HERIOT-WATT UNIVERSITY

MASTERS THESIS

Predicting the evolution of the Bitcoin price using machine learning models

Author:
Joseph CHARTOIS

Supervisor:
Dr. Lilia GEORGIEVA

A thesis submitted in fulfilment of the requirements for the degree of MSc Data Science

in the

School of Mathematical and Computer Sciences

December 2021

HERIOT WATT UNIVERSITY

Declaration of Authorship

I, Joseph Chartois, declare that this thesis titled, 'Predicting the evolution of the Bitcoin price using machine learning models' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: Joseph Chartois

Date: 09/12/2021

«*What we want is a machine that can learn from experience.*»

Alan Turing

Abstract

Since its creation in January 2009, Bitcoin has attracted a lot of interest. There are two reasons for this, firstly because it was designed using a disruptive technology called blockchain, and secondly because its value has grown significantly. Indeed, the concept of blockchain has allowed it to achieve the success that the first cryptocurrencies such as DigiCash and Bit Gold failed to achieve. But the main reason why Bitcoin is so popular is due to the significant increase in its stock market value in recent years.

There has been a lot of research done to anticipate changes in the value of Bitcoin or to determine its price. However, the value presents a high volatility and due to the lack of knowledge of its functioning, it still generates a lot of mistrust.

The objective of this research is to determine which machine learning model is best able to predict the evolution of the Bitcoin price. Particular attention will be paid to updating the results using recent data, improving them by adding relevant features and searching for new areas of improvement in the determination of the hyperparameters of machine learning models.

Acknowledgements

I would like to show my gratitude to Dr. Lilia Georgieva for allowing me to work on this project, as well as for her advice and support throughout my research.

Contents

List of Figures

List of Tables

Abbreviations

ANN	**A**rtificial **n**eural **n**etwork
AUC	**A**ir **u**nder **c**urve
BNC:BLX	**B**raveNewCoin **l**iquid **i**ndex for **B**itcoin
BTC1!CME	**B**itcoin **CME f**utures
BNN	**B**ayesian **n**eural **n**etwork
BTC	**Bitc**oin
CME	**C**hicago **m**ercantil **e**xchange
CNN	**C**onvolutional **n**eural **n**etwork
DNN	**D**ense **n**eural **n**etwork
EMA	**E**xponential **m**oving **a**verage
LR	**L**inear **r**egression
LTSM	**L**ong **s**hort-**t**erm **m**emory
MA	**M**oving **a**verage
MAP	**M**aximum **a p**osterior
MLE	**M**arginal **l**ikelihood **e**stimation
MLP	**M**ulti**l**ayer **p**erceptron
OHLC	**O**pen **h**igh **l**ow **c**lose and **v**olume
PoW	**P**roof **o**f **w**ork
PoS	**P**roof **o**f **s**take
ReLU	**Re**ctified **l**inear **u**nits
RMSE	**R**oot **m**ean **s**quare **e**rror
RNN	**R**ecurrent **n**eural **n**network
ROC curve	**R**eceiver **o**perating **c**haracteristic curve
ROCAUC	**R**eceiver **o**perating **c**haracteristic/**A**rea **u**nder the **c**urve

RSI	**R**elative **s**trength **i**ndex
SVM	**S**upport **v**ector **m**achine
TN	**T**rue **n**egative
TNR	**T**rue **n**egative **r**ate or specificity
TP	**T**rue **p**ositive
TPR	**T**rue **p**ositive **r**ate or sensitivity
tps	**T**ransaction **p**er **s**econd
USD	**U**nited **S**tates **d**ollar
VIF	**V**ariable **i**nfation **f**actors
XBP	**B**itcoin **P**rice Index

Chapter 1

Introduction

1.1 Aim

Bitcoin has become very popular, both in society and in the scientific community, in fact (Redman, 2019) attests that no less than 13700 articles have been published in 2019 on Google Scholar contain the word Bitcoin. As a result, numerous studies have been carried out to test different models which have been used to predict either the price evolution or directly the price of bitcoin.

The objective of this study is to compare five state-of-the-art machine learning models in an identical environment. The specificity of this study lies in the attention paid to the construction of the dataset. Indeed, we will use data going back to 2012, which is close to the creation of Bitcoin. Furthermore, we will not neglect the importance of recent data, with data going back to July 2021.

Many models such as LR, SVM, CNN, LSTM and BNN have already been studied in (McNally et al., 2018), (Jang and Lee, 2018) and (Mallqui and Fernandes, 2019). Based on this research, we will select the best performing models, those which have an accuracy of at least 52% and which are justified by statistical indicators such as F1-score, accuracy, specificity and sensitivity.

1.2 Objectives

The objectives of this research are as follows:

1. Collect data over a time interval of about ten years, with the most recent data in 2021 from the following sources such as Coindesk, Quandl, LunarCrush and the Bitcoin blockchain.
2. Clean up the new dataset by checking for incorrect or missing values, homogenise values, check timestamp format for consistency
3. Preprocess the data by normalising it and create two datasets. The first one includes all the features, and the second one includes a selection of the twenty most relevant parameters.
4. Train and test models to determine their hyperparameters.
5. Compare the results of the models and rank them according to their accuracy.
6. Evaluate the models.

Chapter 2

Literature Review

2.1 Background

2.1.1 Context cryptocurrencies

In order to determine the evolution of the bitcoin price, it is necessary to understand the context in which the first cryptocurrencies emerged as well as the context in which the cryptocurrency market is today.

A cryptocurrency can be defined as a virtual currency that combines two concepts, cryptography to ensure the security of transactions and the peer-to-peer system to ensure the independence of the currency.

(CoinDesk, 2021*b*) is a data aggregator website bought by Binance, one of the main players of cryptocurrency exchange platforms. Although Bitcoin currently dominates the cryptocurrency market, as CoinMarketCap points out with a Market Dominance Index of 60%, it did not achieve this status by being the first cryptocurrency. Indeed, others predate it such as DigiCash which was created in 1989 by David Chaum. However, it was not accepted by the banking industry and went bankrupt in 1998. One of the main reasons for this failure was the need to find a third party to validate the transactions, i.e. to create a partnership with a banking organisation that would assume this responsibility.

Despite Bitcoin's dominant position in the cryptocurrency market, many competing currencies are created each year. (CoinMarketCap, 2021) counted 8364 virtual currencies in January 2021. Some of these currencies offer notable technological advances compared to Bitcoin. For example, Monero is an untraceable currency that ensures the anonymity of its users, whereas Bitcoin keeps the history of all acquirers in its blockchain. Cardano allows 257 transactions per second (tps), while (Georgiadis, 2019) has determined that Bitcoin's theoretical maximum tps value is 27. IOTA has no transaction fees, whereas Bitcoin's fees increase as demand grows and transactions accumulate and fail to be validated. It is also worth noting that the fees associated with each Bitcoin transaction turn the currency into a simple stock. As a result, it loses its primary quality as a currency of exchange, while (SantanderBank, 2021) offers BillPay which allows Americans to pay their bills. One risk for Bitcoin is the emergence of a competitor that would be accepted by a wider community and thus influence its price.

Now that the context in which Bitcoin is evolving has been described, let's see how this currency works.

2.1.2 Bitcoin

In 2008 an individual (or group of individuals) going by the pseudonym Satoshi Nakamoto published (Nakamoto, 2008) in which he explained the need to create an online payment method that is completely non-reversible. The objective was to make transactions without relying on a trusted third party, but based on cryptographic proof to validate transactions.

In order to eliminate the need for a trusted third party for transactions, Nakamoto took inspiration from (Back, 2002) to create the Bitcoin blockchain. In this blockchain each coin represents a chain of signatures of the owner of that coin. Figure 2.1 shows how the Bitcoin blockchain exchange process works. When a transaction takes place the seller will sign the hash of the previous transaction and the buyer's public key and then add this information to the corner.

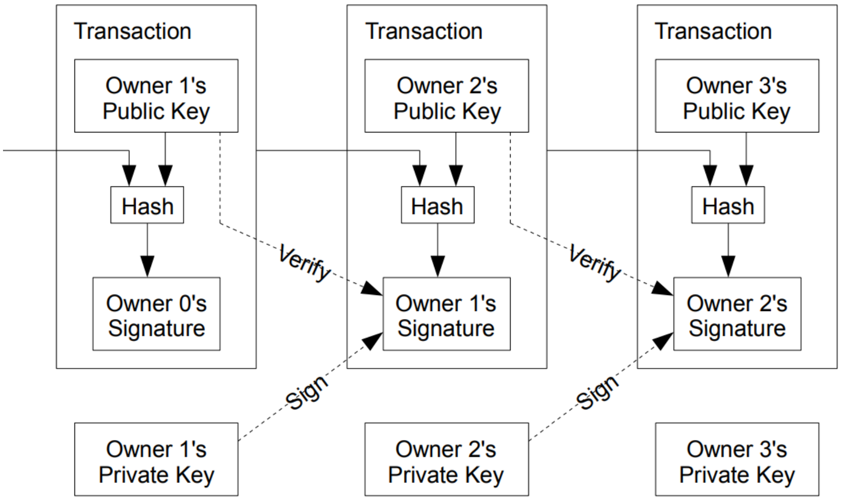

FIGURE 2.1: Blockchain transaction (Nakamoto, 2008)

Having a hash system is not enough to ensure network security and prevent double-spending. Double-spending is a method of spending the same coin several times, which can happen when a coin is duplicated by mistake or falsified. To avoid this, the network must reach a consensus.

