

Multimodal Approach for Cryptocurrency Price Prediction

Master's Thesis

in partial fulfillment of the requirements for
the degree of Master of Science (M.Sc.)
in Web and Data Science

submitted by
Azeddine Bouabdallah

First supervisor: Prof. Dr. Jan Jürjens
Institute for Software Engineering

Second supervisor: Dr. Zeyd Boukhers
Institute for Web Science and Technologies

Koblenz, February 2022

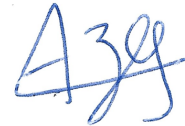
Statement

I hereby certify that this thesis has been composed by me and is based on my own work, that I did not use any further resources than specified – in particular no references unmentioned in the reference section – and that I did not submit this thesis to another examination before. The paper submission is identical to the submitted electronic version.

	Yes	No
I agree to have this thesis published in the library.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
I agree to have this thesis published on the Web.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
The thesis text is available under a Creative Commons License (CC BY-SA 4.0).	<input checked="" type="checkbox"/>	<input type="checkbox"/>
The source code is available under a GNU General Public License (GPLv3).	<input checked="" type="checkbox"/>	<input type="checkbox"/>
The collected data is available under a Creative Commons License (CC BY-SA 4.0).	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Koblenz, 01/02/2022

(Place, Date)



(Signature)

Note

- If you would like us to contact you for the graduation ceremony,
please provide your personal E-mail address: **bouabdallahazeddine@gmail.com**
- If you would like us to send you an invite to join the WeST Alumni
and Members group on LinkedIn, please provide your LinkedIn ID : **bouabdallahazeddine**

Zusammenfassung

Trotz der in den letzten Jahren erzielten Verbesserungen bei der Genauigkeit von Kryptowährungspreisvorhersagen ist dieser Bereich noch weit davon entfernt, ein Standardthema zu sein. Im Gegensatz zu traditionellen Märkten werden die Preise von Kryptowährungen direkt von mehreren Faktoren beeinflusst. Einige der Einflüsse sind die Korrelation zwischen Kryptowährungen und dem globalen Markt, das öffentliche Bewusstsein und die Hash-Rate. Diese Arbeit schlägt DMCCrypt vor, einen multimodalen AdaBoost-LSTM-Ensemble-Lernansatz, um das Problem der Kryptowährungspreisvorhersagen unter Verwendung aller Modalitäten, die die Preisschwankungen antreiben, wie Social-Media-Stimmungen, Suchvolumen, Blockchain-Informationen und Handelsdaten, zu lösen. Experimentelle Ergebnisse zeigen eine vielversprechende Verbesserung der Preisvorhersagen gegenüber anderen State-of-the-Art-Ansätzen mit einem durchschnittlichen RMSE-Rückgang von \$38 (19,29 % Verbesserung). Ausführliche Experimente zeigen außerdem die Bedeutung jeder Multimodalität für die Gesamtleistung des Modells, so dass das Hinzufügen von Blockchain-Daten oder Social-Media-Stimmungen zum Modell die Vorhersagefehler erheblich verringert. Darüber hinaus schätzt DMCCrypt zusätzlich zur einzelnen Preisvorhersage die Verteilung des vorhergesagten Preises, um die Unsicherheit einer solchen Vorhersage zu modellieren und eine bessere Entscheidungshilfe zu bieten. Soweit ich weiß, kann dieser Ansatz als der erste angesehen werden, der alle Multimodalitäten, die die Preise von Kryptowährungen beeinflussen, kombiniert und eine Adaboost-LSTM-Ensemble-Lernarchitektur vorschlägt, die für ein solches Thema verwendet werden kann.

Abstract

Despite the recent improvements in cryptocurrency price predictions accuracy in the last few years, this field is still far from an off-the-shelf topic. Unlike traditional markets, cryptocurrency market prices are directly affected by several factors. Some of the influences are the correlation between cryptocurrency and the global market, public awareness, and the hash rate. This thesis proposes DMCCrypt, a multimodal AdaBoost-LSTM ensemble learning approach to tackle the problem of cryptocurrency price predictions using all the modalities driving the price fluctuations like social media sentiments, search volumes, blockchain information, and trading data. Experiment results show a promising improvement in price predictions over other state-of-the-art approaches with an average RMSE decrease of \$38 (19.29% improvement). Extensive experiments further demonstrate the importance of each modality to the overall performance of the model, such that adding the blockchain data or social media sentiments to the model decrease the prediction errors significantly. Moreover, as an addition to the single price prediction, DMCCrypt estimates the distribution of the predicted price to allow modeling the uncertainty of such prediction and provide better help for decision-making. To the best of my knowledge,

this approach can be considered the first to combine all the multimodalities influencing cryptocurrency prices and proposes an Adaboost-LSTM ensemble learning architecture to be used in such a topic.

Contents

1	Introduction	1
2	Background	4
2.1	Cryptocurrency Market vs. Traditional Market	4
2.2	Introduction to Deep Neural Networks	6
2.3	Deep Neural Networks for Sequential Data	7
2.4	Cost Functions	10
3	Related Work	13
3.1	Traditional Market Price Prediction	13
3.2	Machine Learning for Cryptocurrency Price Prediction	14
3.3	Sentiment Analysis for Cryptocurrency Price Prediction	16
4	Approach	18
4.1	Datasets	18
4.2	LSTM Proposed Approach	26
4.3	Adaptive Boosting and LSTM Ensemble Learning	28
4.4	Model Varieties and Dropouts	30
4.5	Predicted Price Distribution:	32
5	Experiments and Results	36
5.1	Experimental Setup	36
5.2	Baselines	37
5.3	Results and Discussion	39
6	Conclusion and Future Work	57
7	List of Acronyms	58
	List of Figures	59
	List of Tables	60
	References	61

1 Introduction

By January 2022, the most prominent cryptocurrencies measured a market value of almost \$1 trillion in market capitalization, with Bitcoin holding a dominance of 67.79%¹. According to the CoinDesk case study, the global cryptocurrency market is estimated to increase by 12.9% by 2030². The global cryptocurrency market grows at a compound annual growth rate (CAGR) at 30% from 2019 to 2026³. In only one year, the value of a single Bitcoin has increased by 795% going from \$7118 in April 2020 to \$56,608 in April 2021.

The unexpected growth in cryptocurrency prices over the years made it a valuable investment opportunity. Investors and businesses are diverting to cryptocurrency markets, intending to maximize profits and minimize losses. It is common to buy cryptocurrencies when prices are low and sell when prices are higher. Therefore, experts are constantly studying the market to better understand the trends within the price fluctuations [1, 2, 5, 22, 9, 18, 23].

Unlike traditional currencies whose fundamental value can be determined from the cash flows such as dividends and earnings, the core fundamentals of cryptocurrencies are different [22]. When it comes to traditional currencies, the financial system is being controlled by central authorities and banks. However, cryptocurrencies do not have a central authority. All transactions are validated and processed through network nodes via cryptography [10]. After that, they will be recorded in a blockchain, which is a public distributed ledger. With such decentralization, people remain anonymous throughout the transactions. In addition, most cryptocurrencies have a pre-determined and limited supply⁴. For example, Bitcoin comes with a maximum supply of 21 million, and once it reaches the limitation, there will be no new Bitcoin to be mined. All these differences cause its price changes to be tough to predict and still an area of debate [2].

Sentiments play a significant role in price evolution, given possible arbitrage opportunities and intangible fundamental value[1]. Public opinion and sentiments explain the volatility of market trends, especially cryptocurrencies[24]. Yang, et al[44] suggests that social media sentiment is an important leading indicator of future bitcoin price swings. This demonstrates that social media can significantly impact one of the giant market caps. According to a financial study made by Kristoufek [22],

¹<https://gadgets.ndtv.com/cryptocurrency/news/bitcoin-price-btc-cryptocurrency-market-crash-usd-1-trillion-coinmarketcap-2726233>

²<https://www.coindesk.com/markets/2021/08/25/cryptocurrency-market-will-more-than-triple-by-2030-study/>

³<https://www.globenewswire.com/news-release/2021/04/12/2208331/0/en/At-30-CAGR-CryptoCurrency-Market-Cap-Size-Value-Surges-to-Record-5-190-62-Million-by-2026-Says-Facts-Factors.html>

⁴<https://codecondo.com/why-are-cryptocurrencies-unique/>

public sentiments and awareness are not the only factors contributing to the instability of cryptocurrency prices. The hash rate of the cryptocurrency mining process and the correlation between the cryptocurrency and the global financial market also impact the price fluctuation.

This thesis is motivated by the assumption that cryptocurrency investors require a reliable price fluctuation prediction model using the trading data and sentiment analysis from social media, the blockchain information, and search volumes from search engines for better investment decision-making.

Numerous recent studies on cryptocurrency price prediction have used deep learning methods[23, 26, 33, 35, 45] and were able to achieve better results than traditional machine learning and statistical approaches [44, 23]. However, these approaches have a limitation in that they have not considering all the factors influencing the cryptocurrency market. The problem with supervised learning is usually formalized as inferring a forecast function based on the available training sets and then evaluating the obtained functions by how well it generalizes [35]. These inherent limitations in capturing and predicting cryptocurrency prices due to the assumption that price series often exhibit a homogeneous nonstationary. In reality, many factors contribute to the instability of cryptocurrency prices, such as the hash rate of the cryptocurrency mining process, the correlation between cryptocurrency and the global financial market, and public awareness[22].

Contributions, This thesis presents DMCCrypt – Deep multimodal cryptocurrency price prediction is a deep learning approach that employs multimodal data to make price predictions on the following 24th hour. The contributions of this thesis are the following.:

- Employ all the factors that drive the cryptocurrency market, such as trading data, social media sentiments, blockchain information, and search volumes.
- Provide a distribution for the predicted price as an output to help the user understand the certainty of the model's prediction.
- Improve the prediction performance of existing models and provide reliable help for decision-making to investors by proposing an Adaptive Boosting (AdaBoost)-Long Short Term Memory (LSTM) ensemble-learning architecture.
- Provide an open-source implementation and a simple demo web application to help users visualize the model's predictions.
- Conduct extensive experiments and analyses to test and validate the approach.

The cryptocurrency use case is Bitcoin because of its significant market domination and popularity among the other 7812 existing cryptocurrencies and the extensive data availability needed in this research.

In the following sections, section 2 provides a brief background section to get the reader familiar with basic concepts and topics used by this thesis. Section 3 provides an overview of the previous related works in the field of cryptocurrency price prediction. The subsequent section 4 details the necessary data preprocessing steps, this thesis approach, and the model's proposed architecture. Next, all experiments details and results are provided in section 5. Finally, section 6 concludes this thesis and discusses possible future work improvements.

2 Background

The background section contains the necessary details for all the topics a reader might need to get familiar with to follow this thesis' approach. It consists of a brief overview of the fundamental differences between the traditional and the cryptocurrency markets, an introduction to recurrent neural networks, a list of activation functions that DMCrypt adopts, and the multiple cost functions applied in this thesis

2.1 Cryptocurrency Market vs. Traditional Market

Recently, Cryptocurrencies have taken the world by storm. According to the Time.com, the total market value of all cryptocurrencies by the end of 2021 is astonishingly more than 3 trillion dollars ⁵. As a result, people, especially investors, have been switching their focus towards trading cryptocurrencies. Nevertheless, what made this market valuable and different from traditional investments such as stocks? We could write volumes on the nature of cryptocurrency and stock investments, this section will only briefly introduce the fundamental differences between cryptocurrency and traditional markets.

Stock markets:

Stock markets are the first thing that comes to an amateur investor's mind. They are the most common form of traditional investments. People have been grinding profits through them for the past century. It is essential to know the real definition of stocks to understand the differences between it and cryptocurrencies markets. Stocks represent an ownership interest in a public traded company or business. Each share of stocks an investor buys refers to a percentage of ownership in the company itself. Stockholders can earn money in two different ways[31]:

Through capital gain: investors sell their stock shares to others at a higher price.

Receiving dividends and cash flow: any stockholder can benefit from the long-term gains of a company through receiving dividends if a company provides this.

As a result, the price and overall performance of stocks are determined by the company's actual performance and success. The price can rise and fall with the rise and fall of the business or company. It is also important to know that unless the company goes bankrupt and closes, the stocks remain existing to some extent.

There are several main drivers of the stock markets' prices. The following are the main worth mentioning ones:

⁵<https://time.com/6115300/cryptocurrency-value-3-trillion/>

- The stock prices move (down or up) by investors' assessment of a company's performance in the future, for example if investors are optimistic and deem that the company is headed toward success, then it is high likely that the stock's price will rise in the future. Ultimately, the prices depend on the company's success and its ability to grow profits over the long term.
- Single or multiple stocks do not strongly dominate the market. The most dominants of this market are from the MANGA stocks (Meta, Amazon, Netflix, Google, Apple) with almost one-fifth of the entire SP 500⁶. However, this is not enough for a market to be strongly dominated by one party.
- Stock prices are not stable, however the volatility of this market is far less than cryptocurrencies. It tends to be predictable rendering it to be generally stable compared to the cryptocurrency market. Many stocks change prices over the long term. Mainly they can rise or fall roughly 100% in a one-year span.
- Central authorities and banks heavily regulate the traditional market. Almost all the trades are made through large and central exchanges, for example, the New York Stock Exchange. As a result, prices are regulated by a centralized pricing mechanic, such that there can be no unpredictable change over time.

Cryptocurrency markets:

In contrast to traditional markets, cryptocurrencies are purely digital assets. For instance, the Euro is backed by a physical component (money) that an owner can either use in a digital form or extract it in a physical form. However, cryptocurrencies are and can only be used as a digital component. The main two variants of cryptocurrencies are: *Pure currencies* such as Bitcoin (where investors can only sell, buy, or trade), and *Utility tokens* such as Ethereum (Investors can also sell, buy and trade these coins). The main difference between them is that utility token coins function as part of more complex software and can be used in other forms of assets, for example, NFT⁷.

The primary profit source of this market is the ability for an investor to buy coins when the prices are low and sell them when the prices are high; this is also known as capital gains. Essentially, investors can gain profit from cryptocurrency if they can get another investor to buy it at a higher price.

The cryptocurrency price drivers are far more complex and ambiguous than the traditional market due to their core fundamental differences. The following are the main fundamental drivers of the cryptocurrency prices:

- Cryptocurrencies do not represent physical assets and are not backed by cash flows, and there is no real asset to influence or stabilize the market. Instead,

⁶<https://qz.com/2108056/apple-amazon-microsoft-and-alphabet-drove-the-sp-500-in-2021/>

⁷<https://learn.eqonex.com/news/what-utility-token>

it relies upon the public's sentiments to drive its prices. If the public's opinions and sentiments favor cryptocurrencies, this will drive the market's price accordingly.

- In contrast to the traditional market where the price cannot vanish unless the company of the physical asset representing it goes out of business, The cryptocurrency prices can plummet to zero if all investors do not favor the coin and acknowledge its existence.
- Even though there are more than 10,000 cryptocurrencies, the whole market is actually tied up to a single cryptocurrency (Bitcoin) as it dominates more than 70% of the market.
- The cryptocurrency market is known to be the most volatile market; the prices can rise as much as three times and fall on the same day. For example, the price of a single Bitcoin dropped by more than 62% in just ten days from 8th of May 2021 until 19th of May 2021.
- Central authorities or banks do not regulate the market. Instead, all transactions are validated through the network. Essentially cryptocurrencies are traded directly between a sender and a receiver. The absence of such a third party means that there are no centralized pricing mechanics for the prices of cryptocurrencies, making the market highly volatile and unpredictable.

2.2 Introduction to Deep Neural Networks

Nowadays, deep learning power many aspects of our modern society [25]. As a result, its applications are widespread across multiple sciences, business, and government fields. Few of these applications include recommendation systems, speech recognition, and stock market price predictions, making it a major focus of many research works in the past decade. One of the limitations that machine learning methods impose is their lack of ability to process raw data. As a result, solving most problems requires domain expertise and intensive feature engineering, especially with complex problems such as time-series forecasting. Extracting meaningful information from raw data and transforming it into a feature vector is an absolute necessity because the performance of the model may be affected by the quality of the feature vector. On the other hand, deep learning provides various levels of representation learning, starting from the first level of raw data and moving into a higher and a more abstract level. This allows the model to uncover the representations needed for regression or classification. The two main categories of deep learning are *supervised learning* and *unsupervised learning*.

Activation functions:

Activation functions are a main and an important component of neural networks. All activation functions should have a non-zero derivative at each point in order for

the gradient decent to work. The following activations functions are the one used in this thesis:

- **Sigmoid:** is a linear activation function that transforms an input value into a value between the range of (0 to 1). The two main characteristics of this function is that it is differentiable and monotonic. It can be defined as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (1)$$

where $\sigma(x)$ denotes the *sigmoid* function of x .

- **Hyperbolic Tangent Activation Function (Tanh):** Similar to *Sigmoid*, *Tanh* is a linear function that takes a numerical input and returns a value between the range of (-1 to 1). It can be defined as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (2)$$

where, $\tanh(x)$ denotes the *Tanh* function of the value x

- **Rectified Linear Unit (ReLU):** this activation function returns the positive part of the input and rendering all the negative side to zero. It is defined as follows:

$$\text{ReLU}(x) = \max(0, x) \quad (3)$$

2.3 Deep Neural Networks for Sequential Data

Recurrent neural network (RNN) are a type of Artificial Neural Networks that are commonly used to tackle sequential problems. *RNNs* powerfully shine when tackling problems related to *Natural Language Processing (NLP)*, *Language and Speech Recognition*, *Image Captioning*, etc. Many research works have demonstrated that *RNNs* achieve promising results by outperforming other architectures on many challenging problems [13, 6, 42, 40, 21]. Given that *RNNs* was inspired by other artificial neural networks such as *Convolution Neural Networks (CNNs)* and *FeedForward Neural Networks (FNNs)*, there are some similarities in the way they operate such that they all use data to train and learn patterns and relations [46]. However, the main distinction is that *RNNs* have a memory within, such that prior inputs influence the current output. That is a great advantage, especially when working with sequential or time-series data. For instance, the prediction of the next word of a given sentence is influenced by the words behind the one we are predicting. In these situations, it is vital to have a memory that embeds the information of the previous states and

provides outputs based on that.

In traditional artificial neural networks, we usually deal with one input and output. However, working with sequential data can be problematic because we mostly have varying input and output lengths. As a result, different types of *RNNs* can be used depending on the use case. These types can be generalized into the following[8]:

1. **One-to-one:** where we have one input and one output.
2. **Many-to-one:** We have multiple inputs (sequence) and only one output. Like a prediction of the price of a stock market in the next day.
3. **One-to-many:** We have a single input and multiple outputs (sequence of outputs).
4. **Many-to-many:** We have multiple inputs (sequence) and outputs (sequence).

