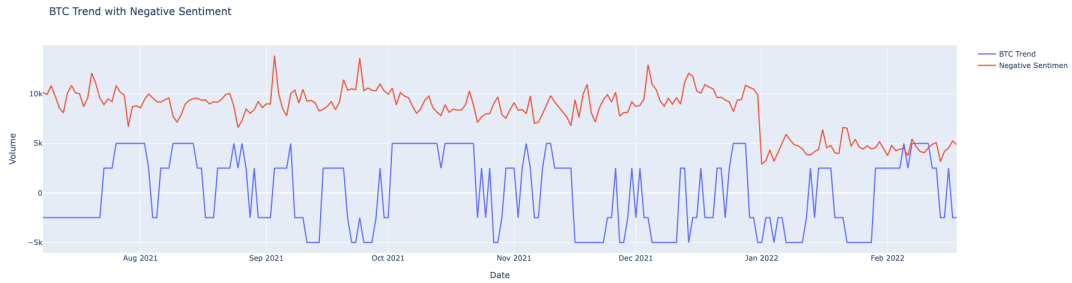

Figure 3.19: Bitcoin's Negative Sentiment with Trend.

The figures 3.17, 3.18 and 3.19 show the volumes related to sentiments with the trend value. For graphic visualization purposes, the trend present in the figures was scaled to compare with the sentiment volume through an easy visualization. The table 3.7 shows the scale where each trend fits to interpret the referring figures.

Table 3.7: Respective Bitcoin's Scaled Trend.

Trend	**Trend classification**	**Scaled Trend**
Strong Uptrend	2	5000
Uptrend	1	2500
Downtrend	-1	-2500
Strong Downtrend	-2	-5000

Regarding the figure 3.17, it is possible to see that during the time interval between August 2021 and September 2021, the peaks related to positive sentiment are well related to the strong uptrend peaks, in contrast during the interval of time between October 2021 and November 2021 the same did not happen. Sometimes it is possible to see that positive sentiment correlates well with uptrends. However, this pattern does not always happen, which complicates the graphic visualisation.

In turn, in the figure 3.18, there is a more uniform distribution of neutral sentiment over time. However, it is impossible to visualize a pattern in the first instance. It should be noted that in January 2022, there was a rise in neutral sentiments. However, it was impossible to detect any event in that period that justified this rise.

Finally, in the figure 3.19, it is possible to verify that there is a certain pattern in the period from August 2021 to September 2021. In fact, after a rise in the trend, it is possible to validate a fall in volume with negative sentiment. At the same time, there is a rise in negative sentiment in periods of a downtrend. However, in the initial period of September 2021, it is possible to verify that the volume of negative sentiment increased significantly when the trend was rising. In the time interval between December 2021 and January 2022, it is possible to verify again some patterns that confirm some relationships. In summary, in January 2022, as shown above in Figure 3.18, there is an event in which volume falls relative to negative sentiments, where no cause could be found.

Regarding the event from December 31, 2021, to January 1, 2022, shown in the figures 3.18 and 3.19, a search for news on the internet regarding the date was made. No incident could have caused a rise in neutral sentiment and a fall in negative sentiment. Another strategy involved analysing the total volume of tweets made daily on the subject in question to verify the possibility of being related to some increase or decrease in the volume of tweets made on that date.

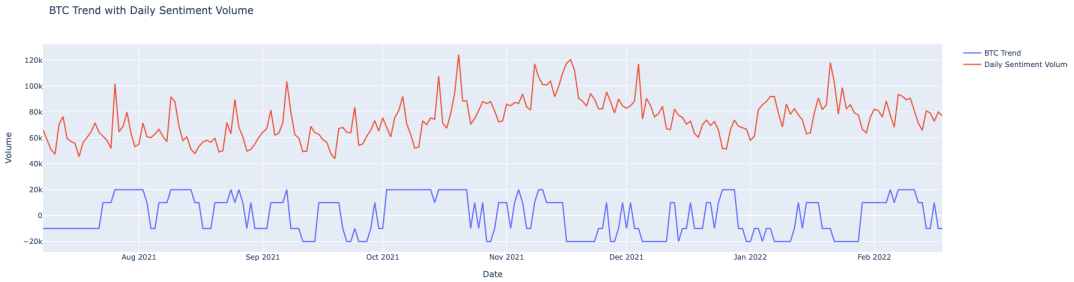

Figure 3.20: Daily Tweets Volume With Trend.

The figure 3.20 shows the total volume of tweets made daily and the trend corresponding to each day. It should be noted that due to the volume of daily tweets being on a larger scale, the respective trend was scaled to improve the comparison. With this, in the table 3.8 are referred to the scales used for the trend present in the figure 3.20.

Table 3.8: Respective Bitcoin's Scaled Trend.

Trend	**Trend classification**	**Scaled Trend For Volume**
Strong Uptrend	2	20000
Uptrend	1	10000
Downtrend	-1	-10000
Strong Downtrend	-2	-20000

After obtaining the analysis shown in the figure 3.20, it was impossible to detect any anomaly from December 31, 2021, to January 1, 2022, in terms of increase or decrease in the volume of tweets made. It should be noted that is possible to verify some patterns in terms of volume with the trend. For example, between August 2021 and September 2021, it is possible to see that the volume of tweets increases when there is an uptrend and decreases if there is a downtrend. In the time interval between January 2022 and February 2022, it is possible to analyze that the volume of tweets increases if there is a downtrend, and that it stabilizes when this downtrend reverses to an uptrend. Taking into account the visualization of some patterns throughout this analysis, a column corresponding to the volume of daily tweets was added to the dataset, to determine whether it could have a positive or negative impact during the training phase of the model described in the 3.5.5 section.

Then, a graphical analysis of the three types of sentiments was performed to assess whether there was any pattern. For example, when the volume of positive sentiment goes up and the volume of negative sentiment goes down. The figure 3.21 represents the volume of the three types of sentiment over time.

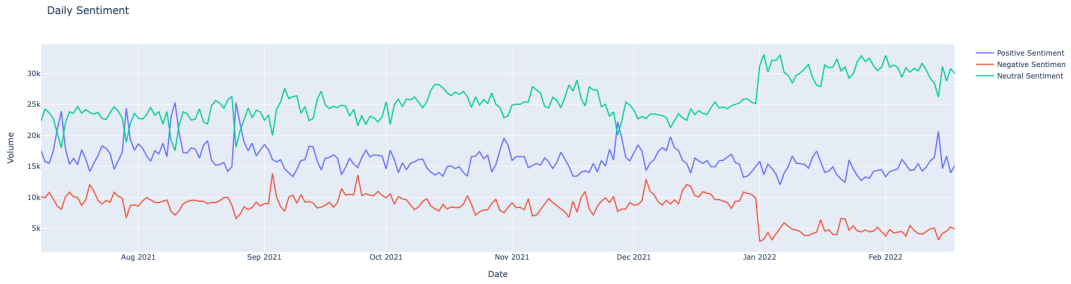

Figure 3.21: Daily Tweets Sentiment.