Currently, the two most used methods to reach a consensus have been described in (Berg et al., 2020) and (Nicolas, 2014):

- Proof of work (PoW), often called "mining", involves individuals scattered all over the world, the miners. These individuals then compete with each other, using their computers to solve mathematical problems (PoW). The first one to solve the PoW will be able to register new block transactions in the blockchain and will receive a reward in Bitcoin.

- Proof of stake (PoS) was designed to avoid the energy and system wear and tear caused by PoW. With PoS when a transaction is to be added, an individual in the network is selected to validate it. The probability that the individual will be selected depends on the number of coins he owns. The more coins he owns, the higher the probability that he will be selected.

To alter the Bitcoin blockchain, it would be necessary to modify all blocks by resolving a PoW for each block and then take control of more than 50% of the peer-to-peer network in order to obtain a consensus for these modifications to be accepted. This is an unlikely event, but one that has already occurred, as reported by (Hertig, 2019). A 51 percent attack was carried out by BTC.com and BTC.top to stop an individual who had exploited a bug to recover Bitcoins.

PoW is not without its flaws, its technological limitations and the image it projects are factors that can influence the Bitcoin price.

- Indeed, since there is competition, each individual accumulates more and more computer resources which leads to an increase in the energy consumed. Moreover, as (Corbet et al., 2021) points out, the energy used to power the largest mining structures is in China and much of it is powered by coal-fired plants, which has a negative environmental impact.
- This competition leads to miners grouping together in structures called mining pools. This induces a centralization of resources that goes against what Bitcoin was meant to be, i.e. a decentralized currency. (Blockchain, 2021) confirms this by saying that there are four Chinese mining pools, Poolin, F2Pool, AntPool and ViaBTC that control more than 50% of Bitcoin's total mining power.

This context, which mixes technological, ecological and geopolitical competitions, creates a complex pattern that influences the Bitcoin price. In (Nti et al., 2020), (El Naqa and Murphy, 2015), (Aivaliotis et al., 2017) and (Bhamare and Suryawanshi, 2018), machine learning model have demonstrated their ability to detect complex patterns in order to make predictions, and in the next section we will see how the most promising models work.

2.1.3 Machine learning models

As (Alessandretti et al., 2018) points out, traditional functional analysis methods have been completely overtaken by machine learning models for prediction. Moreover, since 2009 when Bitcoin was created, its volatile price has been rising steadily, generating large amounts of data with complex patterns that justify the use of machine learning models.

We will discuss the fundamental concepts of machine learning and introduce some of them such as logistic regression, support-vector machines, convolutional neural network, long short-term memory and Bayesian neural network.

2.1.3.1 Logistic regression

We are trying to determine whether the Bitcoin price is increasing or decreasing, this is a binary problem. Logistic regression, despite the term "regression", is usually used to perform a dichotomous classification.

$$z = b + w_1x_1 + w_2x_2 + ... + w_nx_n \tag{2.1}$$

$$y' = \frac{1}{1+e^{-z}} \tag{2.2}$$

FIGURE 2.2: Logistic Regression (Google, 2020)

z represents the output of the LR, with b the bias, w a weight, x a feature and n the number of features.
y' represents a probability, between 0 and 1. The loss function, log loss, is then used to improve the weights and biases at each iteration.

This model, although simple, can predict the direction of the Bitcoin price. However, one must be careful about the choice of features. The risk is that if the correlation coefficient between features is too high, this will cause complete or near-complete separation and render any training of the model useless. (Dormann et al., 2013) specifies that it is customary for the correlation coefficient between features to be less than 0.70, but in other cases such as in (Elith et al., 2006) it has been shown that 0.85 can be accepted.

2.1.3.2 Support vector machine

Support vector machines (SVM) are used to make classifications. (Noble, 2006) explains that in order to understand how an SVM works, one must first understand the following notions, the separating hyperplane, the maximum-margin hyperplane, the soft margin and the kernel function.

The separating hyperplane, visible in figure 2.3 b, is an element that will separate two classes, whatever the number of features (their number corresponds to the number of dimensions).

The maximum-margin hyperplane corresponds to the way the SVM will determine the position of its hyperplane. It may be that several positions are correct, as in figure 2.3 c, and perform an identical classification. However the maximum-margin hyperplane will maximise the margin between the elements to be classified as shown in figure 2.3 d.

In the case where data cannot be separated by a line (figure 2.3 e), the soft-margin allows you to concede a certain amount of data that is misplaced and thus reposition the hyperplane figure 2.3 f.

And to conclude, the notion of kernel function. Let us take the example of figure 2.3 g which is not separable by a hyperplane, here a point. To solve this problem, figure 2.3 h shows that by adding a kernel, an extra dimension is added to make the dataset classifiable by the SVM.

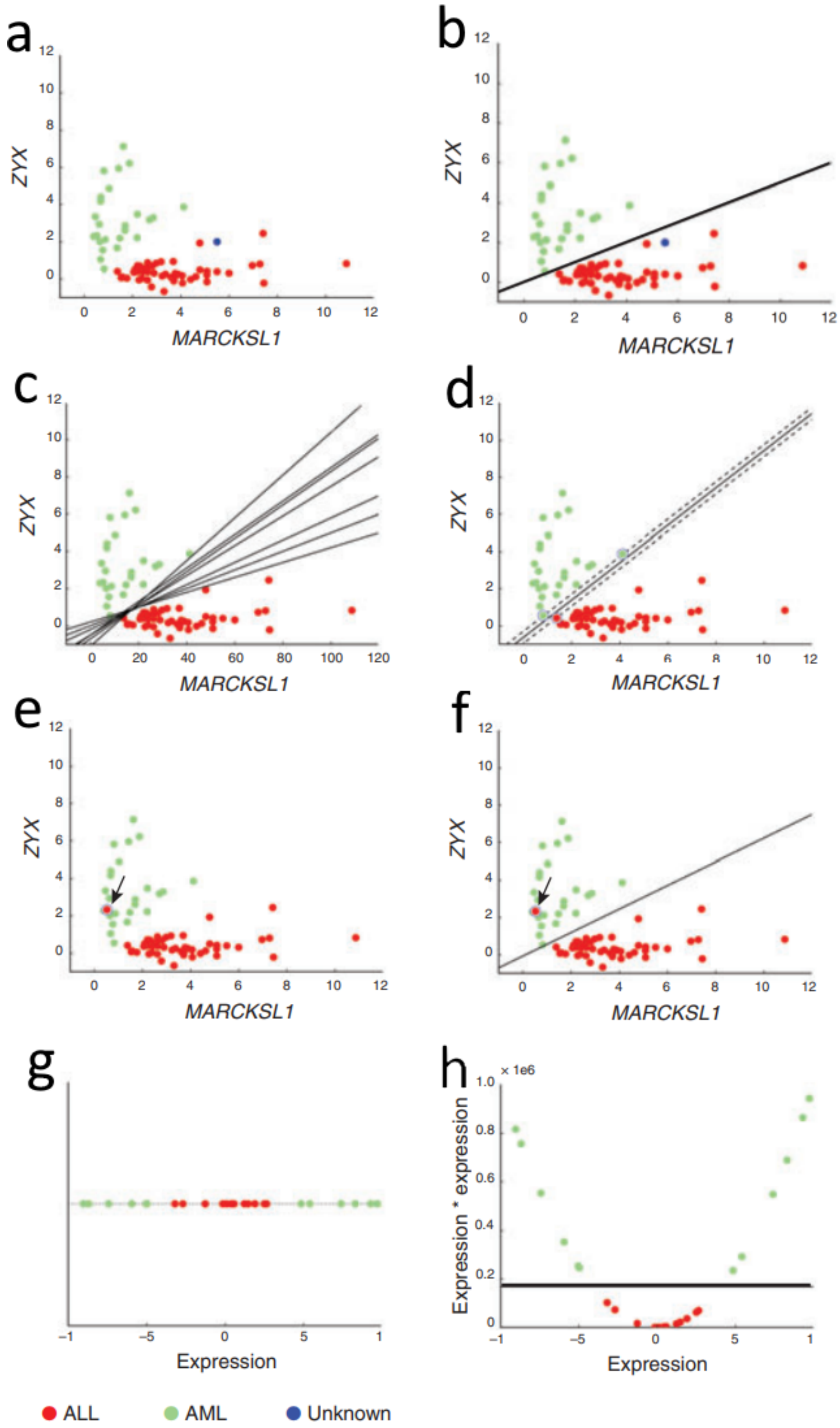

FIGURE 2.3: SVMs at work (Noble, 2006)

In the next section 2.1.3.3 we will discuss a neural network model which, as (Sharma et al., 2018) shows, is almost exclusively used for image classification.

2.1.3.3 Convolutional neural network

As described in (Acharya et al., 2017), convolutional neural networks can be considered as a multilayer perceptron (MLP). However, CNNs are distinguished from classical MLP models by their singular architecture composed of specific layers such as convolutional layers, pooling layers and generally fully-connected layers as output.

One can see on the figure 2.4 the application of a kernel which corresponds to a matrix of weights, to carry out a convolution on an input. Kernels detect shapes and angles, which is very useful in image analysis. This convolution makes it possible to capture an information by taking into account the adjacent information.