The one common characteristic between all the aforementioned architectures is the *RNN block/cell*, as demonstrated in Figure 1. The *RNN cell* is a component that takes a current input and previous memories to provide a current state known as a hidden state. The hidden state is then forwarded to the next *RNN BLOCK* (next timestep); it can also be used to provide an output at each timestep (*One-to-many* or *many-to-many*) [28].

The *RNN block's* hidden state is formally presented as follows:

$$\mathbf{h}^{(t)} = \tanh(W_{hx}\mathbf{x}^{(t)} + W_{hh}\mathbf{h}^{t-1} + b_h), \quad (4)$$

where t represent the current timestep t , W_{ht} and W_{hh} are matrices of length n and width m , and b_h denotes a vector of length n that represent hidden states biases. Moreover, the output of each *RNN block* can be presented as follows:

$$\hat{\mathbf{y}}^{(t)} = \text{SoftMax}(W_{yh}\mathbf{h}^{(t)} + b_y), \quad (5)$$

where $\hat{\mathbf{y}}^{(t)}$ represent the cell's output at the current timestep t , W_{yh} is a matrix of length n and width m , and b_y denotes a vector of length n that represent hidden states biases.

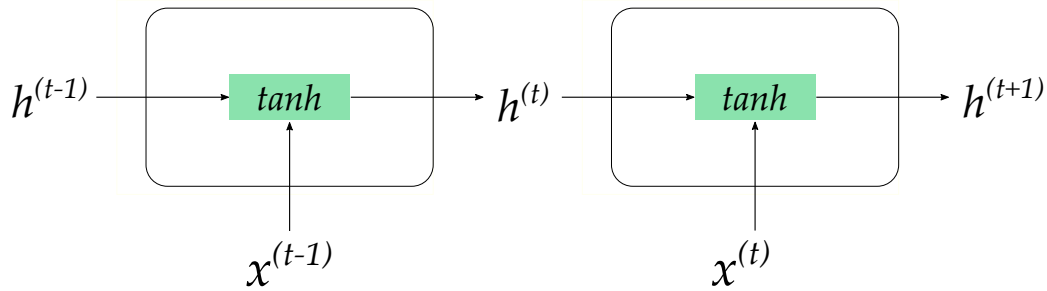


Figure 1: A Simplified Architecture of RNN cells [30]

Notice that the *RNN block's* architecture can be slightly changed to tackle a specific problem. The most common changes apply different activation functions than *tanh* and *SoftMax*.

Due to the dynamic architecture of *RNNs*, researchers have been trying to optimize and develop multiple variants that can serve specific purposes [7, 48, 12, 47]. The most popular variants are the following:

Bidirectional Recurrent Neural Networks (BRNN): In *RNNs*, the output is influenced directly by the current and previous inputs known as a memory. However, the following or future inputs do not affect the current input in any way possible. That can be problematic when the problem at hand requires such relations. Therefore, researchers have come up with *BRNNs* [7].

LSTM: Another limitation of applying *RNNs* is the memory loss during timesteps. *LSTMs* provide two memories to overcome this problem: long-term and short-term memories. The long-term memory captures the information that the network deems essential to keep, and the short-term memory captures the recent information from recent timesteps. Utilizing both memories allows the network to capture all relevant information from past states that influence the current output [12]. Figure 2 briefly illustrates the overall architecture of *LSTMs cells*.

Bidirectional Long short-term memory (BiLSTM): Similar to *BRNNs*, *BiLSTMs* were proposed to deal with the same problem, however, by applying *LSTM cells* instead of *RNN cells* [48].

Gated Recurrent Unit (GRU): Similarly with *LSTMs*, *GRUs* are proposed to tackle the vanishing memory that *RNNs* have. However, in contrast to *LSTM*, *GRU* consists of only two gates (combine gate, and update gate). In addition, *GRU* does not consist of a cell state. Instead, they use the previous hidden state (also known as a working memory) as a current cell state [47]. As a result, many research works found that the *GRUs* deal with long-term memories better than an *LSTM*[38]. In addition, because the *GRU* is less computationally complex than *LSTM*, the training

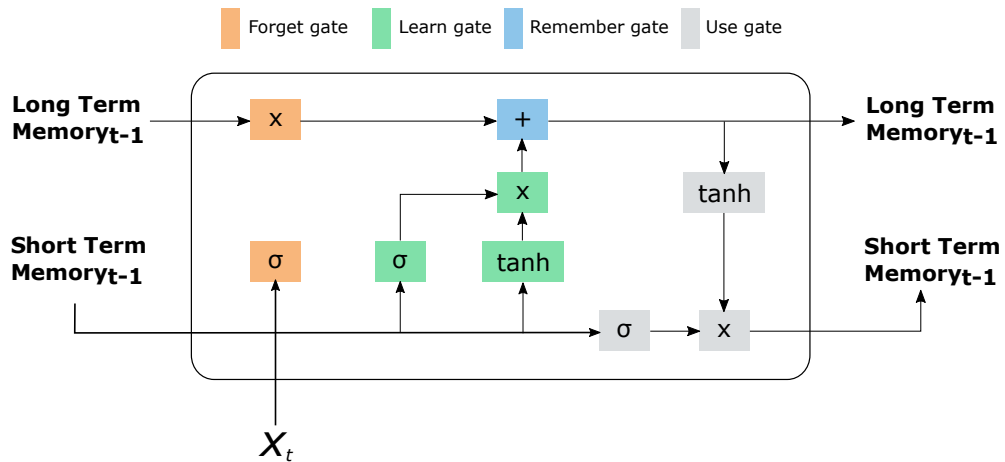


Figure 2: A Simplified Architecture of an LSTM cell [12]

and prediction time appears to be significantly lower when dealing with GRUs [38]. Figure 3 demonstrates the simplified architecture of the unit.

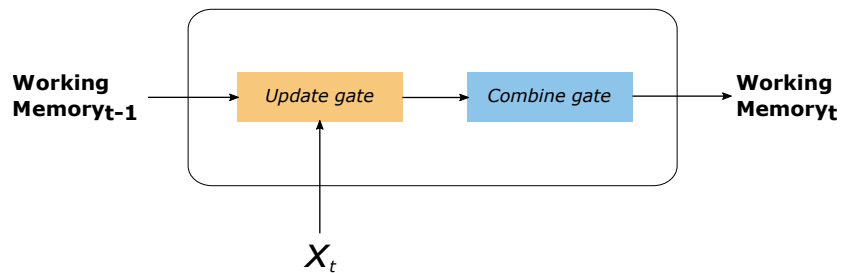


Figure 3: A Simplified Architecture of a GRU cell [47]

2.4 Cost Functions

The standard method for solving a regression problem and generalizing a model to work on test data is to apply and minimize a cost function while training. There are various approaches to measure the performance of the final predictions. One of the most common is to apply a cost function between the predicted and actual outcomes. These functions aim to convey how far a model's prediction is from the actual forecast. The term "Error function" is usually used in place of "Cost Function" to impart that a returned value is an error value of the prediction.

There exist multiple cost functions that are used for regression problems. The most common error functions and the ones that are used in this thesis are the following:

Sum of Squared Errors (SSE):

The traditional and most common function used for training is *the sum of squared errors (SSE)*. As the name suggests, this function returns the sum of the squared difference between the actual and predicted outcomes. The main benefit of this function is making the cost/error value more prominent, which helps during the training phase when working with small values. It is defined as follows:

$$SSE = \sum_{i=0}^n (y_i - \hat{y}_i)^2, \quad (6)$$

where \hat{y} is a vector of length n denotes the predicted outcomes, and y is a vector of length n denotes the tangible outcome.

Even though the *SSE* is widely used in traditional problems, it does not work in most everyday situations, especially when working with large amounts of data. The cost value will be significant even if the predicted outcome is close to the actual one. As a result, understanding this value while testing the model can be challenging.

L1 Loss:

The *L1 Loss* is a cost function that calculates the sum of the absolute differences between all the real and the forecasted outcomes. It is defined as follows:

$$L1 - Loss = \sum_{i=0}^n |y_i - \hat{y}_i| \quad (7)$$

Mean Square Error (MSE):

The *MSE* is a function that calculates the sum of squared differences between the forecast and actual outcome. It is defined as follows:

$$MSE = \frac{\sum_{i=0}^n (y_i - \hat{y}_i)^2}{n} \quad (8)$$

The *MSE* is always positive, with a value of 0, meaning that the forecast is identical to the actual outcome.

Although MSE is preferred in many cases, it cannot handle outliers because the effect of an outlier is enhanced due to applying a square to the difference. Other functions that use the absolute difference instead of squared difference are suggested to tackle this limitation.

Mean Absolute Error (MAE):

In contrast to the *MSE* function, the *MAE* is a function that calculates the sum of absolute differences between the forecast and actual outcome. Using the absolute instead of a square to deal with negative values can preserve the real difference between the values, which can come in handy when testing the results of a particular model. The *MAE* is defined as follows:

$$MAE = \frac{\sum_{i=0}^n |y_i - \hat{y}_i|}{n} \quad (9)$$

Root Mean Square Error (RMSE):

Similar to *MSE*, the *Root Mean Square Error (RMSE)* is the root of the *MSE*. The goal of this cost function is to cancel the squared differences by applying a square root to obtain an understandable cost value with the same scale as the predicted outcome values. It is common to use this function to test and validate regression models and compare them with other approaches. The *RMSE* is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=0}^n y_i - \hat{y}_i}{n}} \quad (10)$$

3 Related Work

Price predictions have always been a primary focus for financial and economic studies, as predicting the prices of a particular market in the future can give an enormous advantage for investors and businesses. Moreover, it can help decision-makers take the right actions at the right moment maximizing the profits and minimizing the tragic losses. Therefore, several research works and studies explored different possibilities and proposed multiple novel models to tackle the problem of cryptocurrency price predictions. Although all the efforts, this field is still far from being an off-the-shelf topic, such that many improvements can still be made to increase the efficiency of the predictions and further help decision-makers in their journey. This section further discusses the few research works that tackled the same and similar problems and the results obtained by their experiments. This section is further divided into subsections that detail studies based on the category of their followed approach.

3.1 Traditional Market Price Prediction

Forecasting stock prices have been a significant focus for various financial and business studies to maximize profit and minimize the losses of investors and businesses. The ability to predict the future prices and trends of a market of interest has a massive impact on making the right decision at the right moment. Researchers have constantly been proposing approaches to solve that particular issue, one of the first approaches applied has adopted a statistical model [14] because it was the most reliable tool achieved at that time. Research works continued chiefly on applying statistical or probabilistic approaches to tackle similar price prediction or forecasting problems. When machine learning had success in multiple other regression and prediction problems, it has also been introduced to the market price predictions. Havaluddin et al. [4] made an in-depth comparison of statistical and machine learning techniques in learning time-series data and making near future predictions. The comparison mainly covers the statistical method ARIMA, Backpropagation neural networks, and genetic algorithms. Results show that the backpropagation neural network is more efficient and reliable in forecasting short-term time series.

Despite traditional machine learning showing a potential in working with time-series data, researchers are constantly breaking ground when it comes to proposing more efficient works, which is reflected by the introduction of more sophisticated deep learning networks. In the recent decade, with all the improvements that deep learning had with the efficiency of its applications, it became a primary focus for financial and economic forecasting and time-series predictions [15, 37, 19, 34]. Deep learning approaches were able to significantly outperform the other methods due to their ability to learn hidden features within time-series and historical market trends. Most traditional markets have fixed factors and rules that directly influence the price

trends, including the number of asks, bids, the number of transactions, and much more. This simplicity makes it even easier for a deep learning model to learn the patterns from historical data and have a much more reliable prediction that can be helpful for decision-making, this is reflected through the high accuracy and low error of the forecasting by just applying prices, asks, and bids time series [34, 37].

In recent years, cryptocurrency has become a massive investment opportunity for many people, not only investors. This gem opportunity became apparent in 2013 when Bitcoin had its first unexpected price growth, such that the prices moved from \$138.13 in October 2013 to \$1131.97 in November. The unpredictable increase shined a light on this particular cryptocurrency, and investors jumped in to invest in this market and try to gain as much profit as possible. Unfortunately, like any new market, there was little to no information on this market and how the fluctuations are affected. Studies have tried applying the same state-of-the-art deep learning approaches used in traditional markets to forecast Bitcoin prices[27]. However, price trend predictions obtained had low accuracy, such that they cannot be relied upon to make such decisions. One reason for this low result is the assumption that the cryptocurrency market is similar in characteristics to traditional markets.

3.2 Machine Learning for Cryptocurrency Price Prediction

Discovering the actual fundamental differences and overcoming the issue of short-term price predictions in the cryptocurrency market was a constant primary focus for many research works. Yiyang and Yaze [45] proposed a novel approach that learns the hidden features within time series, focusing on the price non-stationary dynamics of three cryptocurrencies (Bitcoin, Ethereum, Ripple). The study conducted multiple experimental analyses using various LSTM architectures and classical Artificial Neural Networks (ANN) that take the price, ask, and bids time series as input and outputs several predictions that cover both the long and short-term price predictions. Results concluded that applying the correct ANN architecture can learn long-term patterns and make better long-term predictions. On the other hand, LSTM tends to rely more on the short-term dynamics of time series and perform better short-term predictions. These results indicate the efficiency of LSTM architectures in learning valuable information hidden in the historical memory better than ANNs, and this is reflected in the experiments that show LSTMs outperforming ANNs in short-term predictions.

Furthermore, understanding LSTMs and their performance on cryptocurrencies is essential to understanding the advantages and limitations of LSTMs. For this reason, McNally, et al. [29] conducted comparison experimentation covering two deep learning models for Bitcoin price predictions, a Bayesian-optimized Recurrent Neural Network and an LSTM network. The study applies trading data (prices, asks,

bids, and the number of transactions) on both networks and makes a binary price trend classification (price goes up or down). The predictions were made for both long and short-term predictions, and results demonstrated that LSTMs achieved a higher accuracy of 52% that marginally outperforms the Bayesian-optimized Recurrent Neural Network and the classical statistical method ARIMA. However, one particular feature noticed within the cryptocurrency time series is the high volatility and variance over time, making it difficult for the models to have impressive validation results. As a result, this problem remains a complex and challenging task. Researchers have concluded that there is a very fine line between overfitting and underfitting in this task [29]. During the experimentation phase, even by using the Bayesian optimization to optimize the selection of dropouts it still could not guarantee the learning of the model and the achievement of good validation results.

Similarly, Kumar and Rath [23] focused on forecasting the trends of Ethereum prices using machine learning and deep learning methodologies by applying only trading data. The authors proposed an LSTM architecture designed for this specific task. The evaluation shows that LSTM marginally outperforms the Multi-Layer perceptron (MLP) in short-term predictions but not considerably. That suggests that even though LSTM proves to be a better approach for price predictions in traditional markets and generally works well with time-series forecasting, it does not perform to the same extent when it comes to dealing with cryptocurrency price predictions. Further comprehensive experiments were conducted by Pintelas et al. [35] to understand this phenomenon further. Results show that despite LSTM-based and Convolutional neural network (CNN)-based models being preferable for time-series forecasting, they could not generate efficient and reliable results and forecasting models.

Additionally, a study on the nature of cryptocurrency prices conducted by the same paper concluded that they follow an almost random walk process. At the same time, a few hidden patterns may probably exist, where a model or intelligent framework has to identify them to make more accurate forecasts, but this remains a hypothesis as the experiments were not designed to tackle and answer it. Finally, they have suggested that new alternative approaches and new validation metrics should be explored to solve this task and issue.

Recently, Chevallier et al. [9] proposed a novel approach that improved performance significantly from previous works while also keeping the simplicity of the architecture. They proposed an AdaBoost approach that uses multiple decision tree weak learners to tackle the issue of cryptocurrency forecasting. Surprisingly, the results have demonstrated that AdaBoost outperforms all ANNs, LSTMs, KNN, and SVMs by an average RMSE of \$23.42 per 1 Bitcoin in their testing set. Furthermore, using simple models to perform good results is excellent for generalization over time and allows interpretability of outputs. Authors have suggested that providing

a similar simple architecture that can better learn hidden features will increase the performance of the model when appropriately applied.

3.3 Sentiment Analysis for Cryptocurrency Price Prediction

The high volatility of the cryptocurrency market has introduced a massive challenge in allowing a model to learn hidden features. One of the first in-depth financial studies that shine a light on cryptocurrencies and the factors influencing them was the study made by Krisoufek [22]. In this market study, multiple analyses were made to understand the price fluctuations to determine the driving factors of the prices for multiple cryptocurrencies. Results concluded that multiple factors directly affect the prices, summarized into three categories: the correlation between the number of asks, bids and the other markets, The hash rate and other blockchain information, and the public awareness. According to the study, all these factors contribute to the same extent to the price fluctuations. However, public awareness has proven a much stronger correlation.

Because of the natural complexity of measuring public awareness, several metrics can encode this factor, such as the sentiments. People's sentiments can be both digital (social media sentiments and news blogs) and non-digital (word of mouth and offline ads)[22]. Inspired by this study and from observing bitcoin price trends and social media, Young, et al. [20] has first introduced the hypothesis that cryptocurrency forums' sentiments influence Bitcoin prices. In an effort to validate the hypothesis, the proposed approach considers only the sentiment data as input to the model. Using the *VADER* sentiment analysis tool, the study analyzes user comments from the three most popular cryptocurrency forums, tagging each comment with a sentiment score from 0 being very negative and 1 being very positive. The experiment results have shown that there is indeed an existing correlation between the price trends and sentiments on the most popular cryptocurrency forums, suggesting that fluctuations within the forum's sentiments are an early indication of near-future price fluctuations.

In 2017, Bitcoin experienced the first exponential price growth, where it received a high news coverage with very positive sentiments in that particular period. People, news, and especially investors were hyped towards this new investment opportunity. The positive sentiments at this interval have partly impacted the prices positively. Recently in 2021, when news, social media, and governments were promoting against cryptocurrencies due to the hashing consumption of electricity and the harm that is caused to the environment [36], Bitcoin prices experienced a 41% drop in just 15 days, going from \$58488.21 to \$34259.55. The negative sentiment generated for such announcements resulted in much fear within the market and, therefore, a further collapse of the prices. These events suggest that sentiments partially

influence the cryptocurrency price trend, especially sudden short-term changes. Recent studies [33, 2, 16] utilized sentiment analysis approaches along with an LSTM model to predict its prices on the next trading day. First, they have crawled all social media and forum posts that contain keywords related to cryptocurrencies (Bitcoin, Euthereum, and Ripple), then each tweet got assigned a score from 0 to 1 depending on its sentiment using the VADER tool (zero being very negative, one being very positive). Next, all the sentiment scores are combined with the prices, asks, and bids time series into a single vector which serves as input to the proposed LSTM architecture. Experiments of these approaches demonstrated that applying sentiment analysis marginally improves the prediction results over other previous deep learning models that rely purely on trading data.