Through the figure 3.21 it is possible to verify that in the initial period until September 2021, when positive sentiment rises, neutral sentiment tends to fall. The opposition of positive feelings with negative feelings is also validated. When one goes up, the other tends to go down. An interesting fact has been revealed between September 2021 and December 2021, that positive sentiment does not follow the opposite volume of negative sentiment. Sometimes both tend to rise equally. Finally, it should be noted that the change in sentiment referred to above, from 31 December 2021 to 1 January 2022, increased the volume of neutral sentiment and the volume of negative sentiment decreased both significantly. Effectively, positive sentiment tends to remain stable over time.

Then a graph was constructed to analyze the sentiment distribution in each of the four trends. For this purpose, a boxplot graph was used to analyze the essential metrics related to the distribution of the dataset, as shown in the figure 3.22.

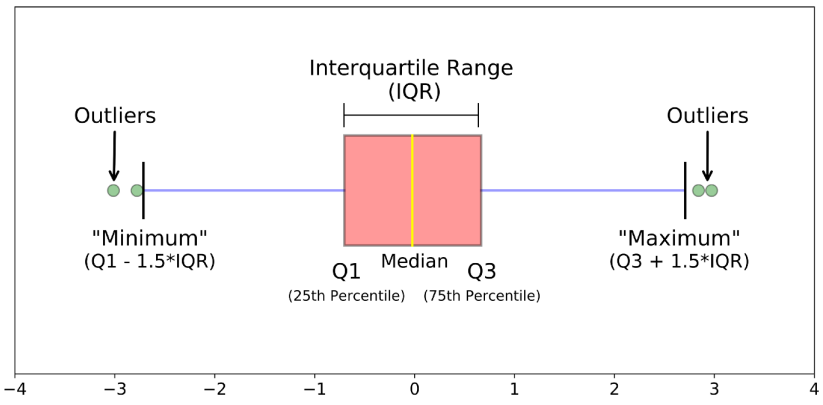

Figure 3.22: Bitcoin's Tweets Boxplot Analysis.

Taking into account the figure 3.22, it is intended to analyze the following parameters:

Median (Q2/50th Percentile) – the middle value of the dataset.

First quartile (Q1/25th Percentile) – the middle number between the smallest number (not the "minimum") and the median of the dataset.

Third quartile (Q3/75th Percentile) – the middle value between the median and the highest value (not the "maximum") of the dataset.

Interquartile range (IQR) – 25th to the 75th percentile.

Whiskers – shown in blue.

Outliers – shown as green circles.

Maximum – Q3 + 1.5*IQR.

Minimum – Q1 -1.5*IQR.

Using this technique allows extracting important information about the data in question. After application to the dataset in question, the graph represented in the figure 3.23 is obtained. Where it is possible to visualize the distribution of data by quartiles. It is possible to verify that the neutral feeling is found with more volume than the negative and positive feelings. It is also verified that the feelings are always with the same value or similar values. In the case of downtrend classification (-1), it is possible to verify some outliers in the neutral sentiment and in the strong uptrend classification in terms of positive and negative sentiments in some outliers. In conclusion, it is possible to observe a somewhat dispersed dataset, where some outliers are identified in some data.

Figure 3.23: Bitcoin's Tweets Sentiment Distribution.

After completing this dataset exploration phase, several patterns confirmed correlations between sentiment and trend. The trend and volume of daily tweets were two new columns added to the dataset during this exploration. This led to a final dataset containing, in addition to those mentioned in 3.5, two more columns represented in the following table 3.9.

Table 3.9: Bitcoin's New Trend With Daily Tweet Volume Columns.

Date	Trend	Daily Tweet Volume
2021-07-08	0	66016
2021-07-09	-1	58476
2021-07-10	1	50854
2021-07-11	0	47357

Subsequently, in the subsection 3.5.3, the pre processing data phase will be addressed, an equally important phase where the data transformation is carried out to facilitate the model's perception.

3.5.3 Pre Processing

In this subsection, the pre-processing phase is detailed, whose main objectives are cleaning the data, applying some transformations and dividing the dataset into training and testing sub-datasets. In this model, since the goal is to predict the trend of the following day, it was necessary to create a column of targets equal to the column of the trend but with one day ahead. The following formula can represent this transformation:

$$T_m = T_{m-1} \tag{3.2}$$

Where T_m is the trend intended as a target for the model and T_{m-1} is the trend of the previous day.

As the trend has been moved for one day, it is then necessary to remove the first element from the dataset, considering that it will be an emoty value. To perform this operation, the dropna function of the pandas data frame was used. To ensure that the number of positive, neutral and negative tweets is consistent throughout the dataset, they were normalized to a fixed value. In this phase of data collection, about 55 thousand tweets were extracted daily. However, sometimes their collection could be smaller or larger and have fewer or more thousand tweets daily. To prevent the model from misinterpreting the volume variations that could occur, the amount in each of the three columns of the dataset were reduced in the same way, resulting in a total of 50 thousand daily tweets. Since the model needs to obtain the data in a two dimensional format, it was necessary to perform a data reshape to transform a dataset from one to two dimensions. This transformation consists of converting a simple array into an array of arrays, using an existing function from the NumPy library shown in the code 3.13.

```
data = data.reshape(len(data), 1)
```

Listing 3.13: Pre Processing Data Reshape.

With this transformation, the model can correctly interpret the training data.

The visualization of the table previously mentioned in 3.5 shows that the data are on very different scales. One example is the columns on the sentiment are in the two tens of thousands and the volume column in the tens of billions. In this context, it is necessary to normalize these data to a common scale so that the model can understand them. For this, the StandardScaler function of the Scikit Learn library was used to resize the distribution of values so that the mean of the observed values is zero and the standard deviation is one.

A value is standardized as follows:

$$y = \frac{x - mean}{standard\_deviation} \tag{3.3}$$

Where the mean is calculated as:

$$mean = \frac{1}{N}\sum_{i=1}^{N}(x_i) \tag{3.4}$$

And the standard deviation is calculated as:

$$standard\_deviation = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - mean)^2} \tag{3.5}$$

Finally, the dataset was divided into two sub datasets, one for the training phase and a second for the test phase, using the train_test_split function of the Scikit Learn library, which allows to easly split the features and targets in two training and test datasets. It should be noted that this division cannot be performed randomly, since it is a time series dataset, which does not allow randomization in the dates. To solve this problem, the train_test_split function provides a parameter called shuffle, once set to false, prevents the dataset from being split randomly. The dataset split was performed with 70% for training data and 30% for test data, this can be defined again in the train_test_split function through the train_size and test_size properties. With this preprocessing phase, the data were all processed and normalized to obtain a better performance during the model training phase.

3.5.4 Long Short Term Memory Model

This subsection addresses the development of an LSTM model to make predictions based on a time series dataset to predict the next day's trend for Bitcoin. The creation of this model goes through several stages of development. Since the model will make predictions based on time series, several particularities must be taken into account when developing this type of model.