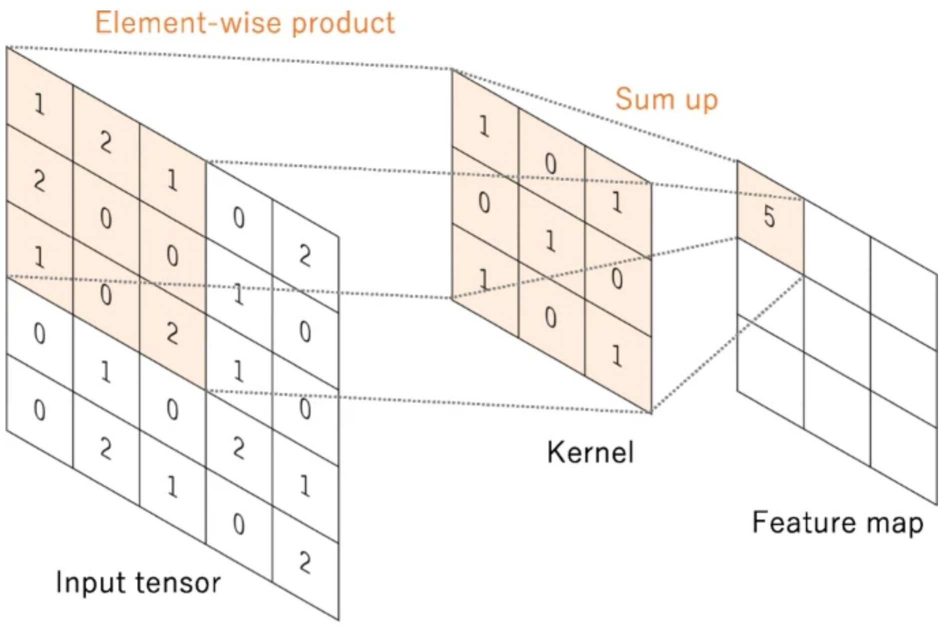

FIGURE 2.4: CNN: Convolution, kernel size 3x3 (Yamashita et al., 2018)

2.1.3.4 Long short-term memory

Recurrent neural networks (RNNs) are subject to the vanishing gradient problem. This phenomenon is described by (Brownlee, 2019) as the progressive decrease of the gradient leading to its disappearance. The consequence of this is the inability of the neural network to perform a gradient descent from its output layers to its first layers at the input level.

The concept of long short-term memory (LSTM) comes from (Hochreiter and Jurgen, 1997). They explain that they wanted to solve the problem of what they call the "decaying error backflow" present in RNN by proposing an innovative architecture (figure 2.5, their version had no forget cell). The logic of the LSTM is based on the use of several gates to learn and unlearn in order to regulate the information to be retained during training, this allows to avoid a vanishing gradient during the training. There is also the cell state, which compensates for the role of the gates by systematically retaining a portion of the information from the previous cells.

(Dolphin, 2021) describes the LSTM as a series of gates acting as filters. The LSTM is composed in order of a forget gate, an input gate and an output gate, each corresponding to a neural network. Figure 2.5 shows a cell of an LSTM with three stages, as well as the data flow and the activation functions of the different gates.

The first step, the forget gate (figure 2.5) which consists in retrieving the information of the previous cell, the previous cell state $c(t-1)$ and the previous hidden state $h(t-1)$) as well as a new data input $x(t)$ in order to determine the cell state of the cell. The previous hidden state $h(t-1)$ and the data input $x(t)$ will both be passed to the first neural node represented by the sigmoid function, whose output between 0 and 1 will determine whether or not it is safe to retain the information. It will then multiply $f(t)$ with the previous hidden state $h(t-1)$ which will result in forgetting some of the information from the previous steps if $f(t)$ is zero, hence the name forget gate.

The second step, the input gate (figure 2.5) takes as input the result of the matrix product of the forget gate, as well as the previous hidden state $h(t-1)$ and the data input $x(t)$ (like the forget gate). These two inputs $h(t-1)$ and $x(t)$ will be passed through a neural node with a tanh function that will aim to produce an output between -1 and 1. This will have two consequences. Firstly, to avoid the vanishing gradient problem as it can be the case with a sigmoid function. Secondly, to allow an increase or a decrease of the cell state thanks to the tanh function output [-1, 1]. It is not possible in the case of a sigmoid function output close to 0 because it would not modify the cell state during the addition of the matrix. As in the first step, in order to avoid conserving unnecessary information, we find a neural node with a sigmoid whose result is multiplied with the matrix output of tanh function before being added to the cell state matrix from the forget state.

The last step, the output gate (figure 2.5) will take the cell state and pass it into a tanh function. At the same time, the two initial inputs $h(t-1)$ and $x(t)$ will be passed back into a neural node with a sigmoid function and the matrix product of these two results will give the hidden state for the next cell.

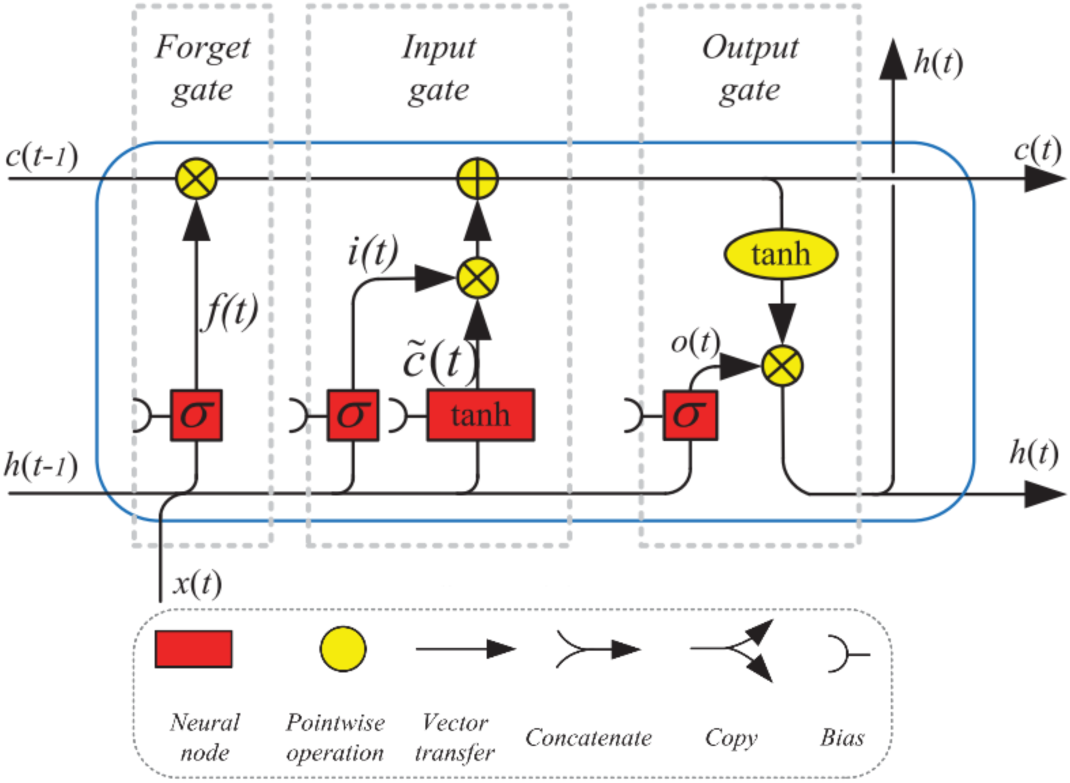

FIGURE 2.5: Architecture of LSTM with a forget gate (Yu et al., 2019)

It should be noted that this process is repeated as many times as necessary. If the training is done over 50 days, then this process will iterate 50 times and each iteration will train 5 neural networks, which is time consuming.

In the next section 2.1.3.5, we will discuss the last type of model addressed in this research, the Bayesian neural network (BNN), which like the LSTM is a special case of the MLP family.

2.1.3.5 Bayesian neural network

(Jang and Lee, 2018) described Bayesian neural network (BNN) as part of the MLP family. Their use is possible in several domains such as natural language processing (NLP) or image recognition but they would be even more relevant in applications with complex time series. In their system, the backpropagation is used to minimize the sum of the errors as shown in the equation 2.3.

$$E_B = \frac{\alpha}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} (t_{nk} - o_{nk})^2 + \frac{\beta}{2} \precapprox_B^T \precapprox_B \tag{2.3}$$

FIGURE 2.6: BNN: Sum of the errors (Jang and Lee, 2018)

What differentiates BNN from MLP is the logic with which it updates the weights of its network. Unlike MLP that rely on marginal likelihood estimation (MLE), the BNN is based on the application of the Bayes' theory and use maximum a posterior (MAP) to improve weights and biases. (Yang, 2019) explains that the advantage of MAP over MLE is its ability to consider the probability of a hypothesis and thus compensate for the fact of having a dataset not satisfying the law of large numbers and not having a uniform distribution.

After discussing the fundamental theoretical basis of this project, we will see in section 2.2 the state-of-the-art of machine learning models for predicting the Bitcoin price.

2.2 Related Work

2.2.1 Indicator

In order to proceed with the training of machine learning models, it is necessary to determine with which type of data it is relevant to train it. The two main categories of analysis methodologies used to take positions on stock markets, technical analysis and fundamental analysis, are opposed by (Gyan, 2015) and (Sankar et al., 2015).