To better understand the extent of these improvements, Huang, et al. [16] made a comparative analysis between their approach that embeds social media sentiments and autoregression using only trading data. The main problem tackled by this study is the binary classification of the future trend of cryptocurrency prices, by either the price going up or down. Results show that embedding sentiments into the input increase the accuracy by 18.5% and the recall by 18.5% from the autoregression model. However, even though such improvements are non-negligible, they are not enough to achieve reliable results that can be used and helpful for financial decision-making, as concluded by the study.

Previously mentioned studies [33, 2, 16, 9, 35, 23] all share a common challenge and limitation within the formulation of the approach and the selection of the data. First, despite the association analysis used by these studies to filter the social media posts and user comments, a more qualitative tweets selection criteria are needed to build a prediction model. Because the level of interactions each post has (likes, comments, and shares) can significantly influence the impact a particular post had on the overall sentiments. Therefore, considering this is vital to encode the sentiments of social media posts better because not all posts have an equal level of the potential impact on the cryptocurrency market's price. The more interactions a post has, the more significant its influence on the price is. For example, famous news channels or people's posts may have high interactions and therefore impact and influence on the price than other regular posts. Furthermore, another limitation with previous studies is the focus on only online communities and social media posts to determine and predict the price fluctuations while ignoring other factors that are proven to be as crucial as sentiments to the fluctuation changes [22]. It is hypothesized that taking into consideration the correlation between prices, asks, and bids, blockchain information, and public awareness will have a significant improvement on the prediction accuracy. In addition, past studies have shown that analyzing social network data and inferring to search volumes on google are conducive to more precise results[22].

4 Approach

DMCrypt is an ensemble learning approach that adopts adaptive boosting and LSTM architectures due to their capability to learn hidden features and trends within time-series data as discussed in section 3. The overall architecture consists of multiple LSTM weak learners that together combine a single cryptocurrency price prediction. The first step of this process is the training phase, where each LSTM model is trained on a sampled dataset from the original one, and then each model gets assigned a weighted score according to its performance to be used in the inference phase. Finally, when making predictions, all LSTM outputs are multiplied with their weights and divided by the total number of LSTM models to calculate the final price prediction. The subsequent subsections further describe all the details of the DMCrypt approach.

4.1 Datasets

This subsection will cover all the necessary details for collecting and preprocessing the used to train and evaluate DMCrypt. Because data is a crucial component for any deep learning approach to learn and perform at its best, it is necessary to ensure that all the data is preprocessed correctly and effectively.

Due to some limitations encountered in the data collections, Bitcoin is the cryptocurrency use case considered in this thesis. The data used covers multiple multimodalities ranging from time-series to texts, therefore, this subsection is divided into the following:

4.1.1 Trading data:

Trading data is a set of time series consisting of different attributes of the Bitcoin cryptocurrency, each of which represents a particular feature. The data is collected using the following source:

- **Kaggle:** an open-source dataset containing a 1-minute interval data of the Bitcoin prices and several other attributes summing up to 8 attributes in total. The data collected started from the 1st of January 2012 until the 1st of March 2021.
- **Binance:** is a digital platform that allows buying, selling, and trading cryptocurrencies of all kinds. The data provided is public. However, there are no specific APIs to collect the data in a suitable format for this thesis. As a result, the data was crawled from the website using a custom python script to translate the data from a visual format to a suitable csv format.
- **Coinbase:** is an online platform that provides selling, buying, and trading services for investors. In contrast to Binance, Coinbase provides APIs to collect

the Bitcoin prices and other attributes in a suitable format. In addition, Google uses its API to provide answers for searches related to Bitcoin data (prices and transactions).

The rate differences are an important point to consider while collecting trading data from different sources, especially prices time series. For example, The bitcoin price for the 1st of December 2021 is \$56,950 in Binance and \$50,516.62 in Coinbase. Binance charges a small percentage fee for Bitcoin prices. Therefore, handling such fees is vital to ensure consistency within the data. In the case of Kaggle, all the data available is similar to Coinbase.

The data provided by these sources contain some missing values for some data, and some days are skipped and not included within the data. Handling this issue was done by using the three different sources together. For example, if there is a missing value for some date in Kaggle, the other sources replace that missing date. So far, there has been no instance where the same value for a particular day is missing in all three data sources.

The overall data for all sources consists of the following attributes:

- **Timestamp:** is the starting time of the 1-minute interval for the specific data point in a UNIX format.
- **Open:** is the price of one Bitcoin at the start of the 1-minute time window for the specific data point in US Dollars (USD).
- **High:** is the highest price of one Bitcoin within the 1-minute interval of the specific data point in US Dollars (USD).
- **Low:** is the lowest price of one Bitcoin within the 1-minute interval of the specific data point in US Dollars (USD).
- **Close:** is the price of one Bitcoin at the end of the 1-minute time window for the specific data point in US Dollars (USD).
- **Volume_(BTC):** is the volume of Bitcoins transacted in the 1-minute time window.
- **Volume_(Currency):** is the volume of Bitcoins in US Dollars transacted in the 1-minute time window.
- **Weighted_Price:** stands for the Volume Weighted Average Price, which is the ratio of the value of Bitcoin traded during the 1-minute time window.
- **Average_fees:** is the average fees charged for transactions in the top 20 trading and exchange platforms and services.

- **Transactions:** is the total number of transactions made within a 1 minute time window.

4.1.2 Social Media Data

As proven by many financial studies and research works, the cryptocurrency market is highly affected by public sentiments, given that the market lacks any physical assets. Therefore, embedding the sentiments is crucial for this approach to have reliable price predictions. Public awareness and sentiments are complex topics by themselves. The sentiments and opinions can be gathered from an endless number of sources, either online (Social media, blogs, news articles, and visual content) or offline (word of mouth and television). Some of the previously mentioned sources are very hard to collect, if not impossible, like word of mouth. For this reason, choosing the appropriate source to collect the data from is needed.

According to the official statistics collected from all social media platforms, 4.48 billion people are actively using social media daily, roughly 57.7% of the world's population⁸. However, a common problem of collecting data is the actual availability of the data, resulting in limitations on what data can be legally collected. Twitter is one of the few social media platforms that facilitate data collection for research purposes without charging fees; besides, Twitter is one of the top 5 social media platforms in 2021 according to the number of active users with almost 202 million daily users and over 500 million tweets a day⁹. Therefore, Twitter is considered as a source to analyze the public's sentiments based on the collected tweets.

Because of the large volume of tweets related to Bitcoin, the tweets collection is made using two methods described below:

- **Twitter Academic Research API:** it is an API provided by Twitter for research purposes only. Applying to the access takes place on the official Twitter developers page, submitting all the necessary details regarding the research and data usage is mandatory to obtain the access. It allows a swift collection of historical tweets using queries up to 1024 characters and a limit of 10 million tweets a month. In addition, the API allows collecting the following information for every tweet: ["*The tweet*", "*Creation date*", "*User ID*", "*Tweet ID*", "*Public metrics: likes, retweets, comments, quotes*"].
- **TweetScraper:** Due to the limitations imposed by the Twitter Academic Research API, a second source was needed to collect all the data necessary. *TweetScraper* is an open-source script developed and published on GitHub to collect tweets legally. Although all the benefits that this crawler provides, it has two main

⁸<https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/>

⁹<https://www.omnicoreagency.com/twitter-statistics/>

limitations: the collection time and the retrieved data. The time needed to collect 1 million tweets is 23 times larger than the Twitter Academic Research API. Furthermore, the data is retrieved in a raw format meaning that data cleaning is mandatory before storing the data. The following are few of the attributes retrieved by the crawler: [*"The tweet and the creation date", "The user ID", "The user account description and creation date", "The user followers count", "The tweet ID", "The number of likes, retweets, comments, and quotes"*].

All the tweets are collected using the query: *"q = 'Bitcoin' or 'BTC' or #Bitcoin or #BTC"*. The query resulted in more than 120 million results. As per any social media data, the tweets contain a good portion of spam tweets (Tweets that use hashtags to gain a more extensive audience, for example, the hashtag *#bitcoin* or *#btc*, while the actual tweet refers to other subjects), therefore, any tweet that contains the word Bitcoin as hashtag and not as text is then removed. All the emojis used are kept in their Unicode format to be used later with a finetuned model for sentiment analysis. Next, all the tweets collected from the Twitter Academic Research API are stored in a MongoDB database awaiting the next preprocessing phase. The database choice of MongoDB is made to allow fast retrieval of semistructured data in a JSON format.

All the raw data collected by the TweetScraper crawler takes a raw and unorganized format. The cleaning of this data requires extracting only the following information: *Created_at, Full_text, Tweet ID (If the Twitter academic research API collected the same tweet, we could remove the duplicates), Retweet_count, Favorite_count, Reply_count, and Quote_count*.

Next, after the data cleaning process, there are three essential data preprocessing phases: sentiment analysis, assigning weighting scores, aggregating tweets sentiments into a day-interval time-series.

4.1.3 Sentiment analysis

The ultimate purpose of collecting tweets is to analyze sentiments and embed them into our input to encode the public awareness factor. Sentiment analysis is a separate field by itself, where researchers are constantly working on making improvements and increasing the accuracy of the sentiment classifications from texts, sounds, and even videos. This thesis aims to use two popular pretrained and publicly available sentiment analysis tools to analyze tweets' sentiments. Below is a clear description of the approaches used for this study:

Vader:

Vader[17] is a rule-based model for social media sentiment analysis. According to the evaluation conducted by the research, Vader outperforms multiple state-of-practice approaches that use rule-based or machine learning. In addition, according to the research, Vader generalizes better across contexts than other approaches.

The outputs of the Vader model consists of the following:

Text	VADER sentiments	Deeply Moving sentiments
👤 I ❤️ decentralized finance on #Bitcoin - @defichain 📈 At this moment I have 100%+ APY on my staked \$DFI on @cakedefi 🍌 🍌 🍌 visit: https://t.co/fZLzb4alug	0.402	-0.321
RT @fastbitcoins: "The gov will ban Bitcoin!" 🙄 Even with unlimited fiat funding, the current system is riddled with misaligned incentives	-0.380	-0.843
RIP #bitcoin 🙄 Ethereum up 16.55% - and bitcoin died 91 times this year. Read the full article: https://t.co/NQvAfy4Xky \$BTC	-0.557	-0.642
Understanding #Bitcoin in #Kenya during the #FestiveSeason #Christmas #Business #Economy #Trade #Cryptocurrency #Payment #Currency #Technology #Internet #Banking #Africa https://t.co/FpT6PaEwGL	0.7034	0.0

Figure 4: Sentiment analysis on tweets using VADER and Deeply Moving

- *Sentiment score*: a value that ranges from $[-1, 1]$ that reflects the sentiment of a given tweet. The higher the value is, the more positive the sentiment is. -1 stand for highly negative, 1 stand for extremely positive, and 0 stands for neutral.
- *Classification*: classify a given tweet into nine classes (extremely negative, very negative, moderately negative, slightly negative, neutral, slightly positive, moderately positive, very positive, extremely positive). The classification is made based on the sentiment score, such that each given interval in the sentiment score represents a specific class ($[-1, -0.75]$: extremely negative).

Figure 4 illustrates a few examples where sentiment analyses have been performed. A common issue that arises from using sentiment analysis tools is having false positives and false negatives. Unfortunately, evaluating sentiment scores is not possible because of the large volume of tweets collected and the sentiment pre-labels' absence. As a result, applying another approach is recommended to minimize the effect of falsely classified tweets.

Deeply Moving: Deep Learning for Sentiment Analysis:

Deep learning has been used in various fields during the past years, and sentiment analysis is one of them. Deeply Moving[39] is an open-source deep learning approach proposed by Stanford to analyze complex sentiments within texts. Generally, in traditional approaches, an analysis of the isolated words is made, averaging the sentiment score at the end, resulting in a loss of information within the text. On the other hand, Deeply Moving analyzes the text as a whole entity to preserve all

the information. Mainly, this approach is proposed to analyze movies' reviews sentiments, meaning that the text is formally represented and lacks the same structure used in Twitter's tweets (Use of emojis, hashtags, and abbreviations). In addition, finetuning the model on tweets sentiments for this study is challenging due to the lack of labeled data. Therefore, this thesis aims to use both models to get the best of both worlds (a model that is explicitly trained on tweets and a model that can understand the sentiments of a whole sentence).

The Deeply Moving model is publicly available in GitHub for use. Similar to Vader, the outputs are the following:

- *Sentiment Score*: a value in the range of [-1, 1] denoting the actual sentiment of the given text. The lower the value, the negative the sentiment is.
- *Classification*: a probability value for the given nine classes (extremely negative, very negative, moderately negative, slightly negative, neutral, slightly positive, moderately positive, very positive, extremely positive), all the probabilities for all the classes should sum up to 1. The text belongs to the class with the highest probability.

Figure 4 illustrates a few tweets where Deeply Moving has been applied. As can be seen from the figure, the model performs at its best when the tweet's text is formally written (without any emojis, abbreviations, and hashtags). However, when a tweet contains emojis or any informal structure, the model struggles to determine the correct sentiment. It is one of the main limitations of this modern sentiment analysis tool.

Both models (Vader, Deeply Moving) are applied to all the tweets that were collected before, and the following new columns are added to each document within the MongoDB database:

- *Vader_sentiment*: contains the sentiment score given by Vader.
- *DeeplyMoving_sentiment*: contains the sentiment score provided by Deeply Moving.
- *Average_sentiment*: is an average sentiment of both models. For each tweet the average is calculated as follows: $average_sentiment = \frac{s_v + s_d}{2}$, where s_v denotes the vader's sentiment, s_d denotes the Deeply Moving's sentiment, and $average_sentiment$ is the final average sentiment that is stored in the new column "Average_sentiment".

4.1.4 Weight scores

According to a recent statistic about Twitter Engagements, the median number of likes and comments is 0, meaning that a large number of tweets receive little to no engagement at all. Therefore, an equal sentiment weighting for all the tweets will

not reflect the actual scenario. Because public sentiments are one factor that affects cryptocurrency prices, adding a weight to each tweet that reflects the engagement rate is essential. A tweet's effect on people's overall sentiment depends on multiple factors, including the tweet's reach and the number of interactions, for example suppose the following tweets:

- The tweet $t_1 = \text{"Bitcoin is the future"}$ written by the person p_1 with 2 million followers, the tweet had 112,200 likes and 50,000 retweets.
- The tweet $t_2 = \text{"Never invest your money in Bitcoin. It is terrible."}$ written by the person p_2 with 100 followers, the tweet had two likes and 0 retweets.

The tweet t_1 is likely to have a more significant effect on people's sentiments than tweet t_2 , because of the larger reach. The weight score of each tweet is calculated in two steps:

1. The first step: calculate the average of $t_{likes} + t_{comments} + t_{retweets} + t_{quotes}$ and assign it to w_t where it denotes the initial weight score for the tweet t , t_{likes} , $t_{comments}$, $t_{retweets}$, t_{quotes} denotes the tweet's total number of likes, comments, retweets, and quotes respectively.

Repeat this until all tweets have an initialized weighted score w_t .

2. The second step: normalizing all the scores such that all weights are between $[0, 1]$ using the min-max normalization and assign it to w_{t-norm}

The second step: normalizing all the scores using the min-max normalization such that all weights are between $[0, 1]$ and assign it to w_{t-norm} , where it denotes the normalized weighted score.

After the second step is finished, we store all the w_{t-norm} for all tweets under a new column in our database to facilitate future retrieval.

The final sentiment score for every tweet is calculated as follows:

$$s_t - f = s_t * w_{t-norm}, \quad (11)$$

where $s_t - f$ denotes the final sentiment score for the tweet t , s_t denotes the average sentiment score of the tweet t and w_{t-norm} denotes the normalized weighted score of the tweet t .

4.1.5. Sentiments Aggregation

So far, all the weighted sentiments are calculated for every tweet and stored in the

database as a new column. However, to make the sentiments suitable for the DM-Crypt model's input, an aggregation of the sentiments is needed to form a daily-interval time series. Therefore, all weighted sentiments of the published tweets for the date d are averaged to form a single value that represents the overall Twitter's public sentiments on the date d . The final output of this phase will be a time series with each date representing the average sentiment of the tweets published at the same date. For each date d , the average weighted sentiment score is calculated as follows:

$$s_d = \frac{\sum_{i=0}^n s_{i-f}}{n}, \quad (12)$$

where, s_d denotes the average Twitter's sentiment at the date d , n denotes the total number of tweets published on the date d , and s_{i-f} denotes the weighted sentiment of the tweet i .

Moreover, the total number of tweets per day is also stored in a separate dataset. It is essential to record the tweet volumes to capture the public's sentiments within a day. For instance, if a particular day has more tweets than usual, the public is conveying certain information, and therefore a specific sentiment.

Finally, all the final averaged weighted sentiment scores are stored in a suitable CSV format as time series type to be used as input for the DM-Crypt model.

5.3.6 Blockchain Data

Blockchain has a critical role in maintaining decentralized and secure records of transactions for all cryptocurrencies. Furthermore, according to the financial study conducted by Krisoufek [22], Blockchain plays a huge role in affecting the price fluctuations of cryptocurrencies, especially Bitcoin. As a result, this thesis embeds multiple blockchain information that multiple financial studies have proved to affect the market [22]. There are multiple sources used to crawl and collect all blockchain data from, and the following are the main ones:

- Blockchain.com.
- Ycharts.
- Bitinfocharts.
- Nasdaq Data Link.

Ultimately, these are the following attributes used to represent all the blockchain data:

- *Hash rate*: the estimated hash rate the Bitcoin network is performing in a given day.

- *Block size*: the size of a complete bitcoin block in a given day.
- *Block time*: the time required to mine and produce a new bitcoin block in a given day.
- *Network difficulty*: the difficulty level of mining Bitcoin blocks through the network in a given day.
- *Active Addresses*: the total number of active addresses in a given day.
- *Mining profitability*: The estimated average mining profitability for a single bitcoin block in a given day.

All the beforementioned blockchain attributes are of type time series data with a one-day interval between each data point as illustrated in Figure N.

5.3.7 Search volumes:

The last type of data used in this thesis is the search volumes. Online searches often reflect the genuine opinion of a public or group of people towards a particular topic. People's curiosity is generally reflected through various actions, and these include getting familiar with a topic or hearing other people's take from the topic or situation. Nowadays, it is easier than ever to search online, thus, making it easier to know and understand public opinion through search statistics.