Based on the last step described in the previous subsection regarding the pre-processing phase, the data set was divided into 70% for training data and 30% for test data. Since it is a model based on time series, this division was carried out to avoid randomness in the data and to maintain the order by date. Since this is a multivariate problem, a time series generator method from the Keras library was used to automatically transform a multivariate time series dataset into a supervised learning problem. In the code 3.14, it is possible to visualize the implementation of a TimeseriesGenerator to create a generator for training and testing. In addition to the ability to transform a multivariate time series dataset into a supervised learning problem, it also transforms the dimensionality of the data so that it can be used directly in the LSTM model.

```
train_generator = TimeseriesGenerator(X_train, y_train, length=look_back
    , batch_size=1)
test_generator = TimeseriesGenerator(X_test, X_test, length=look_back,
    batch_size=1)
```

Listing 3.14: Bitcoin Trend Prediction LSTM TimeseriesGenerator.

The TimeseriesGenerator method receives the input data sequence of the model, the sequence of targets corresponding to the model's output, length and batch size. The length is relative to the expected size of the output, in this case how many days in the future is necessary to predict. While the batch size is the number of time series samples that are desired to include in each batch during training. The batch size value is fixed to the value one, as these are daily forecasts, so it is intended to include only one day at a time in the training phase.

The data is prepared to be passed on as input data in the LSTM network. The next step is to describe the architecture followed for its construction.

```
model = Sequential()
model.add(LSTM(100,
       input_shape=(look_back, n_features),
       return_sequences=False))
model.add(Dense(1))
model.compile(
  optimizer='adam',
  loss='mse',
  metrics=['accuracy'])
```

Listing 3.15: Bitcoin Trend Prediction LSTM Model Architecture.

The code 3.15 represents the base architecture used to compose the LSTM model. It is possible to verify that an LSTM layer with 100 input units is used. Then input shape property defines the input dimensions. The first input dimension is the number of days to forecast, and the second is the size of data features. Then a return sequence property is defined to prevent the model from creating randomness on the passed data. Finally, a dense layer with only one dimension was defined. This dense layer will be the output layer of the model. In summary, the Sequential Layer from Keras was used to add these different layers.

In order to carry out the training, the "fit" method offered by the sequence layer was used. In the code snippet 3.16, you can see how this method is used. Where it is possible to verify that training and test data have been passed, as well as the number of epochs the model will perform.

```
model.fit(train_generator, validation_data=test_generator, epochs=20)
```

Listing 3.16: Bitcoin Trend Prediction LSTM Fit Method.

After some tests with this network, it was possible to verify that the network is inconsistent with the data provided. Several ways of implementing this model were tested, but all without success. The model was found after the training phase ended up overfitting, where no justification for such an eventuality was found. Since the data set contains 220 days of data, it may be that the amount of data is not enough to train this type of network. On this basis, it was decided to proceed with a supervised approach using the Random Forest technique described in the next subsection.

3.5.5 Random Forest Classifier Model

This subsection describes all the steps to build a random forest classifier model. As mentioned in the previous subsection, the LSTM model did not achieve good results during training, so we choose a supervised approach using the random forest classification technique. Although this technique can be used for several tasks in the scope of regression and classification, in this study will be used to classify the intended trends. The random forest is an ensemble method, which means that a random forest model consists of many small decision trees, called estimators, each producing their predictions. The random forest model combines the predictions of the estimators to make a more accurate prediction.

```
params = {
    'max_depth': [1,2,3,5,10],
    'min_samples_leaf': [5,10,20,50,100],
    'n_estimators': [10,25,30,50,100]
}

rf = RandomForestClassifier(random_state=42, n_jobs=-1)

grid_search = GridSearchCV(estimator=rf,
                           param_grid=params,
                           n_jobs=-1,
               verbose=1,
               scoring="accuracy")
```

Listing 3.17: Bitcoin Trend Prediction Random Forest Classifier Architecture.

In the 3.17 code, it is possible to analyse the construction of the random forest classifier model. Where, in general, some training parameters are defined, then the RandomForestClassifier class is instantiated. Finally, the GridSearchCV class is used to optimize the tuning process. The GridSearchCV class is the process of performing hyperparameter tuning to determine the optimal values for a given model. The performance of a model significantly depends on the value of hyperparameters. It is necessary to conduct several pieces of training to find the appropriate hyperparameters for the model. Doing this process manually would take much time. GridSearchSV is used to solve this tuning of hyperparameters problem.

In order to perform this automatic tuning process, it is necessary to define a dictionary parameter with the values to be tested during the training phase. The GridSearchSV then tests all combinations passed in this dictionary and evaluates the performance of each combination through cross-validation. Finally, the accuracy and loss of each combination is achieved, where we want to choose the combination with the best performance. The variable "params" originally define this dictionary parameter.

Finally, GridSearchSV receives an estimator where the RandomForestClassifier is passed to perform all possible combinations based on a random forest classifier estimator. The parameter "n_jobs" is also defined in the GridSearchSV, where it is possible to define the number of processes created to perform the training. When defined with the value -1, the model will use all available processes to perform the training. In conclusion, the "scoring" parameter was defined as the metric to evaluate performance during training, which in this case was defined to accuracy.

In summary, this was the architecture used for the training using the random forest classifier model. In the following subsection, the results obtained in this model and the various tests performed are evaluated to confirm whether sentiment has a positive or negative impact on predicting future trends in the Bitcoin.

3.5.6 Evaluation

This subsection describes the experiments carried out with the algorithm specified in section 3.5.5. In this sequence, several tests were carried out with different hyperparameters and data to investigate the multiple possible variations, evaluate the model's performance and analyze the possible influence of sentiment and the cryptocurrency market.

In a first analysis, the three sentiments were tested with the volume of daily tweets, the price of Bitcoin, the volume traded in Bitcoin and the trend. The table 3.10 shows the percentage of accuracy obtained in each training.

Table 3.10: Random Forest Classifier Model Train With Positive, Neutral And Negative Sentiment Data.

Experiment	Tweet Volume	Price	Volume	Trend	Accuracy (%)
1	Yes	Yes	Yes	Yes	45.58
2	No	Yes	Yes	Yes	38.23
3	No	No	Yes	Yes	41.17
4	No	No	No	Yes	51.47
5	No	Yes	No	Yes	48.52

According to the table 3.10 it is possible to analyze in experiment 1, by maintaining all the data, the model obtained a performance of 45.58%. Then experiments 2, 3, 4 and 5 were carried out, where we intend to remove some of the present data and validate if the model improves or worsens in terms of performance. In this sense, it was possible to verify that after removing the volumes from the equation, the model improves slightly, which leads to the deduction that the volume can negatively influence the trend when sentiment is used.

Then, it is necessary to validate that the sentiment positively impacts the model. Taking into account the need, a second set of experiments was conducted in which the columns related to sentiment were removed from the equation, as well as the volume of tweets carried out every day. The 3.11 table shows the results obtained in this second set of experiments.

Table 3.11: Random Forest Classifier Model Train Without Positive, Neutral And Negative Sentiment Data.