The fundamental analysis according to (Segal and Boyle, 2021) is a process allowing the evaluation of the stock exchange value which is based on the analysis of factors from a macroeconomic and microeconomic point of view of factors in relation to the object to be analysed. Numerous studies (Kaminski, 2016), (Georgoula et al., 2015), (Mittal et al., 2019) and (Matta et al., 2015) have shown a certain degree of correlation between the use of services such as Twitter, Google or Wikipedia and the price of Bitcoin. This way of analysing the market is interesting but the current research raises many questions in terms of bias. For example, an individual can publish several tweets or make several requests on a search engine per day using many devices and accounts, shouldn't this limit the weight of his actions? The study (Ante, 2021) reinforces this notion of weight, but here rather than seeking to reduce the impact that a user has, it would be necessary in certain cases to increase its weight. Indeed, the behaviour of influential personalities on the internet has a much wider impact than an ordinary user. To improve this process, it would be necessary to limit the collection of data to the most influential sources of information, which is beyond the scope of this study.

Because of the bias problems seen earlier that fundamental analysis raises, we will only use indicators that fall under the category of technical analysis. (Hayes, 2020) defines technical analysis as the analysis of historical fundamentals to evaluate the security's value. The term fundamentals corresponds to quantitative data such as open, close, min, max or traded volumes.

According to the conclusions of (Detzel et al., 2021), the use of price-to-moving average ratios coupled with simple real-time strategies allows for better results than the buy-and-hold strategy which consists of buying and holding a position without worrying about market fluctuations. While methods based on technical analysis can generate profits by holding positions over shorter time intervals.

In order to determine the factors influencing the price of Bitcoin, (Vassiliadis, 2017) was able to establish correlations with financial factors as well as other factors such as gold and crude oil. A strong correlation over three years between the price of gold and crude oil with Bitcoin has been demonstrated. The same finding has been made with economic indices such as NASDAQ, DAX and SIP500, however with a more moderate correlation. These factors will prove very useful during our study.

This same correlation was confirmed in the work of (Mallqui and Fernandes, 2019) who included no less than 86 of these financial factors in his datasets. After applying five selection methods using the Weka software, correlation analysis (Corr), correlation-based feature subset selection (CFS), relief technique (Relief), information gain method (InfoGain) and principal component analysis (PCA), they were able to retain around twenty features of interest visible in the table 2.1. It should be noted for our project that Corr obtained the most convincing results, but in the case where the time interval was the largest, it was preferable not to use any selection method.

After having detailed the values and indexes of interest, we will see in section 2.2.2 the state-of-art of machine learning models to make predictions on the Bitcoin price rate.

Day D	**Day (D - i)**	**30-day WMA**
Opening price	Price direction	Opening price
Timestamp	Opening price	Maximum price
	Maximum price	Minimum price
	Minimum price	Closing price
	Closing price	Volume of trades
	Volume of trades	Number of txn
	Number of txn	Transaction fees
	Transaction fees	Cost per txn
	Cost per txn	Hash rate avg
	Cost per txn	Closing crude price
		Closing gold price
		Closing S&P500 price
		Closing Nasdaq price
		Closing DAX price

TABLE 2.1: List of possible input attributes (Mallqui and Fernandes, 2019)

2.2.2 Machine learning model for Bitcoin price prediction

In this section, we will review the results of research proposing machine learning models and see how this relates to our project to predict the evolution of the bitcoin price.

The LSTM studied by (McNally et al., 2018) and (Alessandretti et al., 2018) have correct performances with an accuracy close to 52%, However, they seem to have been outperformed by other models such as 1D-CNN or BNN as shown by (Mern et al., 2017) and (Cavalli and Amoretti, 2021). Nevertheless, the architecture of the LSTM is supposed to give it an advantage over the time series forecasting.

Moreover, (Yu et al., 2019) evokes many variations of LSTM since the first version of (Hochreiter and Jurgen, 1997), such as multidimensional LSTM network, convolutional LSTM network or LSTM-in-LSTM network, which still leaves many research axes to explore.
In (McNally et al., 2018), the limits of the RNN have been proven, showing that it barely exceeds 50% accuracy and loses all interest on time intervals longer than 50 days. Whereas the LSTM model has a better accuracy (+2%) and can be used over larger time intervals, 100 days. Therefore, we will not use RNN in our project, but only LSTM and will take advantage of their characteristics to test them over large time intervals.

Other models such as SVM and a LR method were tested in this (Manoharan et al., 2020) to predict the price of another cryptocurrency, Ether which is the cryptocurrency of the Ethereum network. The accuracy of these two models exceeds 80%, and is close to 99% for the SVM with all features. Although the cryptocurrency studied is not the same, the results are very high and these models deserve to be tested with our research on Bitcoin.

Next we will look at the two most promising models, the BNN and then the CNN. First, the BNN, there is little research on it compared to LSTM or CNN, however their results seem very promising as shown by (Jang and Lee, 2018) whose BNN has a mean absolute percentage error (MAPE) of 1%. Secondly, the CNN which was popularised in 2012 for image classification thanks to the very good results of the CNN (figure 2.7), used in (Krizhevsky et al., 2017) which won the "ImageNet Large Scale Visual Recognition Competition" the same year.

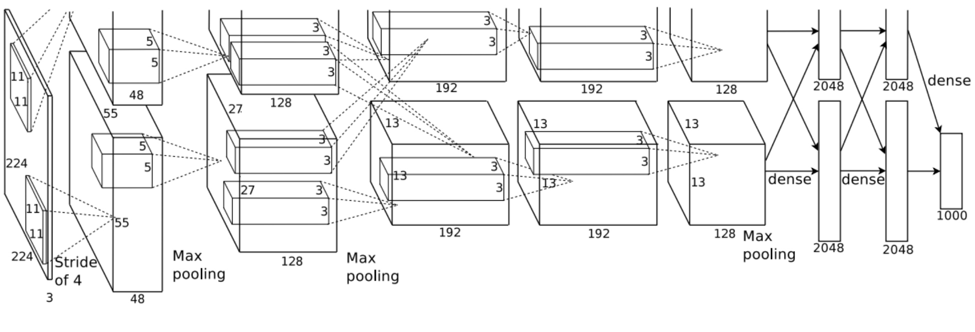

FIGURE 2.7: Architecture du CNN AlexNet (Krizhevsky et al., 2017)

In 2017, (Mern et al., 2017) diverted the primary use of CNN from image classification to prediction on time series data. Their model has achieved an accuracy of 59.7%. In 2020, (Cavalli and Amoretti, 2021) has proposed an improvement with a 1D-CNN where data and kernels have only one dimension. During their tests, their model, which is based on the analysis of seventeen features, achieved 74.2% accuracy. The same model was used in a real-life situation for 248 days, during which it predicted the Bitcoin price evolution with an accuracy of 72.9%, a value in line with their previous test. In view of these results, it can be said that this model (CNN) represents the state-of-art. Special attention will be paid to the realisation of the CNN during our study.

A number of research studies proposing a new machine learning model related to Bitcoin such as (Cavalli and Amoretti, 2021),(Huang et al., 2019), (Shah and Zhang, 2014) and (Alessandretti et al., 2018) are trying to prove the relevance of their results by applying their model to real situations to show potential financial benefits. However, like a simple moving average or any other financial indicator, the up and down prediction of a machine learning model remains an indicator and cannot be used alone. Determining when to enter and exit the market is a trading strategy beyond the scope of this research.

To evaluate and compare models, relying solely on accuracy is insufficient. This is why (McNally et al., 2018) and (Mallqui and Fernandes, 2019) rely on the use of performance metrics. On the one hand, regression performance metrics such as MAE, MAPE and RMSE are used to evaluate their models, and on the other hand, sensitivity, specificity and accuracy (described in chapter 3) are used to evaluate the ability to classify the Bitcoin price evolution. Not relying solely on accuracy and using other statistical indicators is a good practice that we will apply in our study.

In conclusion, (Nti et al., 2020) conducted a critical review of 122 research papers in 2019 dealing with Bitcoin price prediction models. Although not exhaustive, this research found that 87% of these papers claimed that their model was better than those they were benchmarked against. Although in this project we did not intended to provide such a broad review, we did find this phenomenon in the background research. This justifies the interest in comparing the most promising machine learning methods by putting them on an equal footing for comparison with a similar dataset, and applying evaluation methods based on the same performance metrics. We will discuss these processes in Chapter 3.

Chapter 3

Requirements Analysis

According to (Salem and Darter, 2006) software projects fail in part because of incomplete, ambiguous and inconsistent requirements. This section aims to verbalise the requirements necessary for the successful completion of the project.

The requirements of this project are categorised as follows:

- Functional requirements

  These requirements define the functions of the project that are mandatory. The definition of one of these requirements can be reduced to the inputs, behaviour and outputs of the requirement.

- Non-Functional Requirements

  Unlike functional requirements, non-functional requirements are not necessarily mandatory. Non-functional requirements describe the characteristics of the system and are intended to make the project usable and efficient. It would be useless to propose a machine learning model that when the user gives it an input, makes a prediction after several hours. The model should be able to perform its task in a few seconds at most.

- Business Requirements

  These requirements correspond to the functionalities that the end user should be able to use when the project is finished.

We will start with business requirements, then functional requirements and finally non-functional requirements.

3.1 Business Requirements

The program must be able to maintain its database up to date on a daily basis and determine with an accuracy of more than 50% the evolution of the Bitcoin price for the next day.