Google is one of the largest search engines in the world, if not the biggest, with over 52% of the world's population using it every day¹⁰. Therefore, it is one of the most excellent sources for collecting search statistics because the results will cover a large portion of the total searches made online. Likely, Google provides an open-source API that can collect all the necessary data for this thesis. The search volumes were collected using the following query:

- Any search that contains either the word "Bitcoin" or the word "BTC" is collected. Notice that the query is not case sensitive, meaning that Bitcoin can be found either upper or lowercase.

Figure 6 illustrates the search volumes of the above query "Bitcoin" from the year 2013 until the year 2021.

4.2 LSTM Proposed Approach

When working on the problem of price prediction of cryptocurrencies in the next 24th hour, it is vital to choose a model that can handle short-term predictions. According to a research work conducted by Pintelas et al [35], the evaluation results

¹⁰<https://review42.com/resources/google-statistics-and-facts/>

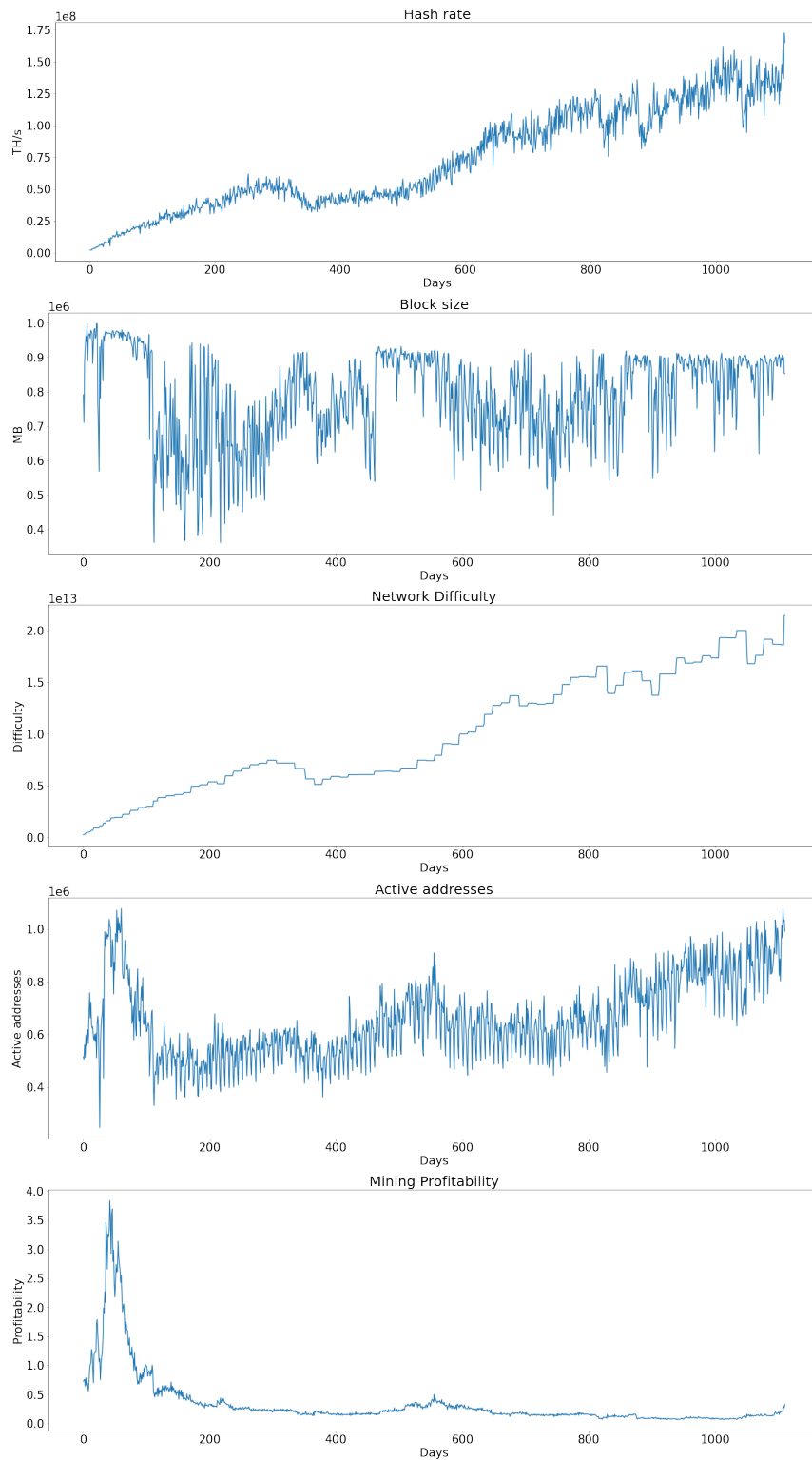


Figure 5: Blockchain data plots

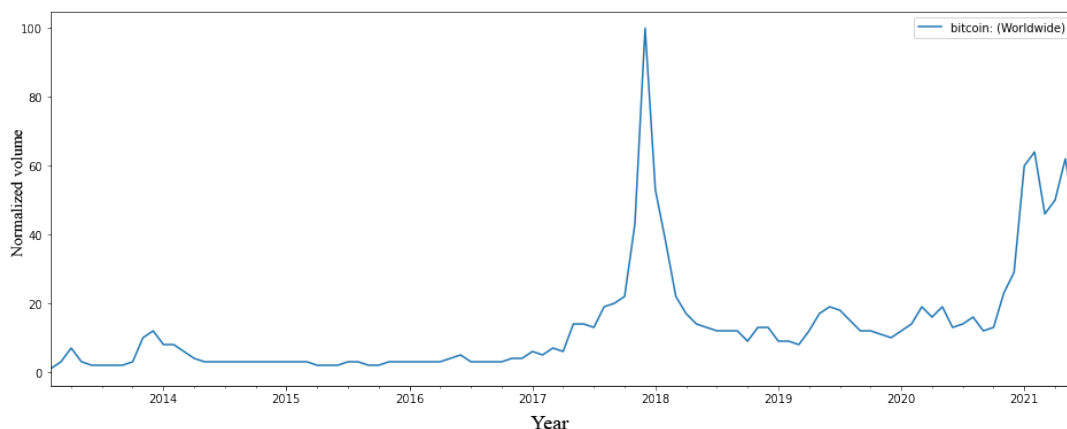


Figure 6: Google search volumes for the word Bitcoin

concluded that LSTMs have a more excellent capability to learn hidden features from data of type time-series than other models like Artificial Neural Networks (ANNs). Moreover, LSTMs are especially good when dealing with short-term goals. Therefore, this thesis aims to propose an LSTM architecture to benefit from all the advantages it provides, given that the data after preprocessing is of type time-series.

Multiple LSTM architectures have been applied and evaluated during the experimentation phase to test the optimal architecture for this thesis’s use case. Figure 7 briefly describes the final optimal proposed LSTM architecture used in this thesis, where the input of the LSTM model consists of the normalized feature vectors of the last seven days ($[X_{t-7}, X_{t-6}, \dots, X_t]$), where X_t denotes the normalized feature vector of length 18 for the date t , and the expected output for the model is a price prediction of 1 Bitcoin for the next day ($t + 1$) in US dollars.

Figure 7 briefly demonstrates the LSTM architecture visually with all its layers.

4.3 Adaptive Boosting and LSTM Ensemble Learning

A recent research paper published by Chevallier et al. [9] that tackled the problem of cryptocurrency price predictions found that applying Adaptive Boosting (AdaBoost) with multiple decision trees improves prediction results and outperforms other state-of-the-art approaches. Many works use Adaboost to boost the performance and results of almost any machine learning model. It combines the performance of multiple weak learners to obtain better results. The limitation that arises from applying Adaboost with decision trees for regression is that the cryptocurrency data used to make predictions is far complex for a decision tree to perform at its best—as a result, preserving advantages that are provided by LSTM, like learning hidden features and patterns.

Likely, a paper published by Sun et al. [41] developed a hybrid ensemble learning architecture that employs LSTM as weak learners for the AdaBoost algorithm. The work is designed to tackle the problem of traditional financial time series forecasting. This thesis aims to propose a similar approach along with the previously mentioned LSTM architecture to make cryptocurrency price predictions. The adopted LSTM-AdaBoost algorithm proposed by Sun S. et al. [41] composes of 6 main steps described as follows with just a few adjustments to suit this thesis's problem:

1. Initialize the sampling weights D_i for all the training samples. The sample weights are calculated as follows:

$$D_i = \frac{1}{N}, i \in \{1, 2, \dots, N\} \quad (13)$$

where N is the number of training samples.

2. The first LSTM predictor $M_i, i = 0$ is trained on the training samples sampled using the sampling weights $D_i, i \in 1, 2, \dots, N$
3. Calculate the error e_i^t of the LSTM predictor M_i as follows:

$$error_i = \frac{|y_i - \hat{y}_i|}{y_i}, \quad (14)$$

where y_i denotes the actual price of the data point x_i , and \hat{y}_i denotes the predicted price for the data point x_i .

4. Calculate the predictor's weight W_m as follows:

$$W_m = \frac{1}{2} \ln\left(\frac{1 - \sum_{i=0}^N error_i}{\sum_{i=0}^N error_i}\right) \quad (15)$$

5. Next, we update the sampling weights D_i of all samples as follows:

$$D_{i_new} = \frac{D_i e^{error_i}}{\sum_{t=0}^N D_t e^{error_t}}, \quad (16)$$

where e^{error_i} denotes the update rate of the sample x_i .

6. Next, repeat the steps of training the LSTM predictor, calculating the error, and updating the weights for all LSTM predictors used.

Finally, after training all the LSTM predictors, we can compute the final price prediction by combining the LSTMs' outputs with their weights to obtain a final prediction as follows:

$$\hat{y}_t^f = \frac{\sum_{m=0}^M \hat{y}_t^m W_m}{M}, \quad (17)$$

where M denotes the total number of LSTM models (predictors), \hat{y}_t^m denotes the price prediction of the LSTM model m , and W_m denotes the LSTM model m 's weight.

Figure 8 visually demonstrates the Adaptive Boosting-LSTM ensemble learning model's architecture.

4.4 Model Varieties and Dropouts

One of the main contributions of this thesis is to provide an output that is helpful for an investor's decision-making process. Unfortunately, a single price prediction does not suffice for making such a decision because the cryptocurrency market's volatility is not uniform over time. For instance, if tomorrow's price fluctuation level is low (approximately 20\$ in standard deviation), the model's price prediction will be extremely close, if not the same as the actual value. However, if the fluctuation level is very high (approximately 2000\$), in this case, the price prediction can be off from the actual value due to the high fluctuations. As a result, performing multiple varieties of the model to make multiple predictions is essential to serve as a basis for creating and estimating the predicted price distribution.

For this particular task, the goal is to have multiple predictions coming from multiple model varieties that are created using two methods described in detail below:

1. **Input varieties:** One way of having different price predictions is by applying the same architecture over multiple varieties of data. The original model is applied to a total of 18 features mainly consisting of the following categories: *Trading data, Twitter sentiments, Blockchain data, and Online search volumes*. All the data attributes description is detailed in section 5

A common way of obtaining multiple price predictions is by applying combinations of these categories to the same model architecture with just a few adjustments to the number of input nodes. Therefore, the following models' varieties are applied:

- **Trading data:** The main component of any market, including cryptocurrencies, is the actual trading data. As a result, applying this data is considered the minimal data that can be applied in cryptocurrency price prediction. The trading data consists of the following 8 features: [*"Open", "High", "Low", "Close", "Volume BTC", "Volume Currency", "Weighted Price", "Average Fees"*].

The model applied for this use case is similar to the approach mentioned before except for some changes in the input layer, where the LSTM model takes eight features as input instead of 18.

- **Twitter sentiments:** As seen previously, multiple research works proposed that social media sentiments and public awareness play a huge role in affecting the price fluctuation levels. A research work done by Abraham, et al. [2] proposed a Bitcoin price prediction applying only Twitter sentiments without any external data. It aims to experiment and test if there is a correlation between the fluctuations of sentiments and prices. Results show that there is indeed a correlation between how social media sentiments change and the price changes. As a result, it is expected for this thesis's approach to have decent predictions while applying only the twitter sentiments. The data

consists of the following two features: [*"The weighted twitter sentiments"*, *"Tweet volumes"*].

This model is similar to the first variant except that the input layer consists of only two input nodes.

- **Trading data with blockchain data:** Blockchain data greatly influences cryptocurrency prices, especially the hash rate. It is their backbone, after all. This model's variant is similar to the previous ones except that the input layer consists of 14 nodes denoting the following: [*"Hash Rate"*, *"Block Size"*, *"Block Time"*, *"Network Difficulty"*, *"Number of Active Addresses"*, *"Mining Profitability"*, *"Open"*, *"High"*, *"Low"*, *"Close"*, *"Volume BTC"*, *"Volume Currency"*, *"Weighted Price"*, *"Average Fees"*]
- **Trading data with search volumes:** Because the online search volumes consist of only a single feature, it is nearly impossible to make an accurate price prediction for a highly volatile market. Therefore, it is advised to apply online search volumes along with the trading data. This model variant is no different from the previous one, except having nine input nodes representing the following: [*"Online Search Volumes (Google searches)"*, *"Open"*, *"High"*, *"Low"*, *"Close"*, *"Volume BTC"*, *"Volume Currency"*, *"Weighted Price"*, *"Average Fees"*]

2. **Model dropouts:** The second method applied to get variant outputs is by applying different dropouts to different layers of the final trained model. A dropout, in simpler terms, is a technique used mostly during training that randomly selects and disables nodes (neurons) on a particular layer. By disabling random neurons each iteration, we are allowing all the paths within a particular network to be trained. It is a common technique used to avoid overfitting. Moreover, it can be used in other phases besides the training phase to obtain multiple variant outputs from the final model to create a sample of outputs. This approach aims at estimating an output distribution based on the given dropout samples. In this thesis, dropouts are the second method used to obtain multiple outputs. The model's architecture proposed can take dropouts in two layers, first, the LSTM's output, and second, the fc1 outputs. For that reason, the following are the suggested dropout varieties:

- *Model V_1:* A dropout of probability 0.2 for the LSTM's last hidden layer.
- *Model V_2:* A dropout of probability 0.2 for the fc1's output.
- *Model V_3:* A dropout of probability 0.1 for the LSTM's last hidden layer.
- *Model V_4:* A dropout of probability 0.1 for the fc1's output.
- *Model V_5:* A dropout of probability 0.35 for the LSTM's last hidden layer.
- *Model V_6:* A dropout of probability 0.35 for the fc1's output.

Finally, after applying the model with all these varieties, we end up with six price samples that will be used to derive the predicted price distribution in the next phase.

4.5 Predicted Price Distribution:

Making investment decisions is not an easy process that can rely on a single price prediction. Instead, it is a thorough process that requires a lot more knowledge on the field as well as many prediction inputs to be sure that a specific decision is the best possible move. Distributions are a great tool that a decision-maker can use to accumulate information about the best possible actions to take. They provide not only a price prediction but the uncertainty levels of the prediction. As a result, distributions can be a very reliable tool to be used in making such difficult decisions. The proposed distribution takes the ten outputs obtained from the model varieties and performs a maximum likelihood estimation on the samples to obtain the parameters of the predicted prices distribution. The parameters that need to be estimated in the distribution are the mean and the standard deviation, which are easy to estimate using the maximum likelihood estimation. Let the ten model variant's outputs be a sequence $X = o_1, o_2, \dots, o_{10}$ of length 10, where $o_i, i = 1, 2, \dots, 10$ is the output of the model variant i (the order is the same same as presented by the previous subsection). The sequence X has a mean μ_0 and a variance σ_0^2 . The aim is to have a probability density function of the predicted prices; this function is defined as follows:

$$f_X(x_i; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2}, \quad (18)$$

where it is required to estimate the mean μ and variant σ^2 for the function.