Experiment	Tweet Volume	Price	Volume	Trend	Accuracy
1	No	Yes	Yes	Yes	35.29%
2	No	No	Yes	Yes	36.76%
3	No	Yes	No	Yes	33.82%

In the 3.11 table, it is possible to see a drop in performance when sentiment is removed from the present dataset, which effectively validates that sentiment has a positive impact and helps predict the Bitcoin trend. A performance drop of around 10% is also verifiable, representing a significant value when it comes only to adding the sentiment expressed on the Twitter social network. However, further experiments have been carried out to confirm whether positive, neutral and negative feelings in isolation correlate more with the trend. Therefore, in the table 3.12 a new set of experiments performed only with positive feelings.

Table 3.12: Random Forest Classifier Model Train With Only Positive Sentiment Data.

Experiment	Tweet Volume	Price	Volume	Trend	Accuracy
1	Yes	Yes	Yes	Yes	35.29%
2	No	Yes	Yes	Yes	42.64%
3	No	No	Yes	Yes	35.29%
4	No	Yes	No	Yes	36.76%

In the set of experiments represented in the table 3.12, it is possible to verify that the positive sentiment alone does not positively impact the performance of the model. However, there is no cadence of performance compared to the experiments carried out in the table 3.11. Therefore, in this sequence, it can be concluded that positive sentiment alone does not have enough capacity to predict the next day's trend. Then a new set of experiments was performed, represented in the table 3.13, analyzing only the neutral sentiment.

Table 3.13: Random Forest Classifier Model Train With Only Neutral Sentiment Data.

Experiment	Tweet Volume	Price	Volume	Trend	Accuracy
1	Yes	Yes	Yes	Yes	39.70%
2	No	Yes	Yes	Yes	54.41%
3	No	No	Yes	Yes	36.76%
4	No	Yes	No	Yes	57.35%

In the set of experiments present in the table 3.13, it is possible to visualize a positive impact on accuracy. After removing the volume of daily tweets and the volume of Bitcoin transactions from the dataset, a performance of around 57% is obtained, which validates that the volume of tweets carried out daily does not directly correlate with the trend, as well as the volume traded on Bitcoin. It should be noted that, despite having been analyzed in 3.5.2 a correlation of positive sentiment with the trend and validating that neutral sentiment remains uniform throughout the timeline, neutral sentiment reveals to have a greater impact when predicting the Bitcoin's trend. Finally, the same experiments were carried out, but this time with the negative sentiment, present in table 3.14, to assess whether this sentiment can be more closely related to the trend.

Table 3.14: Random Forest Classifier Model Train With Only Negative Sentiment Data.

Experiment	Tweet Volume	Price	Volume	Trend	Accuracy
1	Yes	Yes	Yes	Yes	44.11%
2	No	Yes	Yes	Yes	44.12%
3	No	No	Yes	Yes	41.17%
4	No	Yes	No	Yes	38.23%

In the table 3.14, it is possible to visualize the impact of the negative sentiment, where it is better related to the positive sentiment. However, a lower performance was obtained compared to the neutral sentiment. Based on the results, it was possible to obtain a maximum performance of 44.11% with negative sentiment when using all dataset data. This contradicts that the volume of daily tweets and the volume traded in Bitcoin negatively influence the trend forecast.

After carrying out the before mentioned experiments, it was concluded that the data to obtain better performance was the use of neutral sentiment with the price of Bitcoin and the trend of the day. Following this model, where a performance of 57.35% was obtained, some more techniques described below were additionally applied to validate its performance, described below.

Accuracy is a good metric, however it is necessary to validate other metrics to verify if the model correctly classifies the trend classes of our dataset. To this end, a confusion matrix is applied, which aims to summarize the classification performance of the present algorithm. Calculating this matrix can give better visibility into the ratings that the model gets right and what types of mistakes it is making. To generate the confusion matrix, the confusion_matrix function belonging to the Scikit Learn library was used. The final result of the confusion matrix is shown in Figure 3.24.

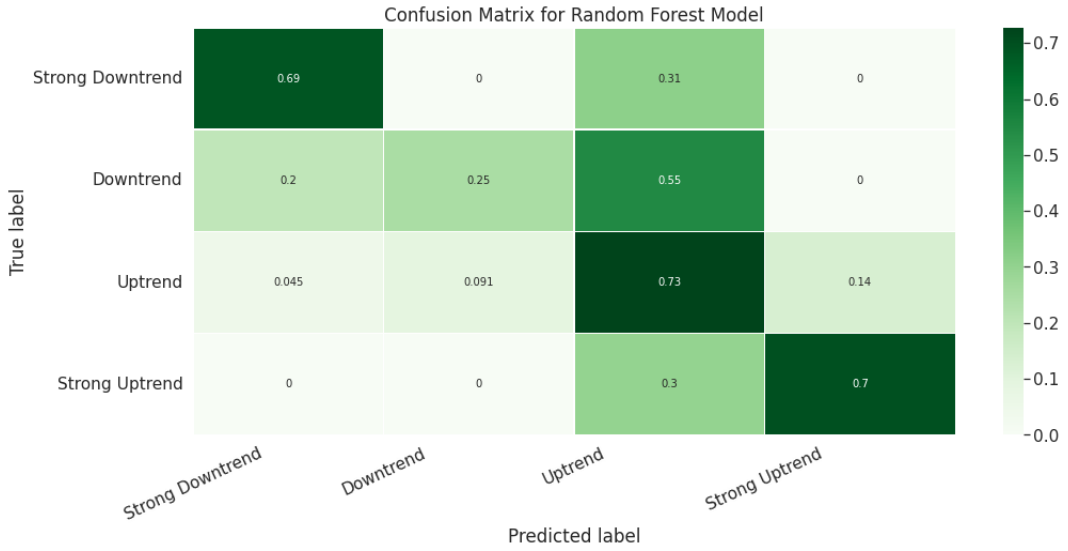

Figure 3.24: Random Forest Classifier Confusion Matrix.

In the figure 3.24 it is possible to visualize the four existing classes and the respective values represented in the table. It is also possible to verify the different cases where the model can perform a correct and incorrect classification. On the other hand, it allows the model to classify more assertively if the classification is strong downtrend, uptrend, downtrend or strong downtrend. However, there are cases where strong uptrend was actually classified as uptrend. An essential piece of information obtained in this model is that in the case of a downtrend, the model cannot classify as assertively. In the case of Uptrend classifications, there are several situations where the model classifies as downtrend, as well as strong uptrend and strong downtrend.

Table 3.15: Random Forest Classifier Precision, Recall and F1 Score Measures.

	Precision	**Recall**	**F1 Score**
Strong Downtrend	0.69	0.69	0.69
Downtrend	0.71	0.25	0.37
Uptrend	0.46	0.73	0.56
Strong Uptrend	0.70	0.70	0.70
Accuracy			0.57
Macro Avg	0.64	0.59	0.58
Weighted Avg	0.62	0.57	0.56

Then, we proceeded to analyse metrics related to precision, recall, F1 score and support. These values are represented in the table 3.15.

Precision is a ratio of correctly predicted values to the total of optimistic predictions. This metric intends to answer: how many observations were classified, for example, as an uptrend, how many were true? High precision is relative to how many false positives there are. In this metric, values around 0.70 were obtained for the strong downtrend, downtrend and strong uptrend classifications, which is good for this model. However, a precision of 0.46 was obtained for the uptrend classifications, which is below 0.50. This means that if the classification is an uptrend, the present model will not be accurate. However, the average is above 0.60, which is a good value.