3.2 Functional Requirements

In a first step, data will be collected from different sources such as Coindesk, Quandl, LunarCrush and the Bitcoin blockchain. Then a database will be created with financial indicators such as crude oil price, gold price which have been proven to correlate with the Bitcoin price in (Vassiliadis, 2017). Then we will add the data related to the Bitcoin price that can be found in the blockchain such as the opening, closing, minimum and maximum value. (Benediktsson, 2021) which is a Python wrapper of the TA-lib library can be used to make calculations generating effective indicators in technical analysis, such as SMA, EMA, MACD and RSI. (TradingView, 2021) also makes these financial indicators available to its users.

In a second step, we will proceed with the data preprocessing:

- Verification of the validity of the data (no erroneous or missing values, standardisation and consistency)
- Normalisation
- Selection of the most relevant indicators (features)
- Separation of the dataset in two, 70% dedicated to training and 30% dedicated to testing the model generated after training

Several iterations of this process will be required to determine the most appropriate feature selection method.

Once the dataset generation and preprocessing steps have been completed, a script will be set up and deployed to automate these two processes to directly generate files for training the specified model.

In order to compare machine learning models and select the most efficient one, the following performance metrics will be used ROC curve (AUC) and accuracy (equation 3.3), which is calculated from the sensitivity (equation 3.1) and the specificity (equation 3.2). Not forgetting F1-score (equation 3.5), which allows better analysis than accuracy when the distribution is heterogeneous.

$$TPR = \frac{TP}{TP + FN} \tag{3.1}$$

$$TNR = \frac{TN}{TN + FP} \tag{3.2}$$

$$TPR = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.3}$$

FIGURE 3.1: Sensitivity 3.1, Specificity 3.2 and Accuracy 3.3 (Baratloo et al., 2015)

$$PPV = \frac{TP}{TP + FP} \tag{3.4}$$

$$F1 - score = \frac{2 \cdot PPV \cdot TPR}{PPV + TPR} \tag{3.5}$$

FIGURE 3.2: Precision 3.4, F1-score 3.5 (Goutte and Gaussier, 2005)

$$Micro{-}averaged\ precision = \frac{TP1 + TP2}{TP1 + FP1 + TP2 + FP2} = \frac{TP1 + TP2}{PP1 + PP2} \tag{3.6}$$

$$Micro - averaged\ recall = \frac{TP1 + TP2}{TP1 + FN1 + TP2 + FN2} = \frac{TP1 + TP2}{P1 + P2} \tag{3.7}$$

FIGURE 3.3: Micro-averaged precision 3.6, Micro-averaged recall 3.7 (Frank, 2019)

According to (Ekelund, 2011), the AUC shows on a graph the relationship between sensitivity and specificity, with on the x-axis $(1 - specificity)$ and on the y-axis $sensitivity$. The sensitivity corresponds to the proportion of cases correctly identified as positive compared to the total number of positive cases. The specificity corresponds to the proportion of cases correctly identified as negative compared to the total number of negative cases.

The figure 3.4 summarises the workflow process of the project once the functional requirements have been completed.

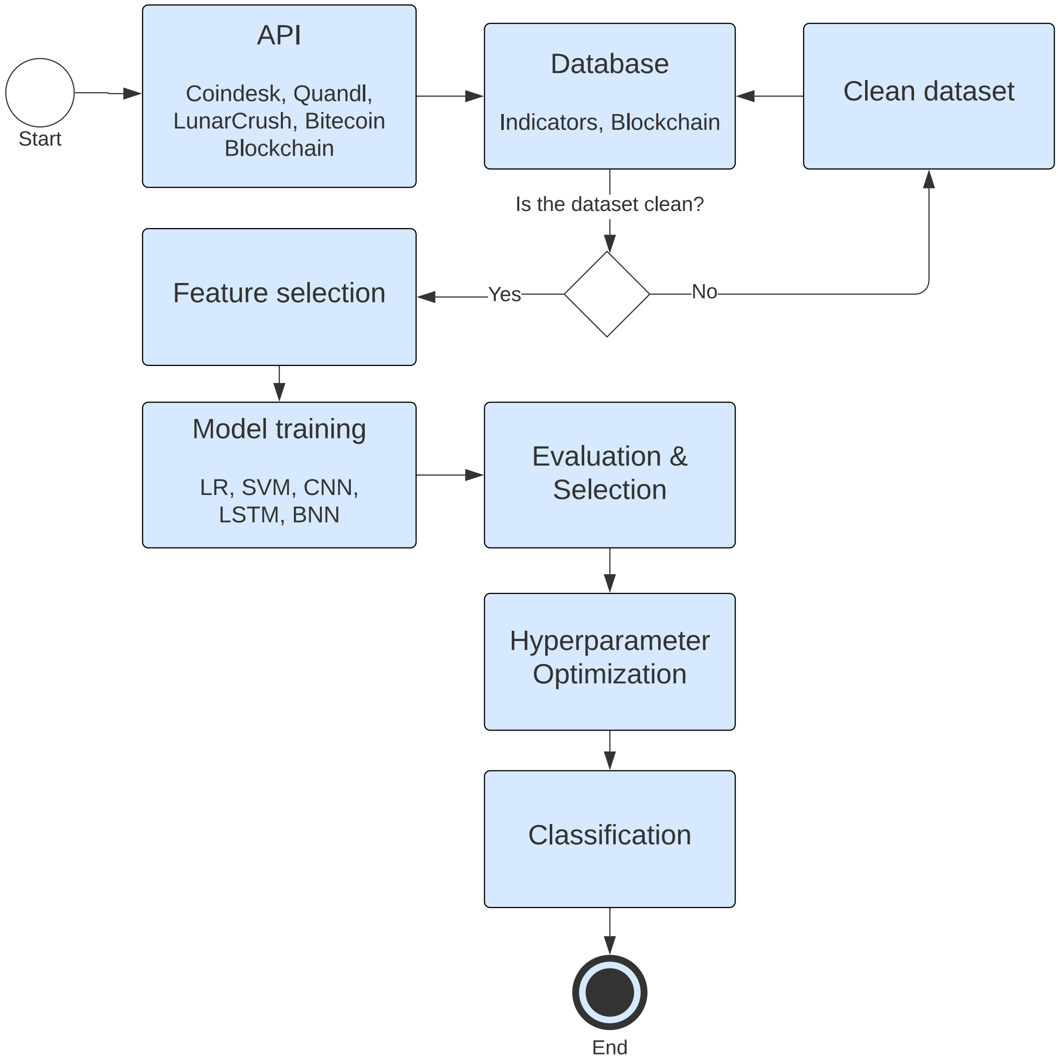

FIGURE 3.4: Project Workflow (BPMN 2.0)

3.3 Non-Functional Requirements

As this project is not intended to be used to generate profits by automating investments, the model does not need to generate predictions over short time intervals. The model should be able to make a prediction in a few seconds is sufficient.

In order to make it easy-of-use, maintainable and reusable a documentation will be generated using the Sphinx framework adapted to the Python language.

The project must be easily deployable and usable on several platforms, to do this it will be developed as a python library.

Now that the project requirements have been defined, the next chapter 4 will describe the methodology for carrying out the various tasks.

Chapter 4

Methodology

In this section we will describe the methodology used during this research project. In order we will look at the working methodology, the software, then the frameworks and finally the dataset.

The methodology used in this research will take elements from the agile method described in (Turk et al., 2005). There will be no scrum master or product owner but the sprint logic will be kept as well as the tools managing the product backlog. Trello will allow the project to be broken down into tasks and to have a better follow-up of the project during its development.

The overleaf web service will facilitate the redaction of documents in LaTeX, while ensuring a versioning of the latter. To increase productivity, the Zotero software and its web browser extension will help us to collect the metadata directly on the web page containing the articles and other types of sources to be referenced. The bibliography will be generated in accordance with the "Cite Them Right 10th edition - Harvard" language style.

For the development of scripts related to dataset generation, dataset preprocessing, creation of machine learning models, genetic algorithms and model performance evaluation, the Python language will be used. However, if Python proves to be a bottleneck in terms of performance, C++ will replace it on critical parts of the code. Depending on the context, one of the following frameworks will be selected: TensorFlow 2.0, Keras (high-level API in TensorFlow), scikit-learn and Pytorch.

Chapter 5

Professional, Legal, Ethical and Social Issues

5.1 Professional issues

The written code will follow the recommendations of the British Computing Society codes of conduct.

In order to create a perennial project that can be shared, understood and improved, the code will be documented and open-source.

In order to maintain quality and prevent regression during development, the project code will be versioned.

5.2 Legal issues

The integrity of no human being will be endangered, as this project does not involve any human user.

The data will be collected in accordance with the conditions of use set by the organisations providing access to the data.

The project will be published under the GNU Affero General Public License (GNU, 2007).

The author can be contacted at jmmc2000@hw.ac.uk.

5.3 Ethical issues

There are no ethical issues with this project.

- No user intervention will be necessary for the realisation of the project.
- Intellectual property will be respected. Sources, figures and programs used will be referenced using Zotero and the language style "Cite Them Right 10th edition - Harvard".
- There will be no confidentiality issues with the dataset because it will not contain any sensitive or personal data that could identify an individual.
- This project is not intended to encourage investment in Bitcoin or any other cryptocurrency, but to better understand the behaviour of this cryptocurrency with machine learning models.

- This project is not intended to make money by investing in a cryptocurrency. Indeed, the project aims to predict a rise or fall in the Bitcoin price, not when to enter or exit the market to make a profit.
- I have no personal or financial interests that would affect my research related to cryptocurrencies.
- The entire code and dataset will be available in an open-source project.