Firstly, it is crucial to derive the likelihood function to be able to proceed with the following steps, the function is derived as follows:

$$\begin{aligned} L(\mu, \sigma^2; x_1, x_2, \dots, x_{10}) &= \prod_{i=1}^{10} f_X(x_j; \mu, \sigma) \\ &= \prod_{i=1}^{10} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2} \\ &= (2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{10} (x_i-\mu)^2}, \end{aligned} \quad (19)$$

Secondly, applying the log to replace the product with a sum greatly facilitates the process. The log-likelihood is defined as follows:

$$\begin{aligned}
l(\mu, \sigma; x_1, x_2, \dots, x_{10}) &= \log(L(\mu, \sigma; x_1, x_2, \dots, x_{10})) \\
&= \log((2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{10} (x_i - \mu)^2}) \\
&= \log((2\pi\sigma^2)^{-\frac{n}{2}}) + \log(e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{10} (x_i - \mu)^2}) \\
&= -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{10} (x_i - \mu)^2 \tag{20} \\
&= -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{10} (x_i - \mu)^2
\end{aligned}$$

Thirdly, extracting the maximum likelihood estimators from the log-likelihood can be achieved by solving the maximization problem given as follows:
 $\max_{\mu, \sigma^2} l(\mu, \sigma^2; x_1, x_2, \dots, x_{10})$. Solving such a problem can be derived as follows:

$$\frac{\partial}{\partial \mu} l(\mu, \sigma^2; x_1, x_2, \dots, x_{10}) = 0 \tag{21}$$

$$\frac{\partial}{\partial \sigma^2} l(\mu, \sigma^2; x_1, x_2, \dots, x_{10}) = 0 \tag{22}$$

The solution to the problem in equation (21) is given as follow:

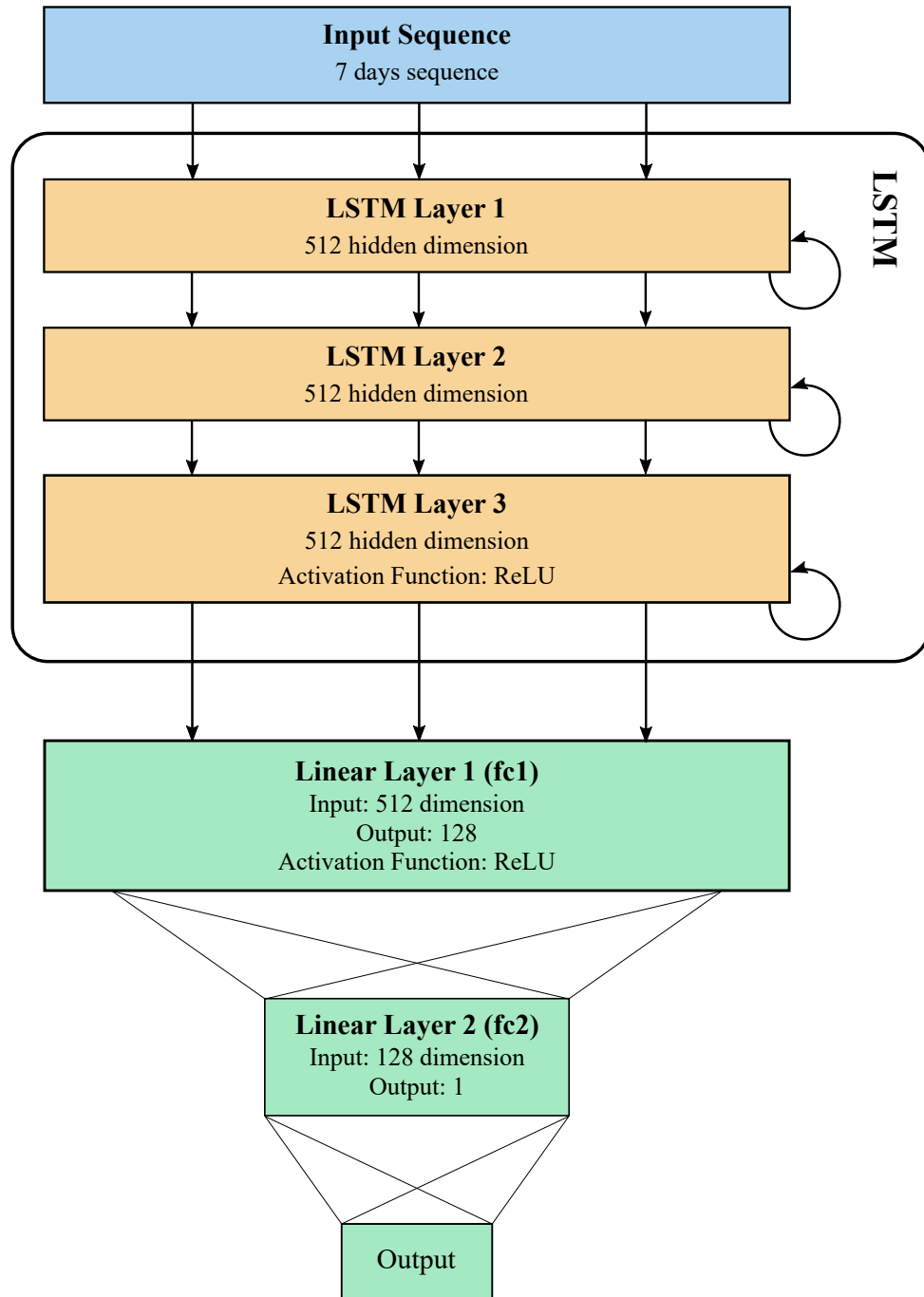
$$\left(\sum_{i=1}^{10} x_i \right) - 10\mu = 0,$$

therefore, implying that the likelihood estimator of the mean is equal to: $\mu = \frac{\sum_{i=1}^{10} x_i}{10}$

Next, in a similar way the solution to the second problem in equation (22) returns the likelihood estimator of the variance: $\sigma^2 = \frac{\sum_{i=1}^{10} (x_i - \mu)^2}{10}$

Finally, after estimating the parameters of the predicted price probability density function, it is possible to provide such distribution to the user.

Figure 7: The LSTM Model's Architecture for Cryptocurrency Price Prediction



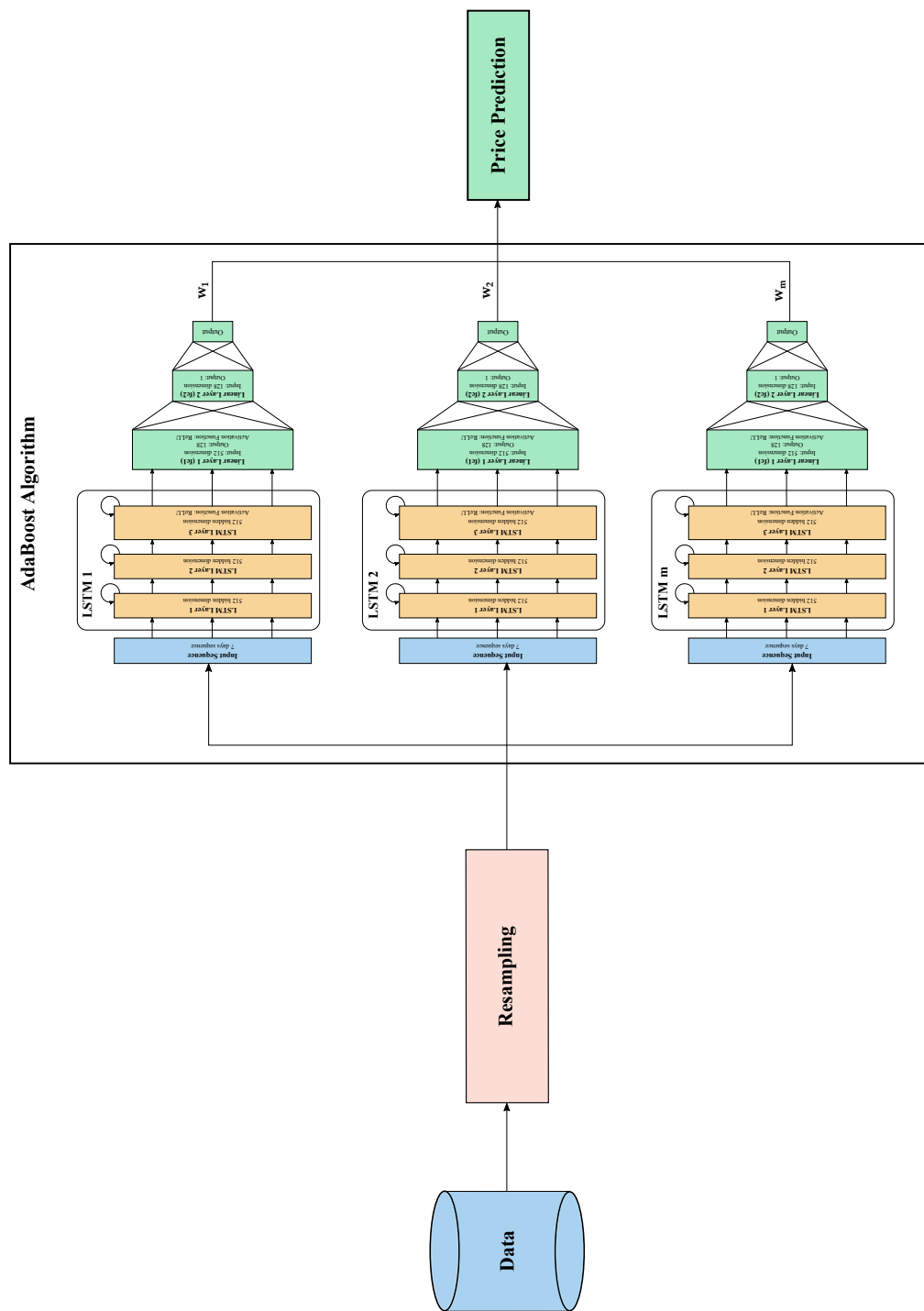


Figure 8: The AdaBoost-LSTM ensemble Learning Architecture for Cryptocurrency Price Prediction

5 Experiments and Results

This section mainly details all experimental setups used to test all hypotheses suggested by this thesis and, most notably, the results and discussion subsection that illustrates the essential findings of this approach and a fair comparison against a few of the most important research works that tackle a similar problem.

The approach and the conducted experiments are all implemented using the Python programming language, PyTorch, and Scikit-Learn frameworks to implement the models and conduct the training on the specific datasets. All the implementations are publicly available and accessible on GitHub¹¹. Moreover, to allow for an interaction with the proposed approach, a simple demo web application¹² is implemented to help viewing the results obtained by the model.

5.1 Experimental Setup

During the experimentation phase of this approach, multiple experiments have been predefined to test and validate the hypotheses suggested by this thesis. Overall, seven major experiments have been conducted to validate and evaluate the approach, each of which tackles a particular hypothesis to test. All conducted experiments have multiple common setups outlined as follows:

- Using the MSE as the training loss function.
- Using the Adam optimizer for all the experiments training as an optimization algorithm for the convergence of the model.
- An initial learning rate of 0.0003.
- Training for 200 epochs.
- The dataset was normalized using the technique proposed on the dataset subsection.
- The training was done on a GPU server with the following specs: AMD Ryzen Threadripper 1950X 16-Core Processor, 128GiB System memory, and NVIDIA GV100.
- All experiments use the MAE as a validation loss for every ten epochs.

Because each experiment tackles a specific hypothesis, there are few differences in the model's architecture and the data used on these experiments which can be summarized as follows:

¹¹<https://github.com/azeddinebouabdallah/DMCrypt>

¹²<http://dmcryptmodel.com/>

Experiment 1: To test the hypothesis “The trading data cannot be enough alone to make reliable cryptocurrency price predictions”, the approach was applied only on the trading data related to Bitcoin.

Experiment 2: Tests the hypothesis “The hash rate has a significant correlation to the cryptocurrency prices”. To test this, the approach was applied on the trading data and the hash rate of the Bitcoin network.

Experiment 3: Tests the hypothesis “The search volumes play a role in affecting the cryptocurrency prices for short term periods”. To test this hypothesis, the approach was applied on the trading data with the Google search volumes of Bitcoin.

Experiment 4: Tests the hypothesis “The social media sentiments have a direct impact on the cryptocurrency price fluctuations”. This hypothesis was tested using two sub-experiments: (1) applying the approach on the trading data and the twitter sentiments, (2) applying the approach only on the twitter sentiments toward Bitcoin.

Experiment 5: Initially, the second experiment was conducted on testing the hash rate and prices correlation. But to further test these correlations, another hypothesis was suggested that states “The blockchain information (hash rate, network difficulty, ..etc) play an important role in determining the cryptocurrency prices for short term predictions. Testing this hypothesis was conducted using two sub-experiments: (1) applying the approach only on the blockchain information, (2) applying the approach on the trading data and the Bitcoin blockchain information.

Experiment 6: The ultimate hypothesis proposed by this thesis states that “Applying all four categories of data significantly increase the price prediction accuracy for short term prediction”. Testing this hypothesis is conducted by using all the data collected on the fully proposed approach.

5.2 Baselines

Comparing and evaluating an approach against the other state-of-the-art approaches is the basis of any research study and serves as the primary validation of any work. For this reason, a total of six papers and approaches have been selected to be tested and compared against this thesis’s work. Each of these approaches proposes a different architecture and deals with different data types to make predictions for the cryptocurrency prices, specifically Bitcoin.

The first fundamental comparison that this thesis does is answering the question of “Does this approach outperform the basic and traditional approaches?” by answering and comparing the approach to other approaches from basic to more so-

phisticated can help understanding where this thesis approach stands among other proposed and validated approaches. As a result, the first selected approach to compare with the approach proposed by Alahmari [3] employs an ARIMA model. The approach uses only the trading data of Bitcoin to predict the prices on the next coming day. This approach was evaluated using the RMSE between the actual and the predicted price.

Secondly, one of the most common methods applied to financial problems especially forecasting, is the Bayesian Neural Networks (BNN), the paper proposed by Huisu, and Jaewook [18] implements a BNN network that serves the purpose of making Bitcoin price predictions on the near future. The paper evaluates the approach using the RMSE between the normalized predicted and actual values. The data interval used for this paper covers the prices from the year 2011 until 2017.

Thirdly, with the recent popularity of deep learning, multiple research papers proposed deep learning approaches that tackle the problem of cryptocurrency price predictions. Few of these approaches that were able to get considerably better results are the ones adopting one of recurrent neural network (RNN) variants. The first selected approach is proposed by Uras, et al. [43] that employs a multivariate LSTM for cryptocurrency price prediction from 2007 until 2017 with an RMSE as evaluation between the normalized output and the actual values. The second selected method is by Mudassir, et al [32] that uses an LSTM architecture to make predictions on the data from 2016 until 2019, and to simplify the evaluation and make it understandable from a user perspective, they have converted the error of one Bitcoin price to the error of a 100 dollars' worth of Bitcoin. Eventually, two other approaches that use a Gated Recurrent Unit (GRU) were selected, the first is proposed by Alkhodhairi [5] that uses the data interval from 2017 until 2020, and the second proposed by Dutta, Kumar, and Basu [11] which uses the time interval from 2010 until 2019. Both methods apply an evaluation using the RMSE between the normalized predicted and actual prices.

Finally, a research work published recently was able to achieve state-of-the-art results utilizing a traditional machine learning approach to make such predictions. The paper is written by Chevallier, et al [9]. have used a traditional AdaBoost algorithm that takes advantage of multiple decision trees weak learners to make predictions. This paper had such good improvements from other papers and was the direct inspiration of using ensemble learning for this thesis. Therefore, including it in the comparison of the results is a vital part of validating this thesis's approach. The paper evaluates the work using the price comparison between US dollars' predicted and actual bitcoin prices. The time interval of the dataset used for this approach covers the period from 2018 until 2020.

The baselines mentioned above are next used to compare with this thesis's ap-

proach to see to what extent this approach can perform better than the other ones. A primary limitation of this comparison is that all the mentioned approaches do not have a publicly available implementation of their work. As a result, implementing the approaches from scratch was a mandatory step to conduct the evaluation. In addition, because the baselines were developed and tested on a certain time interval using specific datasets, all the approaches and this thesis's approach were applied on the time interval and using the exact dataset to ensure fairness. The results of these comparisons are discussed in detail in the next sub section.

5.3 Results and Discussion

In this subsection, a detailed overview of the results obtained by each experiment is provided, along with all the evaluation metrics used. Each part is designed for the sole purpose of testing a specific hypothesis and aiming to validate hypotheses hierarchically to test and validate the thesis's proposed approach finally.

First of all, starting with the first experiment that tests the hypothesis that trading data cannot be enough alone to make reliable cryptocurrency price predictions. This hypothesis was tested using the trading data from 2012 until the end of 2020 and applied on both the LSTM model and the AdaBoost-LSTM ensemble learning. All data points within the dataset represent a single day where the open price is at 00:00 and the close price is at 23:59, resulting in a total of 3285 data points (days) split into 70% training, 15% validation, 15% testing respectively. It can be noticed that the prices nature is not similar on all the time periods, the days from 2012 until 2017 and from the end of 2018 until 2019 experienced minor fluctuations compared to other periods. Therefore, it can be challenging for most models to generalize over all the time period, Figure 9 further illustrates the split of the data for the open prices.

The LSTM and AdaBoost-Ensemble learning training is conducted on 200 epochs using the experimental setup discussed in prior sections. Table 1 represents the obtained evaluation results of both applied on the same data.

From the obtained evaluation results presented in table 1, there are three main apparent observations: firstly, adaptive boosting-LSTM ensemble learning significantly outperformed LSTM in the training and testing metrics, making its predictions much accurate than the LSTM. Secondly, It can be observed that the LSTM is better at generalizing over the data than the AdaBoost-LSTM model, as can be seen than the AdaBoost-LSTM model had good results in both the training and testing phase, but this is at the expense of a higher validation error. This reflects that AdaBoost-LSTM can have troubles generalizing over diverse data. Thirdly, even with the best results, it is still far from being helpful for making decisions. Figure 10 further illustrates what the actual price and predicted looks like for comparison,

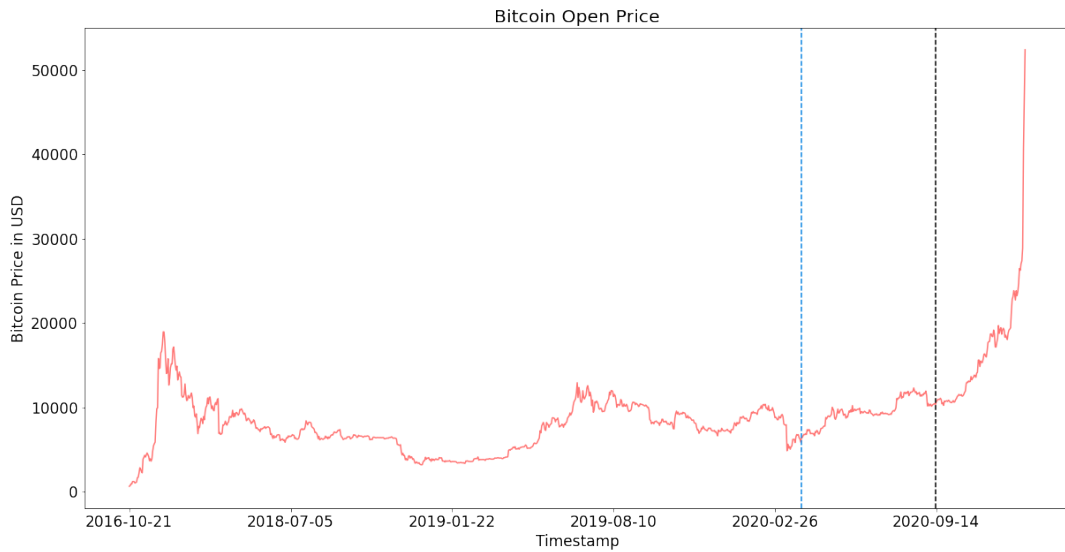


Figure 9: The training, testing, validation data split (75% training, 15% validation, 15% testing)

	LSTM	AdaBoost-LSTM ensemble learning
Training RMSE (\$)	346.507	100.593
Training MSE (\$ ²)	120067.234	10119.134
Training MAE (\$)	204.773	64.651
Validation RMSE (\$)	502.473	1097.734
Validation MSE (\$ ²)	252479.5	1205021.944
Validation MAE (\$)	321.106	709.154
Testing RMSE (\$)	502.473	272.027
Testing MSE (\$ ²)	252479.5	73999.136
Testing MAE (\$)	321.106	207.332

Table 1: First experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on trading data

where the blue line represents the real price, the red line represents the LSTM predicted price and the green line represents the AdaBoost-LSTM ensemble learning model. It is apparent visually that both predictions are not close to the real values, which can be misleading for a decision-maker. In conclusion, it is safe to assume that we accept our first hypothesis, which states, "trading data cannot be enough

alone to make cryptocurrency price predictions.” However, this cannot be proven until the subsequent experiments are conducted.