Subsequently, the recall was analyzed, which is the ratio of correct predictions to the total number of observations of the same class. For this metric, a value of around 0.70 was obtained again for the Strong downtrend, downtrend and strong uptrend classifications, reflecting a good result. For the uptrend type classification, a value of 0.25 represents a very low value. In relation to the average, it presented a value of 0.59, translating into an acceptable value above 0.50.

Then, the F1 score was calculated, consisting of the weighted average of precision and recall. This metric takes into account both false positives and false negatives. In some situations, this metric is more valuable than accuracy. In particular, for the classification performed in this model, it makes sense to observe the F1 score to better evaluate the model. The average of the current F1 score recorded a value of 0.58, 0.01 above the accuracy, ensuring the use of our model to make and make good predictions, always taking into account the cadence of performance in classifying trends of the downtrend type.

Finally, a graphical analysis of the forecasts over time was carried out together with the actual data. In the figure 3.25, it is possible to visualize the training data together with the test data and the respective prediction, and in the figure 3.26, the predictions made by the model in more detail and compare with the actual ratings.

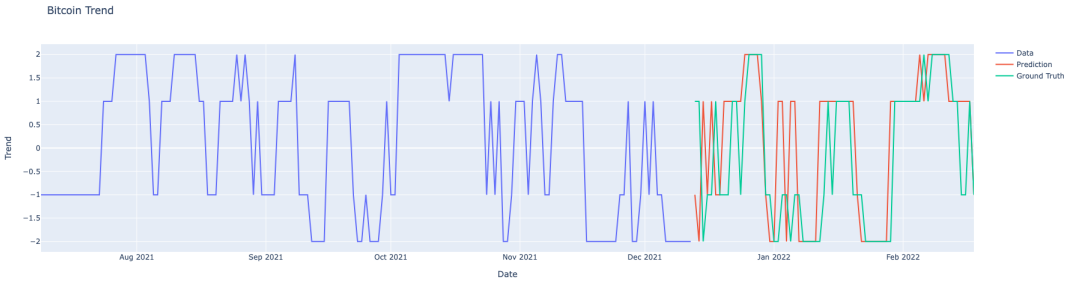

Figure 3.25: Random Forest Classifier Timeline Predictions.

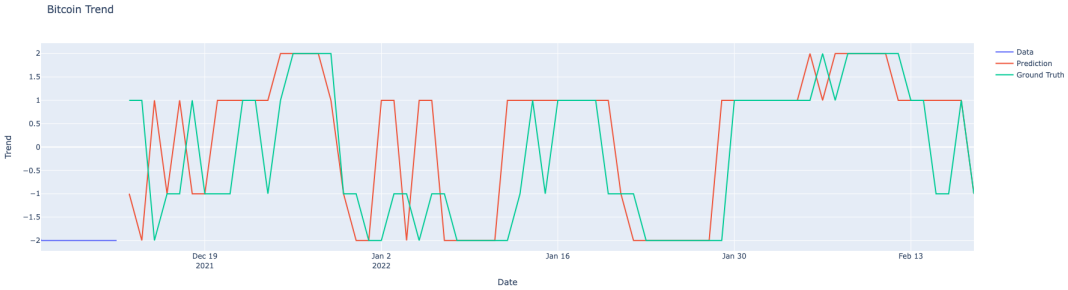

Figure 3.26: Random Forest Classifier Real Values vs Predicted Values.

In short, it became possible to analyze several metrics in detail to validate where the model gets it right and where it fails more often. This analysis is very important since accuracy alone does not provide enough information about the behaviour of the model. In particular, the confusion matrix proved to be a crucial tool to analyze which trends the model fails and how it obtains a reasonable classification. In general, there are some aspects to improve in the model. However, a good performance is obtained, registering a value above 50%, which is a positive factor.

Chapter 4

Conclusion And Future Work

The study carried out in this master's was mainly to answer some existing problems in sentiment analysis and forecasting trends in the bitcoin market. It was discovered that by analysing the general public's sentiment on Twitter's social network, it is possible to optimize the prediction of the bitcoin trend for the next day. Some of the existing forecasting methods are based only on market indicators, which is a limitation. This chapter begins by exploring the contributions of this study in light of the objectives outlined. The hypotheses formulated at the beginning of this study are then discussed to see whether they have been verified or rejected. Next, a summary of the tasks performed during this study is presented. Finally, some considerations are made and possibilities for future work are taken into account.

4.1 Contributions

The work presented in this study has a good number of contributions. The path taken to study the hypotheses listed in sections 3.4 and 3.5, led to the development of a set of models/methods that have become important contributions to the scientific world. The table 4.1 presents the objectives achieved with the work carried out and the document section in which they are described.

Next, the contributions that emerge from this study are listed and described.

1. Identification of correlation between sentiments with the bitcoin trend – The first contribution of this study was the one that started its development. When you started reading about predicting bitcoin prices and trends, several points were found that could be improved. From articles focused only on tweet volumes, others focused solely on market indicators, to those focused on feelings expressed on social networks. The combination of several techniques, from sentiment analysis, analysis of sentiment volumes, and the application of indicators such as the simple moving average to calculate bitcoin trends, made it possible to demonstrate the real impact of sentiment on the bitcoin market. This contribution made the **Objective 1** complete.
2. A model capable of classifying feelings in tweets – This contribution is described in the section 3.4 which aims to build a model of AI capable of classifying feelings where the accuracy of 87%. Since these models were built based on public tweets, it makes the model capable of predicting sentiments related to the bitcoin topic and general topics. Considering three different models were built, an LSTM, a BI-LSTM and a CNN, it brings to the research community various techniques and comparisons carried out during this study. This contribution helped achieve the objective **Objective 2**.

Table 4.1: List of defined objectives and respective sections where they were discussed.

Objective	Section
Objective 1: Understand state of the art in the following areas: Neural networks, recurrent neural networks, convolutional neural networks, sentiment analysis and Bitcoin trend prediction.	2.1, 2.2, 2.3, 2.4 e 2.5
Objective 2: Develop a model that can classify sentiment in tweets according to user intentions.	3.4.1, 3.4.2, 3.4.3, 3.4.4, 3.4.5, 3.4.6 e 3.4.7
Objective 3: Develop a model capable of predicting a trend, positive or negative, for the next day according to current market values.	3.5.1, 3.5.2, 3.5.3, 3.5.5 e 3.5.6
Objective 4: Demonstrate that the sentiment expressed on the Twitter network impacts the prediction of the Bitcoin trend.	3.5.6

3. A model capable of predicting future Bitcoin trends based on sentiments expressed on the twitter network – This contribution is described in the 3.5 section, which aims to demonstrate various AI models capable of demonstrating the effectiveness of different techniques for forecasting trends in bitcoin. Since these models were trained with different input data, it demonstrates which ones are more related to the bitcoin trend and which are less related. This contribution helped realize **Objective 3** and complete the results for **Objective 4**.
4. Demonstrate the impact of sentiment when it correlates with the trend of Bitcoin – It is obtained based on the results of the 3.5.6 section and allowed to prove a relationship between the sentiment expressed with the bitcoin trend. It was possible to demonstrate that the volume of tweets and transactions carried out daily in bitcoin does not directly influence the price for the next day. Furthermore, it was possible to demonstrate that particular sentiments have more impact on trend prediction than when used together. This contribution allowed the completion of **Objective 4**.