5.4 Social issues

This project is related to a cryptocurrency, so it does not deal with socially sensitive issues.

Chapter 6

Implementation

As explained in chapter 4 and in section 3.2, we will mainly use Python 3.8 and Weka 3 (Hall et al., 2009) for the implementation of the project. The use of Python was not a bottleneck and the use of C was not necessary for the realisation of the project. As mentioned in (Pandas, 2021), it turns out that the Python libraries that require good efficiency, such as Pandas, are themselves Python interfaces using C or Cython. We have used Weka, with its graphical interface developed in Java for data preprocessing and increases the memory allocation of its heap to 12GB.

We have developed the "Research-ML-BTC" project (Chartois, 2021) in Python and implemented it with a library logic. Each module is contained in a folder, a `setup.py` file assists in building the library. The advantage of having a library is that it facilitates the sharing of the project content by simplifying the installation at the user's. The documentation is generated through Sphinx thanks to the `conf.py` file.

We will see in the following sections the implementation of these different modules:

- Fetch, to automate the retrieval of data from different sources via APIs and other data provision services.
- Chart, to highlight the data used through charts generated via the plotly framework.
- Convert, to generate arff format datasets for Weka from csv format files.
- Model, to create the machine learning models.

6.1 Dataset

Bitcoin CME Futures (BTC1!-CME) are a popular derivative product for institutional investors and are more recent than Bitcoin. In order to take the BTC1!-CME into consideration, we have set up two datasets. One dataset contains information about the BTC1!-CME, with a time frame starting from 17/12/2017 until 15/07/2021. And a second dataset excluding BTC1!-CME, for the time interval from 03/06/2012 to 15/07/2021.

The creation of the dataset has been automated in order to facilitate access to the data, however some of the data has been retrieved manually as detailed in section 6.1.1 which goes against a total automation of the dataset creation process as discussed in the section 3.1.

In order to generate the datasets, it is enough to instantiate the class `Dataset` then to call the method `create_dataset`. The data is retrieved by the class `Dataset` contained in the module `./source/fetch/fetch.py` and then saved in two files in csv format in the folder `./source/fetch/data/`.

We will look in detail at the process that was required to find the data sources.

6.1.1 Data sources

In order to build the dataset, several iterations were necessary to find qualitative sources to constitute our dataset. These iterations will be described below.

The first criterion for selecting data sources was to propose historical data covering several cycles of the Bitcoin starting between 2010 and 2013.

In the first instance, we turned to (CoinDesk, 2017) for the following reasons.
First, their API provides historical Bitcoin data going back to July 2010, free of charge and in the form of the Bitcoin Price Index (XBP). As explained in their documentation, the XBP represents an average of the Bitcoin price taking into account data from the largest cryptocurrency exchanges, described as intended to serve as a reference for industry and professionals.
The first problem encountered was the lack of consistency between the information provided by their API and the data made available via their chart (CoinDesk, 2021*a*). Indeed, the json that is returned by the API only gives the daily closing values of Bitcoin from 2010 onwards whereas the data extracted from their chart shows daily OHCL data since 2013.
Furthermore, little information is provided in their documentation as it is described as a deprecated public API. Coindesk explains that they have a private API but does not communicate on how to access it.

Secondly, we found that the Lunarcrush service does not allow access to historical Bitcoin data in its free version. After contacting their customer service, they provide a paid API with two years of historical data which is not enough for our project.

Thirdly, we implemented in our project the solution proposed by (Garcia, 2021). This solution announces the possibility to retrieve data close to the Bitcoin Reference Rate (BRR) published by the Chicago Mercantile Exchange (CME). Indeed (Garcia, 2021) and (CME, 2021) gather data from the following exchanges: Bitstamp, Coinbase, Kraken, itBit, with one difference, Bitfinex for the former and Gemini for the latter.

However, when generating an OHLC chart with the transaction volumes, it turned out that the dataset had outliers for the transaction volumes. This is because the API is provided by (Bitcoincharts, 2021). The latter's documentation mentions not to use this data for trading bots as they do not certify the accuracy of the data.
In addition, the initial dataset is made up of unit transactions and when aggregating into a daily OHLCV dataset, Garcia's solution averages the observed values (Open, High, Low, Close) without weighting the result according to the volume done on each exchange, what affects data quality.
This solution has not proved reliable for two reasons, the initial quality of the data and the method of data aggregation.

Then, we turned to another source, (Investing.com, 2021) which was used in (Vassiliadis, 2017) to retrieve gold price and crude oil data. Investing.com does not offer an API, however the python package of (Bartolome del Canto, 2021) allowed us to create a dataset with historical OHLCV Bitcoin's data, crude oil and gold dating from 2010 to 2021.
While performing data mining, it turned out that the data lacked precision. Only two decimal places were present for the OHLCV values of Bitcoin. Indeed, its value during its first years of existence was relatively low, which was characterized by a value close to or equal to 0 over long periods in their historical data. In addition, using the Bitcoin API caused a one-day shift in the middle of the dataset which was visible when compared to the data available on (Investing.com, 2021). Despite this, the Python investing package remains our reference for crude oil and gold.

And finally, the solution we used for Bitcoin (TradingView, 2021). TradingView aggregates all the BTC/USD pairs from the major exchanges, plus it also offers the BraveNewCoin Liquid Index for Bitcoin (BNC:BLX). (BraveNewCoin, 2015) explains that the BNC:BLX represents a fair version of the Bitcoin price as it is based on BTC/USD transactions from the following exchanges: Bitfinex, Bitstamp, Coinbase Pro, Gemini, itBit, Kraken. We use the latter to get a representative Bitcoin price.

This service is one of the most visited sites in the world today, as shown in the (SimilarWeb, 2021) which ranks it 1st in the Global Rank of the Finance section ahead of Investing.com, CoinMarketCap and Binance. It is used by Binance, the largest exchange by volume (de Best, 2021*a*), to help its users buy cryptocurrencies, as seen in (Binance, 2021). Exporting data from their chart is possible, but they do not have an API to automate this process. As mentioned in section 6.1, it hinders the automation of the dataset creation process.

We also had to collect information about the Bitcoin blockchain, the composition of which is described in section 6.1.2.

(Blockchain.com, 2021) provides an API to retrieve access to this data. However, after implementing this API in our project, it is not in accordance with the documentation provided. Indeed, it does not return the entirety of the daily data over the time interval from 2012 to 2021, but only one day out of three of data, the others being missing. In order to remedy this, we used (Quandl, 2021*a*) which has an API that makes available data retrieved from Blockchain.com but without missing day.

The next section 6.1.2 will describe in detail the data types that constitute the dataset.

6.1.2 Composition of the dataset

Here is the list of data collected with their description and name used in the ranking correlation table:

1. Open, High, Low, Close and Volume (OHLCV) with the Date.
   This corresponds to the open, high, low and close values of Bitcoin over a given time interval, in this case 24 hours. As well as the volume traded over the same interval. These values are commonly used to track the price of stocks or cryptocurrencies via candlestick charts.

2. Information about the Bitcoin blockchain. The data comes from Quandl as discussed in the previous section 6.1.1 :

    - Difficulty (Value-DIFF), allows you to determine the difficulty required to mine a block.
    - My Wallet number of transaction per day (Value-MWNTD), number of transactions made by My Wallet users per day on blockchain.com.
    - My Wallet volume (Value-MWTRV), corresponds to the volume of transactions made on blockchain.com in one day.
    - Miners revenue (Value-MIREV), is the total revenue of the miners including the bitcoin block reward as well as the fees paid to them to validate the transactions in USD.
    - Hash rate (Value-HRATE), which corresponds to the number of SHA-256 operations (hash algorithm) per second performed over 24 hours over the Bitcoin blockchain. The unit used is TH/s (terahash per second).
    - Blockchain size (Value-BLCHS), to measure the size of the blockchain, including headers and transactions but excluding database indexes, in megabytes (MB).
    - Average block size (Value-AVBLS), for the average size of a block in MB.
    - Bitcoin days destroyed minimum age one year (Value-BCDDY), this is a measure of the transaction volume of Bitcoin spent (destroyed) that are older than one year.
    - Median transaction confirmation time (Value-ATRCT), the median time it takes to validate a transaction, for a block to be mined and added to the blockchain in minutes.
    - Total Bitcoin (Value-TOTBTC), total number of Bitcoins mined and in circulation on the blockchain.
    - Bitcoin market capitalisation (Value-MKTCP), is the total value of Bitcoins in circulation in dollars.

3. A derivative product, the Bitcoin Chicago Mercantile Exchange (CME) Futures (BTC1!CME).

   According to (Hussey, 2021), more than half of crypto hedge funds use derivatives. These professionals seek to speculate on the future price of Bitcoin, hence the need to use a derivative. Currently (Tiwari, 2021) points to the fact that derivatives on traditional markets are several times larger than the spot market. This is not yet the case for the still immature cryptocurrency market, which was worth around $1 trillion by June 2021, compared to $15.5 trillion for the gross market value of over-the-counter (OTC) derivatives according to (de Best, 2021*b*) and (Hodgson, 2020).

4. (Huang et al., 2019) has demonstrated the value of financial indicators for predicting the price of Bitcoin. Here are the indicators we will use:

   - Moving average (MA) 50 day.
   - Exponential moving average (EMA) 9 day, which corresponds to a weighted MA. The EMA will give more weight to recent data.
   - Volume EMA.
   - Volume MA.
   - Bollinger Bands based on daily close values, a 20 day MA (Basis) and a standard deviation of 2 for the lower (Lower) and upper (Upper) limit.
   - Relative Strength Index (RSI), is an indicator that can take values between 0 and 100. A value above 70 indicates an overbought stock while a value below 30 indicates an oversold stock.