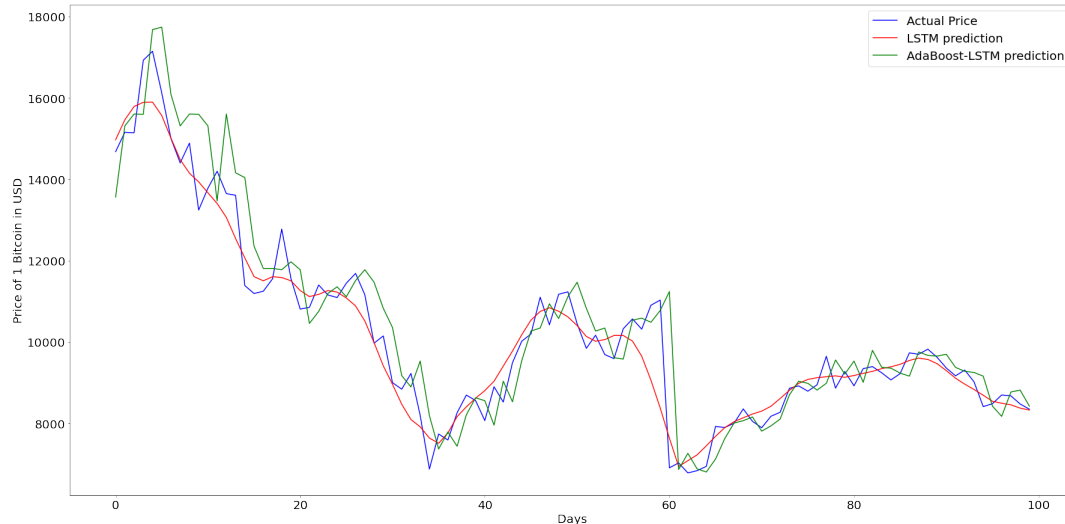


Figure 10: LSTM vs. AdaBoost Bitcoin price prediction by applying only the trading data compared to the actual price in USD

Next, we move to the second experiment that tests the hypothesis: “The hash rate has a significant correlation to the cryptocurrency prices, leading to better predictions.”. Testing this hypothesis requires conducting an experiment that compares the evaluation obtained on both models (LSTM, AdaBoost-LSTM ensemble learning) on both scenarios, first with only trading data and second with both the trading data and the hash rate. The models’ architectures are demonstrated in the approach section, with just a few adjustments in the input dimensions to accommodate our inputs. The dataset is split similarly to the previous experiment because of the similar data nature.

The LSTM and AdaBoost-Ensemble learning training is conducted on 200 epochs using the experimental setup discussed in prior sections. Table 2 represents the obtained evaluation results of both applied on the same data.

From the evaluation results presented in table 2, it is apparent that adding the hash rate as a factor in our inputs serves as a good addition for the model, as the price predictions improved significantly from what is obtained by having only the trading data, this proves the proposition made by Krisoufek [22] that the hash rate plays a role in the price fluctuations. Figure 11 further visualizes the prediction compared to the actual price of Bitcoin with and without the hash rate. Although we notice an improvement from the model that embeds only the trading data, the

	LSTM with no hash rate	AdaBoost- LSTM with no hash rate	LSTM with hash rate	AdaBoost- LSTM with hash rate
Training RMSE (\$)	346.507	100.593	299.818	41.201
Training MSE (\$ ²)	120067.234	10119.134	89891.164	1697.56\$
Training MAE (\$)	204.773	64.651	178.795	14.433
Validation RMSE (\$)	502.473	1097.734	433.68	1156.835
Validation MSE (\$ ²)	252479.5	1205021.944	188078.828	1338268.132
Validation MAE (\$)	321.106	709.154	299.766	782.773
Testing RMSE (\$)	502.473	272.027	433.680	356.554
Testing MSE (\$ ²)	252479.5	73999.136	188078.828	127131.12\$
Testing MAE (\$)	321.106	207.332	299.766	291.09

Table 2: Second experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on trading data and the hash rate

prediction error is still high to be considered an excellent source to help investors make decisions. In conclusion, we can safely accept our hypothesis, and we infer that the hash rate has an actual correlation with the cryptocurrency prices, and it helps improve the prediction results by 10% from having only the trading data.

Thirdly, after conducting the first two experiments, it is time to move on to the third experiment and test the hypothesis that states search volumes play a role in affecting the cryptocurrency prices, and by adding this criterion into the input sequence, it will high likely improve the results. Testing this hypothesis can be done using several methods, but in this thesis, both models from the second experiment are trained on trading data and training data + search volumes. It makes more sense to see the prediction error while applying search volumes from Google and observe whether there is a considerable improvement. Therefore, the LSTM and AdaBoost-Ensemble learning models are trained using 200 epochs following the previous experimental setup. Table 3 represents the obtained evaluation results of both applied on the same data.

From the table 3 evaluation results, a crucial observation can be made: despite the studies suggesting that search volumes have a great contribution to the prediction prices, the results do not strongly validate this statement. Although there is

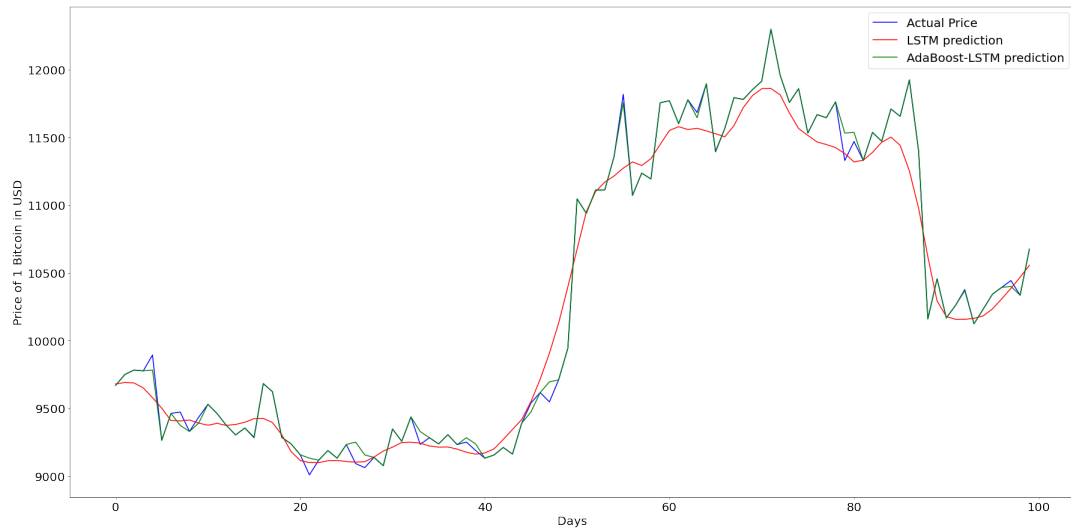


Figure 11: LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with the hashrate compared to the actual price in USD

	LSTM without search volumes	AdaBoost-LSTM without search volumes	LSTM with search volumes	AdaBoost-LSTM with search volumes
Training RMSE (\$)	346.507	100.593	352.823	89.093
Training MSE (\$ ²)	120067.234	10119.134	124484.64	7937.634
Training MAE (\$)	204.773	64.651	206.466	30.981
Validation RMSE (\$)	502.473	1097.734	522.815	1438.16
Validation MSE (\$ ²)	252479.5	1205021.944	273336.187	2068305.97
Validation MAE (\$)	321.106	709.154	369.701	940.658
Testing RMSE (\$)	502.473	272.027	519.175	277.861
Testing MSE (\$ ²)	252479.5	73999.136	274476.23	77207.124
Testing MAE (\$)	321.106	207.332	365.0	201.177

Table 3: Third experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on search volumes

an MAE improvement to some extent, it is not considered significant enough to ac-

cept the hypothesis firmly because the testing MAE improved by just a rate of 2.9%, such low increase in a highly volatile market cannot suggest that search volumes have a massive impact in our use case. Without being said, it is essential to consider this hypothesis for the time being as it is neither rejected nor accepted. In addition, search volumes can be hypothesized to perform better when merged with other attributes such as social media sentiments because they all represent a single factor: public awareness. To better understand the performance of the search volumes on the prediction, figure 12 demonstrates the comparison between the predictions obtained using the search volumes, only trading data against the actual price.

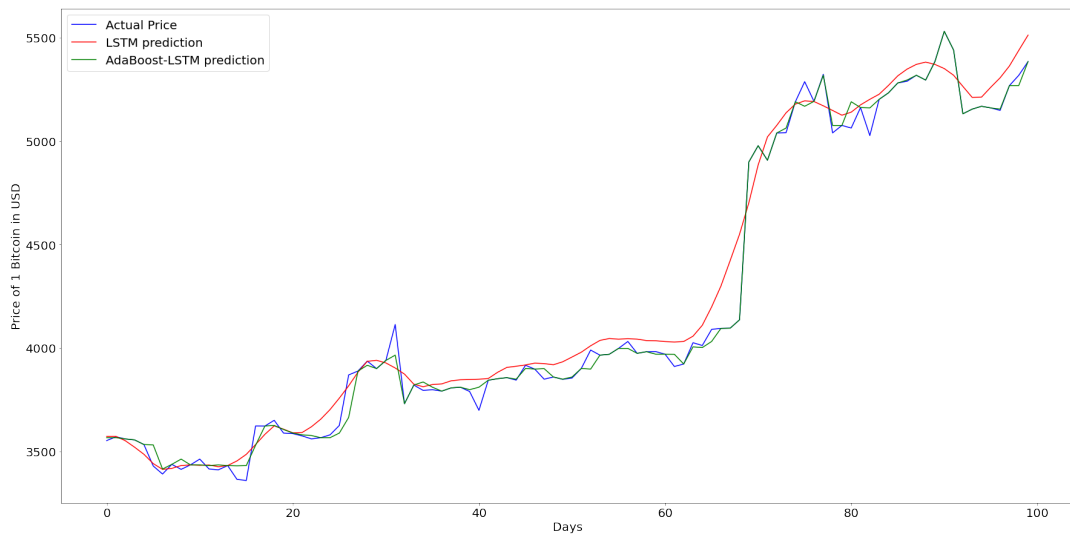


Figure 12: LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with the search volumes compared to the actual price in USD

Next, the fourth experiment tackles the hypothesis that social media sentiments correlate with cryptocurrency prices and can help achieve better predictions. This experiment is divided into two sub-experiments such that the first applies Twitter sentiments alone to the model, and the second applies both trading data and social media sentiments. The first sub-experiment aims to test to what extent sentiments can help predict the price without any additional information. If this manages to obtain an acceptable prediction error rate, it can prove that Twitter sentiments play a significant role in influencing price fluctuations. In addition, the second sub-experiment is to test whether adding social media sentiments will improve the results by a considerable amount if added to the training data.

Starting first with the part that takes only the sentiments into consideration, where the data is split similar to the previous experiments, but the only difference is that the nature of sentiments is not as volatile and diverse as the prices. Table 4 shows

the evaluation results obtained by applying only sentiment data from the first sub-experiment.

	LSTM	AdaBoost-LSTM ensemble learning
Training RMSE (\$)	2617.935	221.707
Training MSE (\$ ²)	6853588.0	49154.406
Training MAE (\$)	2120.712	59.85
Validation RMSE (\$)	8233.91	8650.681
Validation MSE (\$ ²)	67797288.0	74834288.06
Validation MAE (\$)	7210.621	6068.537
Testing RMSE (\$)	8233.91	4006.787
Testing MSE (\$ ²)	67797288.0	16054343.133
Testing MAE (\$)	7210.621	2969.586

Table 4: The first part of the fourth experiment’s evaluation results: comparison between LSTM and AdaBoost-LSTM ensemble learning applied on only Twitter sentiments

Table 4 results show an impressive discovery supporting the hypothesis that social media sentiments correlate with cryptocurrency prices. It is clear that the prediction error is very high compared to other experiments, but it is important to note that both models had only a single input that is the twitter sentiments and were able to learn the existing relation between the prices and sentiments and make a prediction as it sees fit. Figure 13 further demonstrates the prediction results by showing the actual price compared to the obtained prediction with only sentiments, and it is awe-inspiring that the AdaBoost-LSTM model was able to get such results with only a single input, but as can be seen, the LSTM model struggles at learning any patterns within the given input and appears to be giving a constant prediction overtime. As a result, it is evident that the hypothesis stating social media sentiments impact the price fluctuations is true, as the AdaBoost-LSTM model was able to have close predictions despite only having one input.

Now, moving into the following sub experiment of adding the trading data with Twitter sentiments as input to the LSTM and the AdaBoost-LSTM ensemble learning models. The data is again split similar to previous experiments and trained on 200 epochs. Table 5 further demonstrates the evaluation results obtained from applying sentiments analysis with the trading data against the models with only trading data.

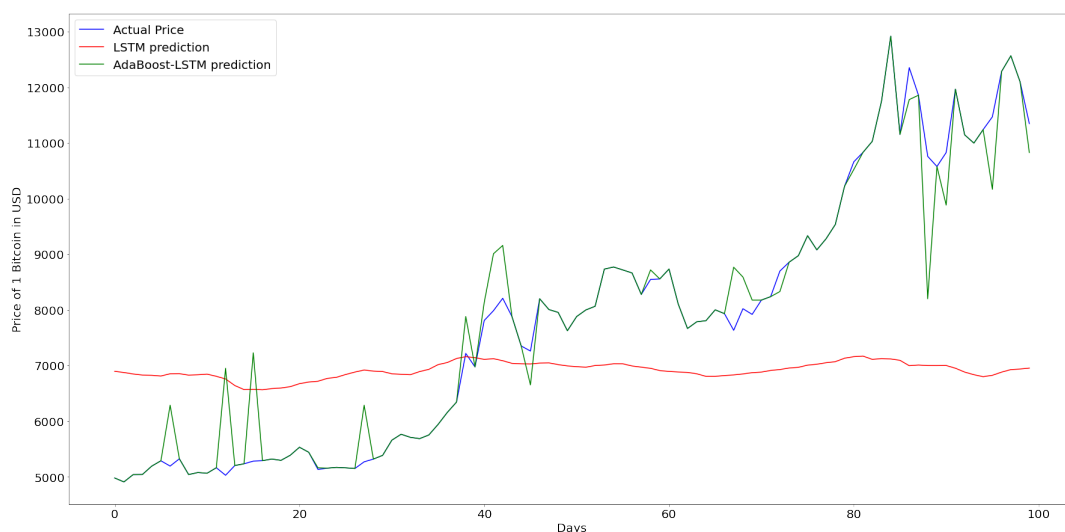


Figure 13: LSTM vs. AdaBoost Bitcoin price prediction by applying only Twitter sentiments compared to the actual price in USD

	LSTM only trading data	AdaBoost-LSTM only trading data	LSTM with sentiment analysis	AdaBoost-LSTM with sentiment analysis
Training RMSE (\$)	346.507	100.593	344.082	104.497
Training MSE (\$ ²)	120067.234	10119.134	118392.593	10919.757
Training MAE (\$)	204.773	64.651	209.094	61.134
Validation RMSE (\$)	502.473	1097.734	436.684	1098.085
Validation MSE (\$ ²)	252479.5	1205021.944	190693.062	1205791.844
Validation MAE (\$)	321.106	709.154	309.384	708.485
Testing RMSE (\$)	502.473	272.027	354.071	243.47
Testing MSE (\$ ²)	252479.5	73999.136	125366.981	59258.164
Testing MAE (\$)	321.106	207.332	312.009	201.568

Table 5: The second part of the fourth experiment’s evaluation results: comparison between LSTM and AdaBoost-LSTM ensemble learning applied on Twitter sentiments and trading data

From the obtained results given in table 5, it is important to note that the results

obtained by including sentiment analysis in the mixture generated better prediction results and lower error rates in most stages (training, testing, validation). This suggests that social media sentiments can influence the cryptocurrency price fluctuation levels and serve as a good information source for the models to learn from. Furthermore, we can also note that adaptive boosting-LSTM ensemble learning achieved a much lower error rate making the prediction much better, which is expected by following the trend of previous experiments. In addition, adding sentiment analysis to the adaptive boosting-LSTM ensemble learning improved by 15% from applying only trading on both AdaBoost-LSTM and LSTM, which can be visualized in Figure 14 that shows the comparison of the prediction obtained by all combinations on this sub-experiment.

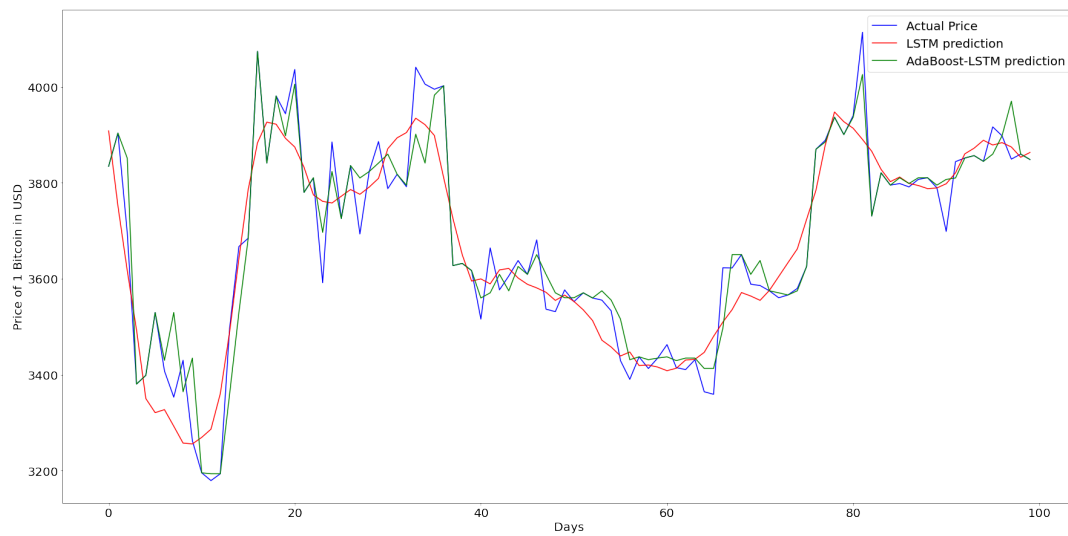


Figure 14: LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with Twitter sentiments compared to the actual price in USD

The fifth experiment is an auxiliary experiment to the second one, where initially, we tested if the hash rate affects the price prediction performance or not. However, because the results were positive towards the suggested hypothesis, it is also appropriate to test whether the other attributes of the blockchain information affect the price prediction or not, which is the purpose of this experiment. In this situation, both the LSTM and the AdaBoost-LSTM models are adjusted appropriately and trained on a total of 16 attributes that cover the trading and blockchain data. Table 6 shows this experiment's detailed evaluation results compared to the hash rate.