4.2 Validation of the Research Hypotheses

This study was developed following the scientific method. The methods demonstrated and the experiments carried out in Chapter 3 aimed to answer the hypotheses formulated in Section 1.2 of this document. These hypotheses are discussed and validated in this section.

In general, it was possible to validate and prove with several metrics extracted during the experimentation process, that the sentiment expressed on the Twitter network positively influences the prediction of the Bitcoin trend. The sentiment obtained through the sentiment classification model proved to be a crucial element, with the ability to make predictions with

high accuracy. Therefore, the existence of a correlation with Bitcoin's historical data was confirmed. On the other hand, it was possible to validate that after removing the sentiment data, the Bitcoin trend prediction model showed a cadence of 10% to 20%, confirming again the power of sentiment in predicting future trends. Then, it was validated that each typology of feelings, in isolation, also allows to positively influence the trend forecast, emphasizing that neutral sentiment achieves the best performance, compared to positive and negative sentiment. Finally, it was possible to prove that the volume of tweets performed does not directly influence the trend forecast, which reinforces the need to previously classify them as positive, neutral and negative, to obtain more concrete data that best fits with the Bitcoin trend.

4.3 Final Remarks and Future Work Considerations

The present study has the importance of applying AI techniques in the scope of Bitcoin trend prediction, taking into account the power of the Twitter social network. It has successfully demonstrated that the social sentiment impact positively affects the Bitcoin trend. During the development of this study, several AI techniques were used. From implementing models capable of performing the sentiment analysis in tweets to implementing two AI models to forecast Bitcoin's trends. An implementation and connection of two AI models demonstrated how to perform the inference process and maintain the separation of concepts between Bitcoin sentiment analysis and trend prediction.

The importance of the analysis of articles developed by other researchers should be highlighted. This analysis allowed different perspectives on implementing AI models and what had already been researched by other researchers. This allowed to get a lot of knowledge about the area, existing difficulties and obstacles, and data and techniques to be used to obtain better results.

The study results showed that the LSTM and BI-LSTM models were more successful in investigating the prediction of sentiments in tweets extracted from the Twitter network. However, the model based on the CNN network proved to be efficient in performing text analysis. The application of a supervised technique, the random forest classifier, has demonstrated consistency and stability in predicting Bitcoin trends. It was essential to discover that the opinion of several individuals can influence as much or more than the opinion of a single individual. Moreover, the sentimental impact of a group of people can positively influence the Bitcoin trend forecast.

The results reinforce that Bitcoin trend prediction involves several variables, apart from existing market indicators such as the volume of daily Bitcoin transactions or the use of techniques such as Simple Moving Average. In addition, social sentiment proved to be another variable that can be considered when making trend predictions in Bitcoin.

In terms of limitations, there were several difficulties in carrying out the present study. One first difficulty was to obtain historical tweets to perform inference with the first implemented model. Several steps took a lot of time and computational resources to be completed. The complexity of implementing LSTM networks for forecasting trends in time series datasets is also worth mentioning.

In future work, we propose developing deeper models of neural networks capable of predicting datasets based on time series to validate the impact of other AI techniques. It is also proposed to deploy the models developed for online environments, so that daily data collection

is carried out in real time without human intervention and that the training of the models becomes online learning. Thus, obtaining an interconnected architecture capable of making predictions autonomously in an online environment would be possible. Finally, developing a graphical interface to better visualize forecasts, so that other people, such as day traders, can optimize their decision-making when investing in the Bitcoin cryptocurrency market.

Bibliography

Abiodun, O. I. et al. (2018). "State-of-the-art in artificial neural network applications: A survey". In: *Heliyon* 4, p. 938. doi: 10.1016/j.heliyon.2018. url: https://doi.org/10.1016/j.heliyon.2018.e00938 (visited on 06/11/2022).

Abraham, J., K. Dowling, and S. Florentine (Dec. 2018). "Influence of controlled burning on the mobility and temporal variations of potentially toxic metals (PTMs) in the soils of a legacy gold mine site in Central Victoria, Australia". In: *Geoderma* 331, pp. 1–14. issn: 00167061. doi: 10.1016/j.geoderma.2018.06.010. (Visited on 04/23/2022).

Ahmed, W., P. Bath, and G. Demartini (2017). "Using Twitter as a data source: An overview of ethical, legal and methodological challenges". In: pp. 1–25. (Visited on 06/18/2022).

Alqaryouti, O. et al. (2019). "Aspect-based sentiment analysis using smart government review data". In: *Applied Computing and Informatics*. issn: 22108327. doi: 10.1016/j.aci.2019.11.003. (Visited on 05/29/2022).

Audrino, F., F. Sigrist, and D. Ballinari (Apr. 2020). "The impact of sentiment and attention measures on stock market volatility". In: *International Journal of Forecasting* 36 (2), pp. 334–357. issn: 01692070. doi: 10.1016/j.ijforecast.2019.05.010. (Visited on 06/12/2022).

Behdenna, S., F. Barigou, and G. Belalem (2016). "Sentiment analysis at document level". In: *Communications in Computer and Information Science* 628 CCIS, pp. 159–168. issn: 18650929. doi: 10.1007/978-981-10-3433-6_20. (Visited on 05/21/2022).

Birjali, M., A. Beni-Hssane, and M. Erritali (2018). "Learning with big data technology: The future of education". In: *Advances in Intelligent Systems and Computing* 565, pp. 209–217. issn: 21945357. doi: 10.1007/978-3-319-60834-1_22. (Visited on 05/29/2022).

Birjali, M., A. Beni-Hssane, and E. Mohammed (Apr. 2017). "Measuring Documents Similarity in Large Corpus using MapReduce Algorithm". In: (visited on 05/07/2022).

Birjali, M., M. Kasri, and A. Beni-Hssane (Aug. 2021). "A comprehensive survey on sentiment analysis: Approaches, challenges and trends". In: *Knowledge-Based Systems* 226. issn: 09507051. doi: 10.1016/j.knosys.2021.107134. (Visited on 06/12/2022).

Bollen, J., H. Mao, and X. Zeng (Mar. 2011). "Twitter mood predicts the stock market". In: *Journal of Computational Science* 2 (1), pp. 1–8. issn: 18777503. doi: 10.1016/j.jocs.2010.12.007. (Visited on 01/29/2022).

Carvalho, B. V. and C. H. P. Mello (2011). "Scrum agile product development method - literature review, analysis and classification". In: *Product Management & Development* 9 (1), pp. 39–49. issn: 16764056. doi: 10.4322/pmd.2011.005. (Visited on 06/12/2022).

Cavalli, S. and M. Amoretti (Mar. 2021). "CNN-based multivariate data analysis for bitcoin trend prediction". In: *Applied Soft Computing* 101. issn: 15684946. doi: 10.1016/j.asoc.2020.107065. (Visited on 05/06/2022).