5. Daily closing value of gold (Close-xau) and crude oil (Close-crude oil) prices.

6.1.3 Fetch

Fetch is the name of the Python module used for fetching in our project. The data is retrieved by the class `Dataset` contained in the module `./source/fetch/fetch.py`. The method `create_dataset` will call the following four methods:

- `fetch_btc_data`, for OHLCV data (figure 6.1) and financial indicators.
- `fetch_blockchain_btc_data`, for data related to blockchain.
- `fetch_data_by_symbol`, for WTI crude oil futures (ICE exchange) and AUX/USD gold (NASDAQ exchange) data.
  Unlike the cryptocurrency market, traditional markets have opening and closing hours and are closed on weekends. To overcome this problem, we have decided to fill in the missing days with the data from the previous day.
- `fetch_future_btc_data`, for BTC1!CME.

We use the dataframes provided by the Pandas library to dynamically store and modify the dataset. Indeed (Mckinney, 2011) confirms the interest of using this framework in order to manipulate structured data related to domains like statistics or finance, which meets the criteria of our study.

We concatenate the first three dataframes, check their integrity with the absence of null values and the right data type. Then we create the first dataset in csv format with a header in `source/fetch/data/dataset.csv`, the data starting at 2012-06-03.
Following the same procedure a second dataset is created by removing the dates prior to 2017-12-17, then concatenating the dataframe returned by `fetch_future_btc_data`.
This csv is stored in `source/fetch/data/dataset_with_future.csv`.

Each daily instance of the dataset is labelled. If the previous day's closed value is lower than the current day's closed value, then label "True" is used, the Bitcoin price has increased. Otherwise the label False is applied to the instance.

6.2 Chart

Chart is the name of the Python module used to create a candlestick chart displaying OHLC data and trading volumes since 2012 with (Plotly, 2015). We need to instantiate the class `Candlesticks` which is in the module: `source/chart/candlesticks.py` and call the method `display_candlesticks_chart` and then give it the dataframe containing the OHLCV data as parameter.

Once the script is executed, it is possible from your browser to select a specific time interval as shown in figure 6.1. The green candles represent days where the open value is lower than the close value, and vice versa for the red colour.

FIGURE 6.1: Sample OHLCV data from the 2012-2021 dataset showing the evolution of the bitcoin price from the start of the last bull run in October 2020 to July 2021.

In the next part we will see the results of the preprocessing of the attributes described in section 6.1.2.

6.3 Data preprocessing

Before proceeding to the selection of the attributes we have converted the date into a number, their value being the number of milliseconds from 01/01/1970 00:00:00 GMT to the corresponding date and we have normalized the dataset.

(Mallqui and Fernandes, 2019) showed that the use of correlation analysis in Weka 3.8 was the most relevant method of attribute selection. However, the absence of selection and the retention of all attributes seemed beneficial for large time intervals.

We will therefore apply their method, using correlation analysis (Corr) on the dataset with the shortest time interval to select the twenty most relevant attributes. We will integrate by default the BTC1!-CME in addition to the twenty most relevant attributes. And we will not apply a filter on the largest dataset, which starts 2012 without BTC1!-CME.

We note that the CFS Subset and Gain Ratio feature evaluator selection methods seem to confirm the importance of the RSI attribute by putting it at the top of the list of attributes to be selected. These results are in line with those of the correlation analysis below.

Correlation Ranking Filter		
Rank	Attribute	Correlation value
1	RSI	0.251467
2	Value-MWNTD	0.088124
3	Volume	0.063125
4	Close-crude-oil	0.057615
5	Value-AVBLS	0.050588
6	Value-TOTBC	0.042735
7	Date	0.039839
8	Value-ATRCT	0.039389
9	Close-xau	0.038814
10	Value-BLCHS	0.037684
11	Volume-MA	0.030133
12	Value-HRATE	0.028043
13	Value-DIFF	0.025996
14	MA	0.021085
15	Open	0.020425
16	Value-MIREV	0.018423
17	Basis	0.017365
18	Upper	0.017351
19	Lower	0.01714
20	Value-MWTRV	0.014359
21	EMA	0.013147
22	Close	0.011019
23	Value-MKTCP	0.005684
24	High	0.004809
25	Low	0.000132
26	Value-BCDDY	0

Table 6.1: Weka 3: Correlation ranking filter applied on small dataset with BTC1!CME

6.4 Model

Model is the name of the Python module allowing to set up the different machine learning models. The instantiation of the class `MachineLearning` contained in the file `machinelearning.py` allows to call the five methods generating the five models (LR, SVM, LSTM, CNN, BNN) and their results.

To use of the machine learning models, the Position column is changed from boolean to integer. The date is converted from datetime to epoch timestamp (in milliseconds from 01/01/1970). And then the dataset is divided into two parts. A first part for the training (70%) and a second for the tests (30%).

Below we will see the implementation of the five machine learning models used, in order: LR and SVM, CNN, LSTM and finally BNN.

6.4.1 Logistic regression and support vector machine

The library Sklearn (Pedregosa et al., 2011) allows us to create the logistic regression (LR) and support vector machine (SVM). The LR uses the Broyden-Fletcher-Goldfarb-Shanno (lbfgs) limited-memory solver and the SVM uses a radial basis function kernel (rbf).
The libraries Yellowbrick (Bengfort et al., 2021) and Sklearn (Pedregosa et al., 2011) allow us to visualise the following results either graphically or textually:

- Receiver operating characteristic/area under the curve (ROCAUC).
- Root mean squared error (RMSE).
- Confusion matrix.
- Precision.
- Recall.
- Accuracy.
- F1 score.

6.4.2 Convolutional neural network

The CNN is realised with the help of the framework Keras (Chollet et al., 2015). The architecture of our CNN has been implemented following the model described in (Sezer and Ozbayoglu, 2018) and used by (Chen, 2020) to predict the closing price on the New York stock exchange (S&P 500).
In order we use a Conv2D layer (shape: 1, 47, 8), a MaxPooling2D layer (shape: 1, 11, 8), then we flatten the network (shape: 88), then a dense layer (shape: 4) and finally a last dense layer (shape: 1) to get a prediction.

Our CNN uses a period of 50 days to make its predictions. We had to modify the shape of our dataset to make it compatible with the CNN. To do this transformation, we transformed the input from a csv file into an image file. In the figure 6.2 you can see three images that were each made from 50 days of data.

We observe in the 13th column of pixels (x = 13), white and black pixels. This column is the one named Position in our csv used as a label.
To read the image, the price increases are in black (1) and the price decreases/stagnations are in white (0). Each line corresponds to the 27 attributes collected during one day, and there are 50 days of data recorded in each image.

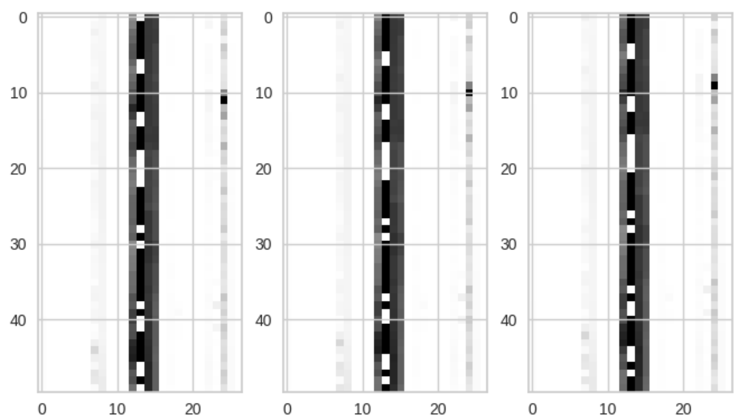

FIGURE 6.2: The first three images from the 2012-2021 dataset for our CNN training set

In the next section 6.4.3 we will see the implementation of the LSTM which uses the same framework and the same statistical evaluation methods as the CNN .

6.4.3 Long short-term memory

We introduced the long short-term memory in detail in section 2.1.3.4, and now we will discuss its implementation.

The preparation of the data for the LSTM was done following the model used in Brownlee (2017). This LSTM is composed of 50 units (50 day time period) and uses the sigmoid activation function to make its prediction. Moreover, the chosen optimiser is adam and the loss function is the mean squared error.

The result of our LSTM is a regression and not a classification like the previous models (LR and SVM). Therefore, other statistical metrics were used to study the results of the model such as the R2 score and the root mean squared error. Nevertheless, we wished to compare the results of the LSTM with other models. To do so, during our implementation we discretised the results between 0 (Bitcoin price stagnates or decreases) and 1 (Bitcoin price increases) in order to allow a classification and obtain the precision, recall, F1 score and accuracy.

6.4.4 Bayesian neural network

The Bayesian neural network (BNN), whose operating logic has been described in the section 2.1.3.5, is implemented with the torchbn framework (Kim, 2020).

The main interest of a Bayesian neural network is its singularity in the way it renders predictions. Indeed, the other solutions we have presented LR, SVM, LSTN and CNN give a result in the form of classification or regression. Here, the BNN gives a result but also a confidence interval of probability of occurrence of the result. For example, there is a 40-60% chance that the price of Bitcoin will increase. Although this characteristic is very interesting, this value is hardly comparable with other models. Therefore we had to use the BNN in a purely predictive way. This has the consequence of losing a large part of its interest.