First of all, the evaluation results shown in table 6 demonstrate that applying blockchain information and trading data significantly improves prediction errors

	LSTM with hash rate	AdaBoost-LSTM with hash rate	LSTM with blockchain data	AdaBoost-LSTM with blockchain data
Training RMSE (\$)	299.818	41.201	280.944	96.525
Training MSE (\$ ²)	89891.164	1697.56	78929.796	9317.136
Training MAE (\$)	178.795	14.433	171.552	62.029
Validation RMSE (\$)	433.68	1156.835	1052.821	1151.279
Validation MSE (\$ ²)	188078.828	1338268.132	1108433.75	1325444.897
Validation MAE (\$)	299.766	782.773	707.377	765.778
Testing RMSE (\$)	433.680	356.554	200.263	281.607
Testing MSE (\$ ²)	188078.828	127131.12	40105.472	79302.879
Testing MAE (\$)	299.766	291.09	156.694	213.981

Table 6: Fifth experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on blockchain data

over applying only the hash rate, which is reflected in both models. However, AdaBoost-LSTM ensemble learning testing errors in this experiment are considerably higher than the LSTM architecture, where there is a 47% improvement over the overall price prediction error. To understand how much better this improvement is compared to the actual price, figure 15 further visualizes the prediction results compared to the actual ones.

Second of all, applying the blockchain data as a single input to both models as illustrated in figure 16 shows that despite not having the prices and the trading data as input, it could still learn hidden features and make extremely close price predictions. Similar to the fourth experiment, these results illustrate an existing correlation between blockchain information and cryptocurrency prices to the extent that they can help models predict the prices to some extent. Figure 16 better demonstrates this relation when observing the output given by applying only the blockchain information as input, where the price trends are kept despite the large prediction error. In conclusion, based on the results, blockchain information is an essential factor influencing cryptocurrency prices in both long-term changes and short-term fluctuations.

Finally, after executing all five experiments, the turn of the final experiment tests the ultimate hypothesis of this thesis that proposes applying all the four categories



Figure 15: LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with the blockchain data compared to the actual price in USD

of data to the model will significantly increase the price prediction performance and outperform other state-of-the-art approaches that tackle a similar problem. Furthermore, going through the previous experiment so far proves that all these four categories directly impact the price fluctuations, especially social media sentiments and blockchain data. It is highly probable that merging all these factors as a single input would make even better predictions as the model will be able to learn more hidden features and make reliable predictions. Nevertheless, running this experiment will prove if the hypothesis is accepted or rejected based on the results. This experiment executes both the LSTM and the AdaBoost-LSTM ensemble learning model on all the collected and preprocessed data with an aim to get the final prediction evaluation results. After training and evaluating both models, the evaluation results are shown in Table 7.

From observing the obtained evaluation, it is apparent that embedding all four categories of data (trading data, social media sentiments, search volumes, and blockchain data) contribute to obtaining significantly better results as there is an improvement of \$75.312 in MAE, resulting in a 36.32% improvement in the price prediction performance. However, to understand the extent of such improvements, it is essential to visualize the predicted price compared to the actual cryptocurrency price in the testing period. From Figure 17, it is evident that the predictions obtained are far close to the actual price, making the output helpful because it is less likely to mislead a decision-maker with such predictions.

To further understand the extent of the error obtained by this model, an MAE

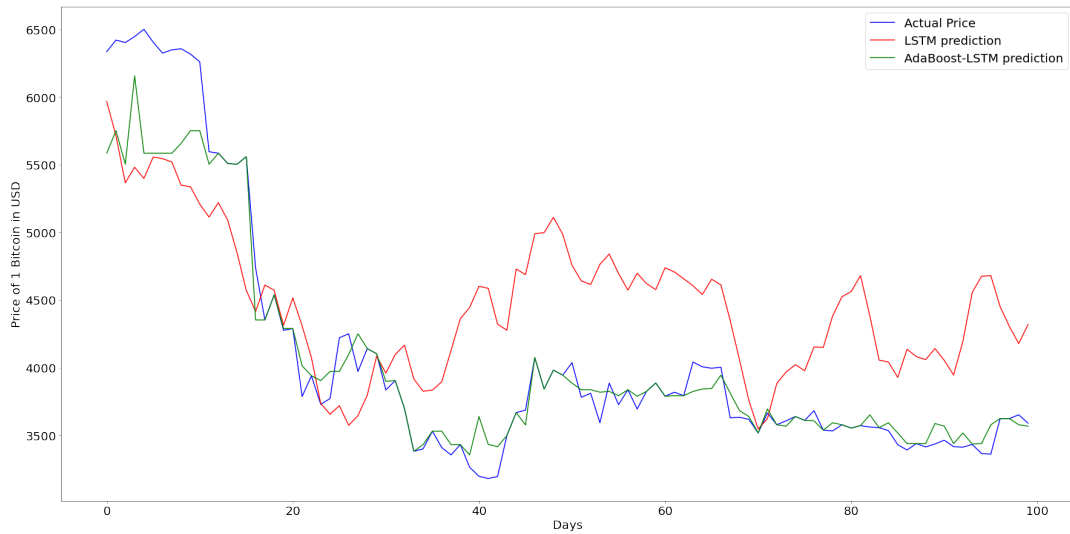


Figure 16: LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with only the blockchain data compared to the actual price in USD

	LSTM with only trading data	AdaBoost-LSTM with only trading data	LSTM with full data	AdaBoost-LSTM with full data
Training RMSE (\$)	346.507	100.593	280.013	83.564
Training MSE (\$ ²)	120067.234	10119.134	78407.63	6982.959
Training MAE (\$)	204.773	64.651	173.698	25.780
Validation RMSE (\$)	502.473	1097.734	389.245	234.718
Validation MSE (\$ ²)	252479.5	1205021.944	151512.093	55092.589\$
Validation MAE (\$)	321.106	709.154	281.399	234.666
Testing RMSE (\$)	502.473	272.027	389.245	158.929
Testing MSE (\$ ²)	252479.5	73999.136	151512.093	25258.60
Testing MAE (\$)	321.106	207.332	281.399	132.027

Table 7: Last experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on the full data

distribution of the whole testing set was calculated and visualized in Figure 18. It

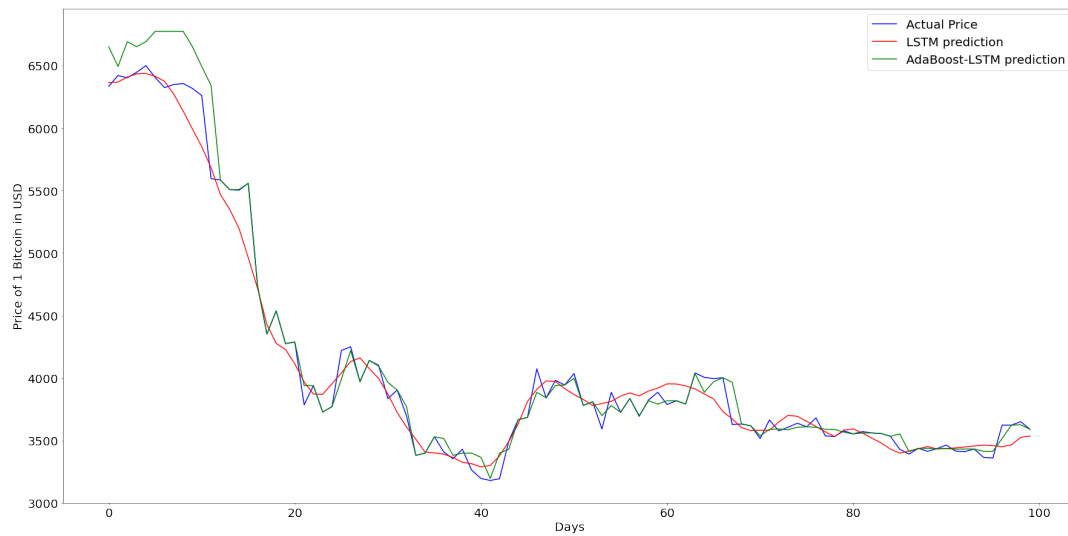


Figure 17: LSTM vs. AdaBoost Bitcoin price prediction by applying all four categories of data compared to the actual price in USD

can be seen that 68% of the times the price prediction varies by ± 500 of the price of 1 Bitcoin, making the whole model's prediction extremely close to what the actual price is considering the average price of bitcoin in the testing set is \$16553.59. Figure 19 further visualize all the modalities in one plot.

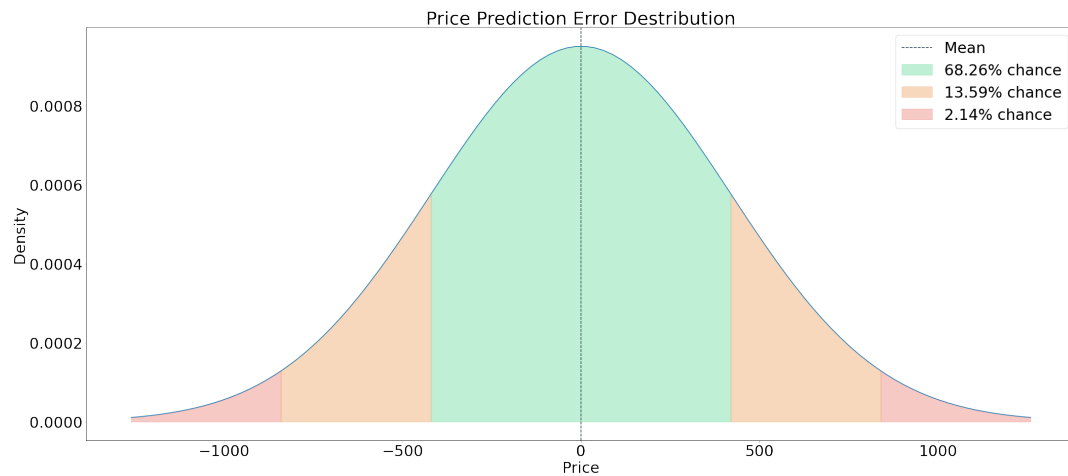


Figure 18: MAE distribution of the testing set

Over the past years, multiple research papers proposed other approaches that tackle the same problem. Some of which propose a deep learning approach, and others tackle the issue by implementing a traditional machine learning or statistical approach. One of the main hypotheses suggested by this thesis is that this proposed

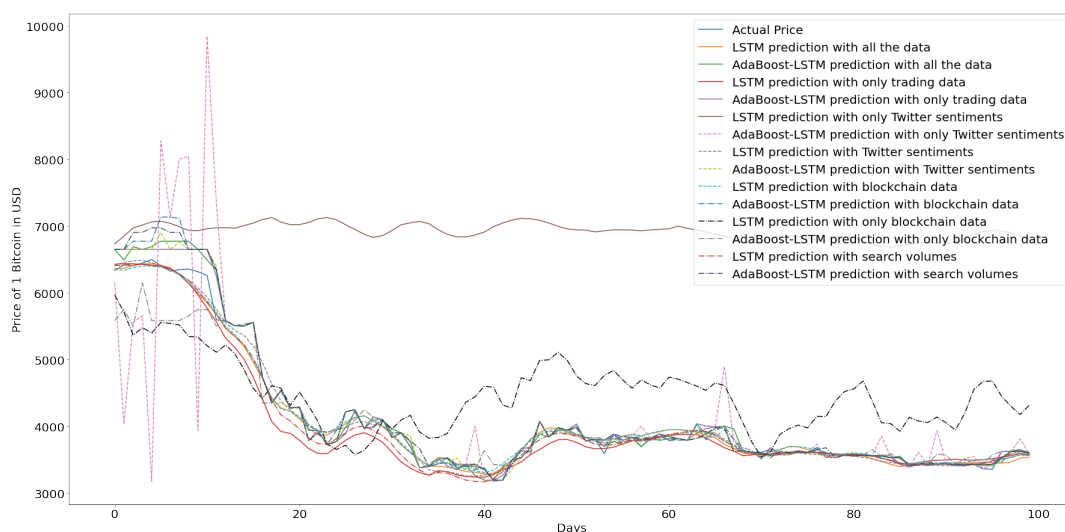


Figure 19: The Bitcoin price prediction applying multiple modalities visualized with different colors and line styles.

approach can outperform the others if applied with the given four categories of data. Therefore, it is crucial to compare the other baseline approaches and evaluate them based on similar evaluation metrics to test this claim. Therefore, all baselines models mentioned previously and DMCrypt (The approach that uses AdaBoost-LSTM and all four categories of data as input) were evaluated and tested on the same data from the same time interval to ensure that all the results obtained can be comparable together. The following Table 8 details all the evaluation results obtained by testing the models over the period ranging from (the 1st of January 2020 until the 1st of July 2020). It is important to note that all the selected baselines in this comparison make predictions for the next 24th hour for the Bitcoin cryptocurrency.

The evaluation results obtained by this comparison clearly outline that this thesis's proposed approach outperforms the other baseline models that tackle a similar problem to make predictions for the next 24th hour. As noted before, the cryptocurrency use case in this comparison is Bitcoin; therefore, these results do not necessarily mean that DMCrypt can outperform other ones if applied to other cryptocurrencies. It was not possible to conduct further experiments with other cryptocurrencies due to the limitation of data collection, especially social media data. It is recommended to tackle this challenge and evaluate the approach on other cryptocurrencies to test whether the performance drops if the cryptocurrency of choice changes. Nevertheless, the results obtained from the Bitcoin cryptocurrency provide huge support for the claim that all of the trading data, social media sentiments, blockchain data, and search volumes are major factors that influence the cryptocurrency price fluctuations and serve as an excellent source for making reliable price-

	Paper	MSE (\$ ²)	RMSE (\$)	MAE (\$)
DMCrypt	AdaBoost-LSTM	25258.60	158.929	132.027
Huisu, et al. [18]	Bayesian NN	49,125.346	221.642	184.124
Uras, et al. [43]	Multivariate LSTM	27,847.415	166.875	148.628
Mudassir, et al. [32]	LSTM	41,122.567	202.787	177.02
Chevallier, et al. [9]	AdaBoost	39,023.631	197.544	156.346
Reem, et al. [5]	GRU	43,266.496	208.006	160.44
Alahmari [3]	ARIMA	341,504.659	584.384	542.73
Dutta, el at. [11]	GRU	34,282.003	185.154	157.632

Table 8: Evaluation Comparison Between the DMCrypt Model and the results obtained by the baselines

prediction models. However, as this thesis’s motivation is to aid investors in the decision-making process, a single price prediction for the next 24th hour cannot be solely helpful. Although a considerable number of investors indeed target the concept of short-term investment, the majority of investors are more interested in long-term investments. DMCrypt approach is mainly proposed to tackle the short-term cryptocurrency price prediction and specifically make predictions for the next 24th hour, but it is interesting to push this approach to its limits and test to what extent it can make predictions and what is its performance in making long-term predictions. Therefore, the following experiment was to analyze the results obtained by DMCrypt for short and long-term predictions. The model predicts the price of the next 24th hour, and then the prediction is appended to the original input to make another prediction for the second day, and repeat this process for the next days until reaching the 30th day in the future. As can be observed by this technique, there are a couple of limitations given that we are taking the model’s input and feeding it again to get new predictions, it is not possible to obtain the values for social media sentiments and the other useful data inputs. As a result, only the first run takes all the input, but the subsequent runs will only take the previously predicted price as an input. This automatically means that the performance will drop given the fact that less data is used to infer the next day’s price, but it is worth evaluating the DMCrypt’s extent and how far it can reach with all these limitations. After executing

the experiment for all data points and calculating the MAE of the predictions, a plot was created and illustrated in Figure 20. The red line shows the average MAE of each time-window prediction, and the maximum and minimum MAE are shown by the higher and lower boundaries of the blue area respectively.

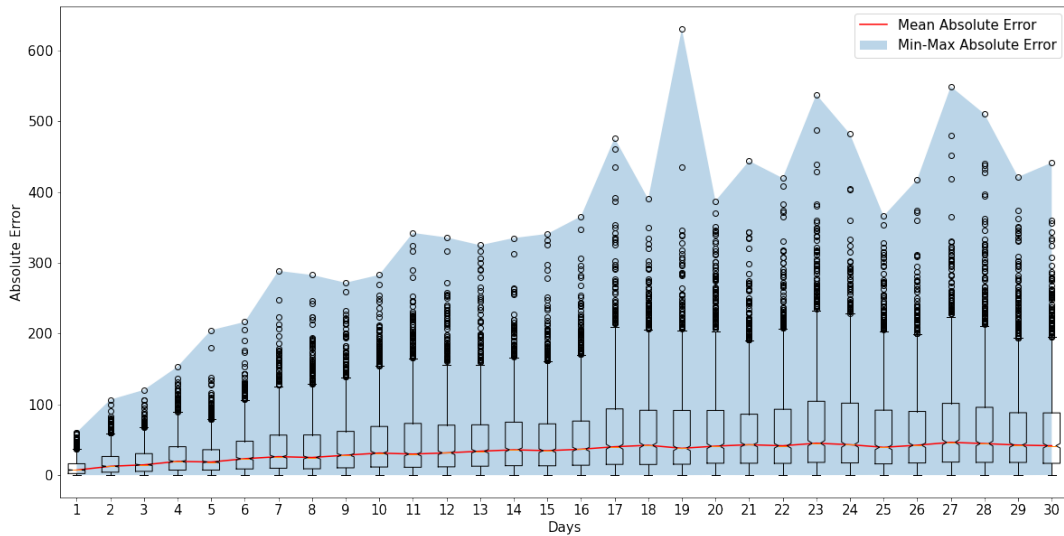


Figure 20: Bitcoin price prediction absolute testing errors over different time windows

From analyzing the plot in Figures 20 and 21, it is evident that the DMCrypt approach can get a precise price prediction in the short term period reaching its peak performance when targeting the one-day prediction, but the performance degrades the more long-term prediction it goes. Such that not only the prediction error increases with time but also the uncertainty of the model rockets, the high and low MAE range increases and reaches its peak by the prediction of the 30th day. These results prove that the model is indeed great at short-term predictions, but the performance drops considerably with long-term predictions, making this approach less reliable when targeting long-term predictions.