Chaturvedi, I. et al. (Nov. 2018). "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges". In: *Information Fusion* 44, pp. 65–77. issn: 15662535. doi: 10.1016/j.inffus.2017.12.006. (Visited on 04/23/2022).

Chen, T. et al. (Apr. 2017). "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN". In: *Expert Systems with Applications* 72, pp. 221–230. issn: 09574174. doi: 10.1016/j.eswa.2016.10.065. (Visited on 05/22/2022).

Choi, Y. and H. Lee (Oct. 2017). "Data properties and the performance of sentiment classification for electronic commerce applications". In: *Information Systems Frontiers* 19 (5), pp. 993–1012. issn: 15729419. doi: 10.1007/s10796-017-9741-7. (Visited on 04/23/2022).

Coinmarketcap (2022). *Coinmarketcap*. url: https://coinmarketcap.com/ (visited on 06/18/2022).

Deloitte (2017). "2017 Global Mobile Consumer Survey". In: 1, p. 18. (Visited on 01/15/2022).

Do, H. H. et al. (Mar. 2019). "Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review". In: *Expert Systems with Applications* 118, pp. 272–299. issn: 09574174. doi: 10.1016/j.eswa.2018.10.003. (Visited on 05/22/2022).

Erritali, M. et al. (2016). "An approach of semantic similarity measure between documents based on big data". In: *International Journal of Electrical and Computer Engineering* 6 (5), pp. 2454–2461. issn: 20888708. doi: 10.11591/ijece.v6i5.10853. (Visited on 05/08/2022).

Frizzo-Barker, J. et al. (Apr. 2020). "Blockchain as a disruptive technology for business: A systematic review". In: *International Journal of Information Management* 51. issn: 02684012. doi: 10.1016/j.ijinfomgt.2019.10.014. (Visited on 06/04/2022).

Garcia, D. and F. Schweitzer (Sept. 2015). "Social signals and algorithmic trading of Bitcoin". In: *Royal Society Open Science* 2 (9). issn: 20545703. doi: 10.1098/rsos.150288. (Visited on 04/30/2022).

Georgoula, I. et al. (2015). *Using Time-Series and Sentiment Analysis to detect the Determinants of Bitcoin Prices*, pp. 1–14. url: http://ssrn.com/abstract=2607167 (visited on 01/01/2022).

Giachanou, A. and F. Crestani (June 2016). "Like it or not: A survey of Twitter sentiment analysis methods". In: *ACM Computing Surveys* 49 (2). issn: 15577341. doi: 10.1145/2938640. (Visited on 01/30/2022).

Gramoli, V. (June 2020). "From blockchain consensus back to Byzantine consensus". In: *Future Generation Computer Systems* 107, pp. 760–769. issn: 0167739X. doi: 10.1016/j.future.2017.09.023. (Visited on 05/08/2022).

Graves, A., S. Fernández, and J. Schmidhuber (2005). "Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition." In: 1, p. 7. (Visited on 01/22/2022).

Greaves, A. and B. Au (2015). *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*, pp. 1–8. (Visited on 05/06/2022).

Guo, H. et al. (Dec. 2021). "Bitcoin price forecasting: A perspective of underlying blockchain transactions". In: *Decision Support Systems* 151. issn: 01679236. doi: 10.1016/j.dss.2021.113650. (Visited on 06/11/2022).

Hao, P. et al. (Jan. 2021). "Predicting stock price trends based on financial news articles and using a novel twin support vector machine with fuzzy hyperplane". In: *Applied Soft Computing* 98. issn: 15684946. doi: 10.1016/j.asoc.2020.106806. (Visited on 06/11/2022).

Huang, X. et al. (Mar. 2021). "LSTM Based Sentiment Analysis for Cryptocurrency Prediction". In: p. 4. (Visited on 01/16/2022).

Hutto, C. J. and E. Gilbert (2014). *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. url: http://sentic.net/ (visited on 05/06/2022).

Ji, D. H. et al. (2013). *LNAI 7717 - Active Learning on Sentiment Classification by Selecting Both Words and Documents*, pp. 49–57. (Visited on 05/07/2022).

Jing, T. W. and R. K. Murugesan (2019). "A theoretical framework to build trust and prevent fake news in social media using blockchain". In: *Advances in Intelligent Systems and Computing* 843, pp. 955–962. issn: 21945357. doi: 10.1007/978-3-319-99007-1_88. (Visited on 06/05/2022).

Kaggle (2022). *Kaggle*. url: https://www.kaggle.com/ (visited on 06/18/2022).

Kaminski, J. (June 2014). "Nowcasting the Bitcoin Market with Twitter Signals". In: pp. 1–16. url: http://arxiv.org/abs/1406.7577 (visited on 01/01/2022).

Karalevicius, V. (2018). "Using sentiment analysis to predict interday Bitcoin price movements". In: *Journal of Risk Finance* 19 (1), pp. 56–75. issn: 09657967. doi: 10.1108/JRF-06-2017-0092. (Visited on 04/09/2022).

Kim, Y. (2014). "Convolutional Neural Networks for Sentence Classification". In: p. 6. doi: 10.48550/ARXIV.1408.5882. url: https://arxiv.org/abs/1408.5882 (visited on 01/15/2022).

Kraaijeveld, O. and J. Smedt (Mar. 2020). "The predictive power of public Twitter sentiment for forecasting cryptocurrency prices". In: *Journal of International Financial Markets, Institutions and Money* 65. issn: 10424431. doi: 10.1016/j.intfin.2020.101188. (Visited on 06/04/2022).

Kulkarni, P., S. Londhe, and M. Deo (2017). "Artificial Neural Networks for Construction Management: A Review". In: 1, p. 19. (Visited on 01/22/2022).

Lecun, Y. et al. (1998). "Gradient-Based Learning Applied to Document Recognition". In: 1, p. 47. (Visited on 01/23/2022).

Li, T. et al. (2017). "More than just noise? Examining the information content of stock microblogs on financial markets". In: doi: 10.1057/s41265-016. url: https://doi.org/10.1057/s41265-016- (visited on 01/30/2022).

Li, X. et al. (June 2020). "A survey on the security of blockchain systems". In: *Future Generation Computer Systems* 107, pp. 841–853. issn: 0167739X. doi: 10.1016/j.future.2017.08.020. (Visited on 05/07/2022).

Linardatos, P. and S. Kotsiantis (Jan. 2020). "Bitcoin Price Prediction Combining Data and Text Mining". In: pp. 49–63. (Visited on 05/06/2022).

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. 5th ed. Vol. 5. Bing Liu, pp. 1–167. (Visited on 05/21/2022).

Lyu, H. et al. (Dec. 2021). "Sense and Sensibility: Characterizing Social Media Users Regarding the Use of Controversial Terms for COVID-19". In: *IEEE Transactions on Big Data* 7.06, pp. 952–960. issn: 2332-7790. doi: 10.1109/TBDATA.2020.2996401. (Visited on 01/09/2022).

Madan, I., S. Saluja, and A. Zhao (2014). *Automated Bitcoin Trading via Machine Learning Algorithms*, pp. 1–5. (Visited on 05/06/2022).