As discussed by the motivation of this thesis, DMCrypt aims to provide a decision-making aid for investors to make the right actions at the right moment. A single price prediction for the next day can be misleading into making an investment decision because sometimes a day might have high volatility, such that the difference between the open and close price is extreme. As a result, making a price distribution that can convey the certainty of the prediction to the user is vital to providing decent help. The technique used in obtaining such distributions is covered in the

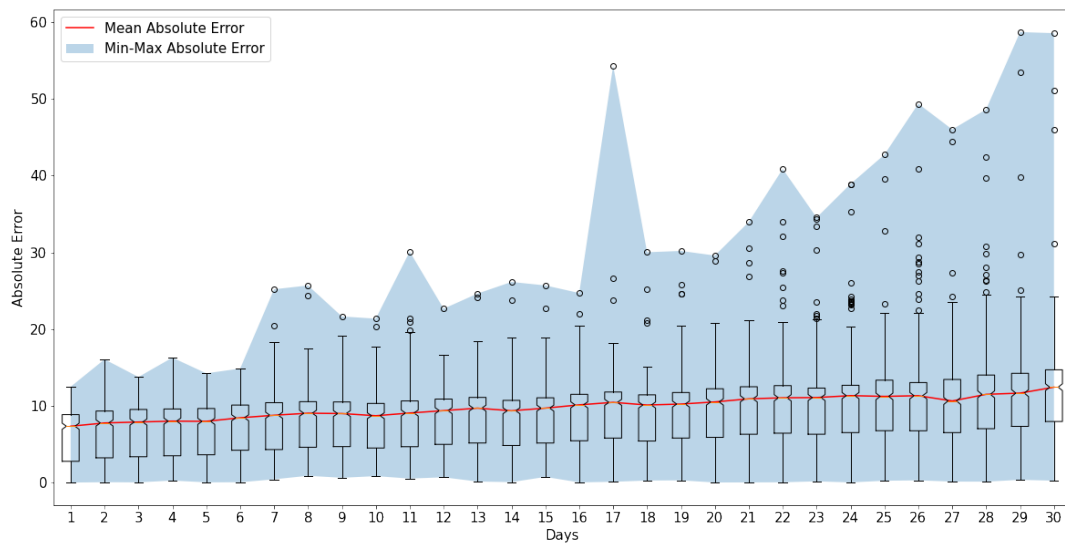


Figure 21: Bitcoin price prediction absolute validation errors over different time windows

previous section. To show few results obtained by this technique, Figure 22 shows the predicted price distribution of 6 selected days.

It is noticeable that the price prediction certainty can differ from one day to another, where if there is a massive volatility level on that particular day, the price distribution is closer to being flat and having a high standard deviation, which conveys the information that the price is highly likely to fluctuate from the single model's prediction. In contrast, when the model gives a prediction with a high certainty level, it is apparent that the price distribution has a high peak and low standard deviation. All these prediction distributions are an excellent source for an investor for decision-making as it provides a clear interpretation of how confident the model is for a given prediction on a particular day.

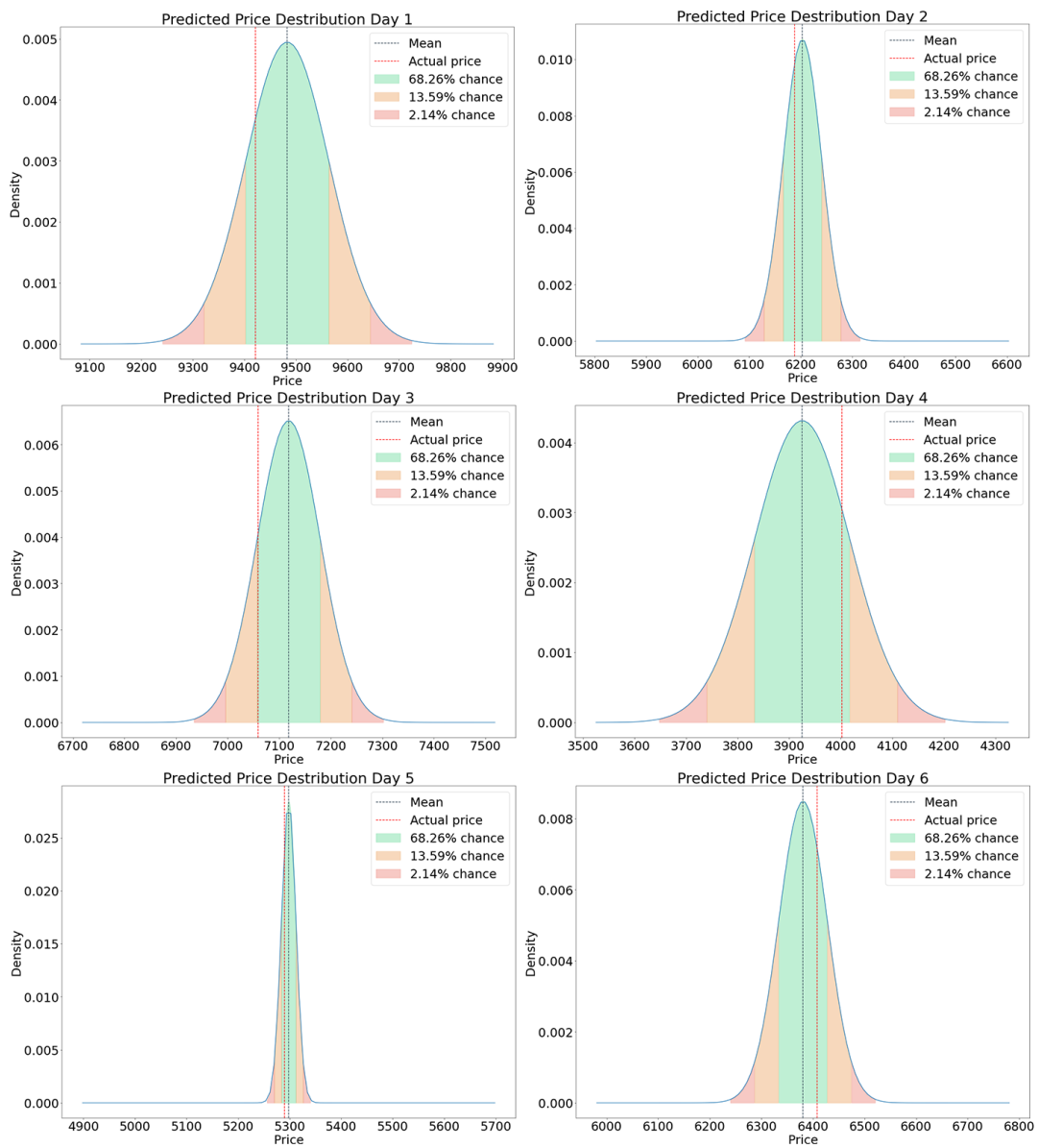


Figure 22: Predicted price distribution of six randomly selected days

6 Conclusion and Future Work

This thesis's main contribution is to make cryptocurrency price predictions further reliable and credible for assisting investors in their decision-making journey. It proposes DMCrypt, an ensemble learning approach that adopts multiple LSTM weak learners and an adaptive boosting algorithm to make price predictions. Each LSTM model is trained on a sampled subset of the dataset and assigned a weight according to its performance. The value of each model is related to the weight it represents (high weight expresses low error prediction). All LSTM models consist of an LSTM with three stacked layers and two fully connected layers with a ReLU activation function between each of the layers. Finally, the final price prediction is made the average of the output of all the weak learners multiplied by their weights. The usage of this proposed approach on the dataset containing all the factors contributing to the cryptocurrency price fluctuations contributed to achieving low error rate predictions. Furthermore, because a single prediction can be hard to trust by an investor to make decisions, DMCrypto provides a price prediction distribution along the actual prediction to inform the user of the certainty of the forecast.

Multiple well-prepared experiments were executed on different categories of data to test various suggested hypotheses and validate this thesis proposed approach (DMCrypt). The experiments tested and evaluated all the multimodalities of the approach individually to observe each multimodality's contributions to the overall performance. Results obtained show remarkable prediction performance compared to other state-of-the-art approaches as DMCrypt was able to outperform all other baselines on Bitcoin cryptocurrency price predictions. Although having impressive prediction results, the approach was not tested on other cryptocurrencies, and it is not guaranteed that similar results will be achieved on other cryptocurrencies. Another noticeable observation from all the experiments is the high validation results of Adaboost-LSTM compared to LSTM. This suggests that Adaboost-LSTM has trouble generalizing all the data because of its high volatility. This raises a concern that the model may need regular retraining to ensure its performance over time; nevertheless, this needs more experimentation to validate and be sure of this assumption. For future work, it is interesting to test this thesis's approach and experiment more on other cryptocurrencies to evaluate the prediction performance, as it will be more helpful if it can tackle other cryptocurrencies besides Bitcoin. Furthermore, because the AdaBoost-LSTM shows signs of having trouble generalizing over high volatile data, it is interesting to propose other normalization techniques to reduce the volatility and help the approach generalize more.

7 List of Acronyms

RNN Recurrent neural network	7
LSTM Long Short Term Memory	2
GRU Gated Recurrent Unit	9
Tanh Hyperbolic Tangent Activation Function	7
MAE Mean Absolute Error	12
MSE Mean Square Error	11
RMSE Root Mean Square Error	12
BiLSTM Bidirectional Long short-term memory	9
ANN Artificial Neural Networks	14
BRNN Bidirectional Recurrent Neural Networks	9
MLP Multi-Layer perceptron	15
CNN Convolutional neural network	15
AdaBoost Adaptive Boosting	2

List of Figures

1	A Simplified Architecture of RNN cells [30]	9
2	A Simplified Architecture of an LSTM cell [12]	10
3	A Simplified Architecture of a GRU cell [47]	10
4	Sentiment analysis on tweets using VADER and Deeply Moving	22
5	Blockchain data plots	27
6	Google search volumes for the word Bitcoin	28
7	The LSTM Model's Architecture for Cryptocurrency Price Prediction	34
8	The AdaBoost-LSTM ensemble Learning Architecture for Cryptocurrency Price Prediction	35
9	The training, testing, validation data split (75% training, 15% validation, 15% testing)	40
10	LSTM vs. AdaBoost Bitcoin price prediction by applying only the trading data compared to the actual price in USD	41
11	LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with the hashrate compared to the actual price in USD	43
12	LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with the search volumes compared to the actual price in USD	44
13	LSTM vs. AdaBoost Bitcoin price prediction by applying only Twitter sentiments compared to the actual price in USD	46
14	LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with Twitter sentiments compared to the actual price in USD	47
15	LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with the blockchain data compared to the actual price in USD	49
16	LSTM vs. AdaBoost Bitcoin price prediction by applying the trading data with only the blockchain data compared to the actual price in USD	50
17	LSTM vs. AdaBoost Bitcoin price prediction by applying all four categories of data compared to the actual price in USD	51
18	MAE distribution of the testing set	51
19	The Bitcoin price prediction applying multiple modalities visualized with different colors and line styles.	52
20	Bitcoin price prediction absolute testing errors over different time windows	54
21	Bitcoin price prediction absolute validation errors over different time windows	55
22	Predicted price distribution of six randomly selected days	56

List of Tables

1	First experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on trading data	40
2	Second experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on trading data and the hash rate	42
3	Third experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on search volumes	43
4	The first part of the fourth experiment's evaluation results: comparison between LSTM and AdaBoost-LSTM ensemble learning applied on only Twitter sentiments	45
5	The second part of the fourth experiment's evaluation results: comparison between LSTM and AdaBoost-LSTM ensemble learning applied on Twitter sentiments and trading data	46
6	Fifth experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on blockchain data	48
7	Last experiment's evaluation results comparison between LSTM and AdaBoost-LSTM ensemble learning applied on the full data	50
8	Evaluation Comparison Between the DMCrypt Model and the results obtained by the baselines	53

References

- [1] David Aboody, Omri Even-Tov, Reuven Lehavy, and Brett Trueman. Overnight returns and firm-specific investor sentiment. *Journal of Financial and Quantitative Analysis*, 53(2):485–505, 2018.
- [2] Jethin Abraham, Daniel Higdon, John Nelson, and Juan Ibarra. Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1(3):1, 2018.
- [3] Saad Ali Alahmari. Using machine learning arima to predict the price of cryptocurrencies. *The ISC International Journal of Information Security*, 11(3):139–144, 2019.
- [4] Rayner Alfred, Joe Henry Obid, Mohd Hanafi Ahmad Hijazi, Ag Asri Ag Ibrahim, et al. A performance comparison of statistical and machine learning techniques in learning time series data. *Advanced Science Letters*, 21(10):3037–3041, 2015.
- [5] Reem K Alkhodhairi, Shahad R Aljalhami, Norah K Rusayni, Jowharah F Alshobaili, Amal A Al-Shargabi, and Abdulatif Alabdulatif. Bitcoin candlestick prediction with deep neural networks based on real time data. *CMC-COMPUTERS MATERIALS & CONTINUA*, 68(3):3215–3233, 2021.
- [6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [7] Mathias Berglund, Tapani Raiko, Mikko Honkala, Leo Kärkkäinen, Akos Vetek, and Juha T Karhunen. Bidirectional recurrent neural networks as generative models. *Advances in Neural Information Processing Systems*, 28:856–864, 2015.
- [8] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [9] Julien Chevallier, Dominique Guégan, and Stéphane Goutte. Is it possible to forecast the price of bitcoin? *Forecasting*, 3(2):377–420, 2021.
- [10] David LEE Kuo Chuen, Li Guo, and Yu Wang. Cryptocurrency: A new investment opportunity? *The Journal of Alternative Investments*, 20(3):16–40, 2017.
- [11] Aniruddha Dutta, Saket Kumar, and Meheli Basu. A gated recurrent unit approach to bitcoin price prediction. *Journal of Risk and Financial Management*, 13(2):23, 2020.
- [12] Alex Graves. Long short-term memory. In *Supervised sequence labelling with recurrent neural networks*, pages 37–45. Springer, 2012.

- [13] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.
- [14] James V Hansen, James B McDonald, and Ray D Nelson. Time series prediction with genetic-algorithm designed neural networks: An empirical comparison with modern statistical models. *Computational Intelligence*, 15(3):171–184, 1999.
- [15] Mohammad Asiful Hossain, Rezaul Karim, Ruppa Thulasiram, Neil DB Bruce, and Yang Wang. Hybrid deep learning model for stock price prediction. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1837–1844. IEEE, 2018.
- [16] Xin Huang, Wenbin Zhang, Yiyi Huang, Xuejiao Tang, Mingli Zhang, Jayachander Surbiryala, Vasileios Iosifidis, Zhen Liu, and Ji Zhang. Lstm based sentiment analysis for cryptocurrency prediction. *arXiv preprint arXiv:2103.14804*, 2021.
- [17] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014.
- [18] Huisu Jang and Jaewook Lee. An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *Ieee Access*, 6:5427–5437, 2017.
- [19] Kaustubh Khare, Omkar Darekar, Prafull Gupta, and VZ Attar. Short term stock price prediction using deep learning. In *2017 2nd IEEE international conference on recent trends in electronics, information & communication technology (RTE-ICT)*, pages 482–486. IEEE, 2017.
- [20] Young Bin Kim, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, and Chang Hun Kim. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PloS one*, 11(8):e0161197, 2016.
- [21] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.
- [22] Ladislav Kristoufek. What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. *PloS one*, 10(4):e0123923, 2015.
- [23] Deepak Kumar and SK Rath. Predicting the trends of price for ethereum using deep learning techniques. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pages 103–114. Springer, 2020.
- [24] Connor Lamon, Eric Nielsen, and Eric Redondo. Cryptocurrency price prediction using news and social media sentiment. *SMU Data Sci. Rev*, 1(3):1–22, 2017.

- [25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [26] Ioannis E Livieris, Emmanuel Pintelas, Stavros Stavroyiannis, and Panagiotis Pintelas. Ensemble deep learning models for forecasting cryptocurrency time-series. *Algorithms*, 13(5):121, 2020.
- [27] Isaac Madan, Shaurya Saluja, and Aojia Zhao. Automated bitcoin trading via machine learning algorithms. URL: <http://cs229.stanford.edu/proj2014/Isaac%20Madan>, 20, 2015.
- [28] Navin Kumar Manaswi. Rnn and lstm. In *Deep Learning with Applications Using Python*, pages 115–126. Springer, 2018.
- [29] Sean McNally, Jason Roche, and Simon Caton. Predicting the price of bitcoin using machine learning. In *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)*, pages 339–343. IEEE, 2018.
- [30] Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, and Marc’Aurelio Ranzato. Learning longer memory in recurrent neural networks. *arXiv preprint arXiv:1412.7753*, 2014.
- [31] Stefan Mittnik, Nikolay Robinzonov, and Martin Spindler. Stock market volatility: Identifying major drivers and the nature of their impact. *Journal of banking & Finance*, 58:1–14, 2015.
- [32] Mohammed Mudassir, Shada Bennbaia, Devrim Unal, and Mohammad Hamoudeh. Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*, pages 1–15, 2020.
- [33] Sergey Nasekin and Cathy Yi-Hsuan Chen. Deep learning-based cryptocurrency sentiment construction. Available at SSRN 3310784, 2019.
- [34] Mahla Nikou, Gholamreza Mansourfar, and Jamshid Bagherzadeh. Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, 26(4):164–174, 2019.
- [35] Emmanuel Pintelas, Ioannis E Livieris, Stavros Stavroyiannis, Theodore Kotsilieris, and Panagiotis Pintelas. Investigating the problem of cryptocurrency price prediction: a deep learning approach. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 99–110. Springer, 2020.
- [36] Christophe Schinckus. The good, the bad and the ugly: An overview of the sustainability of blockchain technology. *Energy Research & Social Science*, 69:101614, 2020.

- [37] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017.
- [38] Apeksha Shewalkar, Deepika Nyavanandi, and Simone A Ludwig. Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru. *Journal of Artificial Intelligence and Soft Computing Research*, 9(4):235–245, 2019.
- [39] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.
- [40] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.
- [41] Shaolong Sun, Yunjie Wei, and Shouyang Wang. Adaboost-lstm ensemble learning for financial time series forecasting. In *International Conference on Computational Science*, pages 590–597. Springer, 2018.
- [42] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [43] Nicola Uras, Lodovica Marchesi, Michele Marchesi, and Roberto Tonelli. Forecasting bitcoin closing price series using linear regression and neural networks models. *PeerJ Computer Science*, 6:e279, 2020.
- [44] Liuqing Yang, Xiao-Yang Liu, Xinyi Li, and Yinchuan Li. Price prediction of cryptocurrency: An empirical study. In *International Conference on Smart Blockchain*, pages 130–139. Springer, 2019.
- [45] Wang Yiyi and Zang Yeze. Cryptocurrency price analysis with artificial intelligence. In *2019 5th International Conference on Information Management (ICIM)*, pages 97–101. IEEE, 2019.
- [46] Saizheng Zhang, Yuhuai Wu, Tong Che, Zhouhan Lin, Roland Memisevic, Russ R Salakhutdinov, and Yoshua Bengio. Architectural complexity measures of recurrent neural networks. *Advances in neural information processing systems*, 29:1822–1830, 2016.
- [47] Rui Zhao, Dongzhe Wang, Ruqiang Yan, Kezhi Mao, Fei Shen, and Jinjiang Wang. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, 65(2):1539–1548, 2017.

- [48] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212, 2016.