Mai, F. et al. (2015). *From Bitcoin to Big Coin: The Impacts of Social Media on Bitcoin Performance*. (Visited on 01/30/2022).

Mäntylä, M. V., D. Graziotin, and M. Kuutila (2018). "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers". In: *Computer Science Review* 27, pp. 16–32. issn: 15740137. doi: 10.1016/j.cosrev.2017.10.002. (Visited on 05/21/2022).

Mao, H., S. Counts, and J. Bollen (Dec. 2011). "Predicting Financial Markets: Comparing Survey, News, Twitter and Search Engine Data". In: url: http://arxiv.org/abs/1112.1051 (visited on 01/29/2022).

Marston, S. et al. (Apr. 2011). "Cloud computing - The business perspective". In: *Decision Support Systems* 51 (1), pp. 176–189. issn: 01679236. doi: 10.1016/j.dss.2010.12.006. (Visited on 05/29/2022).

Matta, M., M. I. Lunesu, and M. Marchesi (2015). *Bitcoin Spread Prediction Using Social And Web Search Media Governing the smart city: a governance-centred approach to smart urbanism View project Fast wavelet transform assisted predictors of streaming time series View project Bitcoin Spread Prediction Using Social And Web Search Media*. url: https://www.researchgate.net/publication/279917417 (visited on 06/11/2022).

McNally, S., J. Roche, and S. Caton (June 2018). "Predicting the Price of Bitcoin Using Machine Learning". In: *Proceedings - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018*, pp. 339–343. doi: 10.1109/PDP2018.2018.00060. (Visited on 05/06/2022).

Medhat, W., A. Hassan, and H. Korashy (Dec. 2014). "Sentiment analysis algorithms and applications: A survey". In: *Ain Shams Engineering Journal* 5 (4), pp. 1093–1113. issn: 20904479. doi: 10.1016/j.asej.2014.04.011. (Visited on 05/28/2022).

Minaee, S., E. Azimi, and A. Abdolrashidi (Apr. 2019). "Deep-Sentiment: Sentiment Analysis Using Ensemble of CNN and Bi-LSTM Models". In: p. 6. (Visited on 01/16/2022).

Mittal, A. et al. (Aug. 2019). "Short-Term Bitcoin Price Fluctuation Prediction Using Social Media and Web Search Data". In: pp. 1–6. (Visited on 05/06/2022).

Nakamoto, S. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*, pp. 1–11. url: www.bitcoin.org (visited on 05/08/2022).

O'connor, B. et al. (2010). *From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series*. url: http://www.sca.isr.umich. (visited on 05/14/2022).

Pant, D. R. et al. (Oct. 2018). "Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis". In: pp. 128–132. doi: 10.1109/CCCS.2018.8586824. (Visited on 01/09/2022).

Pimprikar, R., S. Ramachandran, and K. Senthilkumar (2017). "USE OF MACHINE LEARNING ALGORITHMS AND TWITTER SENTIMENT ANALYSIS FOR STOCK MARKET PREDICTION". In: 1, p. 6. (Visited on 01/15/2022).

Ramírez-Tinoco, F. J. et al. (2018). "A brief review on the use of sentiment analysis approaches in social networks". In: *Advances in Intelligent Systems and Computing* 688, pp. 263–273. issn: 21945357. doi: 10.1007/978-3-319-69341-5_24. (Visited on 05/15/2022).

Rognone, L., S. Hyde, and S. S. Zhang (May 2020). "News sentiment in the cryptocurrency market: An empirical comparison with Forex". In: *International Review of Financial Analysis* 69. issn: 10575219. doi: 10.1016/j.irfa.2020.101462. (Visited on 06/05/2022).

Sattarov, O. et al. (Nov. 2020). "Forecasting bitcoin price fluctuation by twitter sentiment analysis". In: *2020 International Conference on Information Science and Communications Technologies, ICISCT 2020*. doi: 10.1109/ICISCT50599.2020.9351527. (Visited on 01/01/2022).

Schuller, B., A. E. D. Mousa, and V. Vryniotis (Sept. 2015). "Sentiment analysis and opinion mining: On optimal parameters and performances". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5 (5), pp. 255–263. issn: 19424795. doi: 10.1002/widm.1159. (Visited on 04/24/2022).

Schumaker, R. and H. Chen (Feb. 2009). "Textual analysis of stock market prediction using breaking financial news: The AZFin text system". In: *ACM Trans. Inf. Syst.* 27. doi: 10.1145/1462198.1462204. (Visited on 01/15/2022).

Sprenger, T. O. and I. M. Welpe (2010). *Tweets and Trades: The Information Content of Stock Microblogs*. url: http://ssrn.com/abstract=1702854 (visited on 01/29/2022).

Stenqvist, E. and J. Lönnö (2017). *Predicting Bitcoin price fluctuation with Twitter sentiment analysis*. (Visited on 05/27/2022).

Symeonidis, S., D. Effrosynidis, and A. Arampatzis (2018). "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis". In: *Expert Systems with Applications* 110, pp. 298–310. issn: 0957-4174. doi: https://doi.org/10.1016/j.eswa.2018.06.022. url: https://www.sciencedirect.com/science/article/pii/S0957417418303683 (visited on 01/08/2022).

Tubishat, M., N. Idris, and M. A.M. Abushariah (July 2018). "Implicit aspect extraction in sentiment analysis: Review, taxonomy, oppportunities, and open challenges". In: *Information Processing and Management* 54 (4), pp. 545–563. issn: 03064573. doi: 10.1016/j.ipm.2018.03.008. (Visited on 05/22/2022).

Twitter (2022a). *Twitter Privacy Policy*. url: https://twitter.com/en/privacy (visited on 06/18/2022).

– (2022b). *Twitter Terms of Service*. url: https://twitter.com/en/tos (visited on 06/18/2022).

Valdivia, A. et al. (Nov. 2018). "Consensus vote models for detecting and filtering neutrality in sentiment analysis". In: *Information Fusion* 44, pp. 126–135. issn: 15662535. doi: 10.1016/j.inffus.2018.03.007. (Visited on 05/07/2022).

Xu, Y. and V. Keselj (2019). "Stock Prediction using Deep Learning and Sentiment Analysis". In: pp. 5573–5580. doi: 10.1109/BigData47090.2019.9006342. (Visited on 01/16/2022).

Yahoo (2022). *Yahoo Finance*. url: https://finance.yahoo.com/ (visited on 06/19/2022).

Yaqoob, I. et al. (Dec. 2016). "Big data: From beginning to future". In: *International Journal of Information Management* 36 (6), pp. 1231–1247. issn: 02684012. doi: 10.1016/j.ijinfomgt.2016.07.009. (Visited on 05/28/2022).

Zhang, X., H. Fuehres, and P. A. Gloor (2011). "Predicting Stock Market Indicators Through Twitter "I hope it is not as bad as I fear"". In: *Procedia - Social and Behavioral Sciences* 26, pp. 55–62. issn: 18770428. doi: 10.1016/j.sbspro.2011.10.562. (Visited on 01/30/2022).