We have implemented here the BNN with two layers of Bayesian Linear and a ReLU activation, prior μ fixed at 0 and prior σ at 0.1. The first layer takes one hundred outputs as output parameters and the second takes one hundred as input parameters and two as output parameters to make a prediction.

Chapter 7

Results

In this chapter we will analyse and compare the results obtained after implementing the solutions described in the chapter 6.

As discussed in Chapter 6.1, we used two datasets. (Mallqui and Fernandes, 2019) suggests that having a non-preprocessed dataset, with all its attributes, allows for a better determination of the Bitcoin price over their wider time interval. We therefore made a first dataset, that contains data from twenty-six features over the period 2012 to 2021. Then a second one, which we preprocessed with the Correlation analysis (Corr) proposed by Weka. This method is based on the use of the Pearson correlation coefficient to select the twenty most relevant attributes. In addition, we added the BTC1!CME to the latter in order to take into account the derivatives linked to Bitcoin. As the BTC1!CME only exists since 2017, this second dataset covers a smaller time span from 2017 to 2021.

After implementing and testing the machine learning models with the two datasets, the second dataset proved to be unusable with some machine learning models like logistic regression (LR) and support-vector machine (SVM). With the 2017-2021 dataset including the BTC1!CME, the results were all upward predictions of the bitcoin price for LR and SVM. For the other models, the results were worse than those from the 2012-2021 dataset. We have therefore decided not to show the graphs from the 2017-2021 dataset to avoid being redundant and uninformative. But all the results will be in the comparison table 7.3.

In the following sections we will analyse the results of each model. First we introduce the results obtained and we explain their significance. Then we will discuss the results of similar research and finally we will put in perspective this current knowledge with our results in order to better understand their significance.

7.1 Analysis of machine learning models results

7.1.1 Results: logistic regression

We discussed the concept of air under curve (AUC) in section 3.2 to compare model results. In figure 7.1, four curves have been generated, in our case the macro-average is of little interest as it is adapted for multiclasses and we have only two. The two receiver operating characteristic (ROC) curves are associated with the two classes UP and DOWN and the micro-average is harmonic mean of the micro-precision (equation 3.6) and micro-recall (equation 3.7).

The values are very close to 0.5 which shows that this model has results close to a model making random predictions. Logistic regression (LR) does not provide reliable results as shown by its ROC curves (figure 7.1). Furthermore, the micro and macro average have different values which shows that our dataset is slightly imbalanced.

Furthermore, the table 7.2 shows that the LR has the lowest accuracy with an F1-score of 0.32 and 0.64 too far away from 1, a value meaning ideal precision and recall. The LR is therefore not the best performing model we have tested.

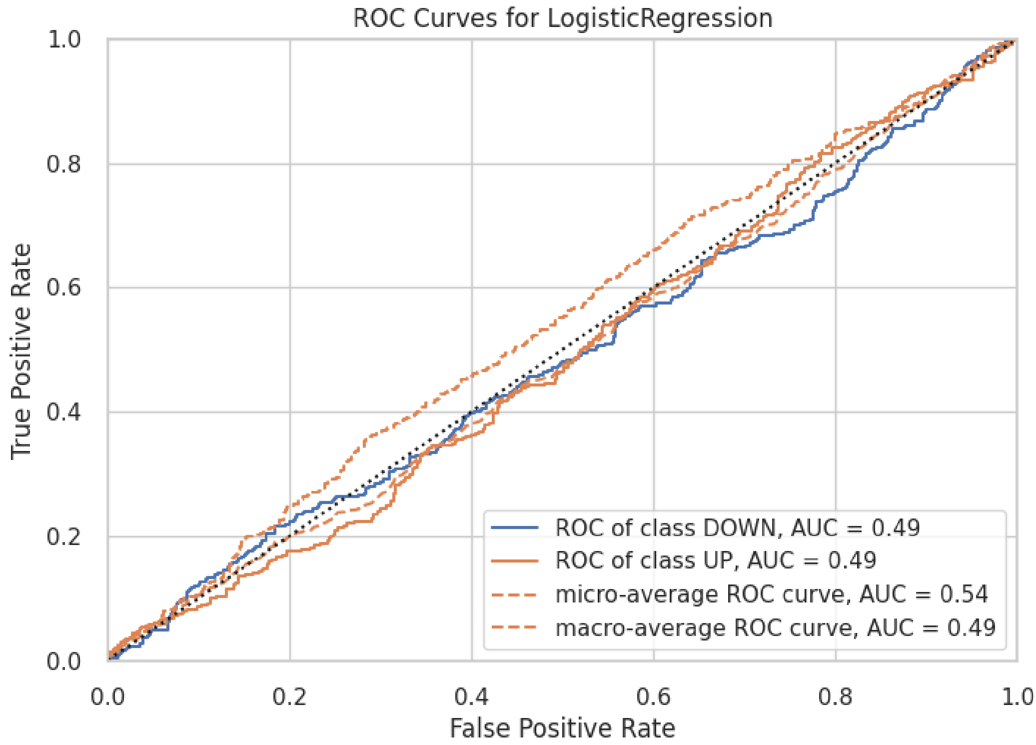

FIGURE 7.1: Logistic regression, ROC curves

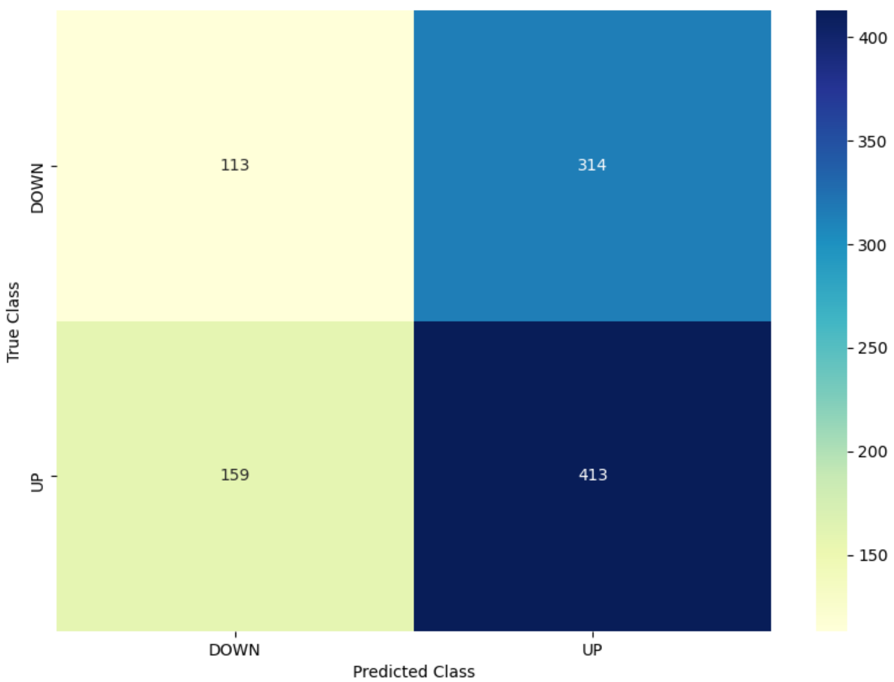

FIGURE 7.2: Logistic regression, confusion matrix

According to (Jang and Lee, 2018)] and (Shrestha, 2020), LR can be subject to the multicollinearity problem. They describe this phenomenon as the presence of variables that in addition to being correlated to the value to be predicted, are highly correlated to each other. Furthermore, they state that in the case where the multicollinearity problem is present, the addition of features worsens this phenomenon. Two consequences follow, the first is a decrease in the impact of the most relevant features on the model output. The second is an increase in the standard errors, which makes it more difficult to interpret the results. More globally, this leads to a decrease in the predictive capacity of the model. (Bhandari, 2020) explains that the variable inflation factors (VIF) is an indicator of multicollinearity. (Sarstedt et al., 2014) states that a VIF with a value higher than five would indicate the presence of collinearity.

In our case, the figure 7.1 demonstrates that the multicollinearity phenomenon impacts our dataset. Indeed, among the twenty features only ten (in bold) are not affected because they have a value lower than five.

Features	**VIF**
date	6.90e+00
open	2.21e+03
high	3.53e+03
low	1.77e+03
close	3.10e+03
EMA	2.63e+03
MA	8.44e+01
Volume	**2.30e+00**
Volume MA	**3.00e+00**
Basis	9.01e+15
Upper	1.00e+15
Lower	1.36e+14
RSI	**1.60e+00**
Position	**1.20e+00**
Close-crude oil	**4.70e+00**
Close-xau	**4.40e+00**
Value-DIFF	9.40e+01
Value-MWNTD	**4.70e+00**
Value-MWTRV	**1.10e+00**
Value-MIREV	2.31e+01
Value-HRATE	8.10e+01
Value-BLCHS	6.60e+01
Value-AVBLS	1.89e+01
Value-BCDDY	**1.10e+00**
Value-ATRCT	**1.70e+00**
Value-TOTBC	5.86e+01
Value-MKTCP	3.07e+03

TABLE 7.1: Variance inflation factor (VIF) values to measures the impact of multicollinearity in our dataset

7.1.2 Results: support vector machine

With the support vector machine (SVM) predictions on the 2012-2021 dataset, we obtain a confusion matrix (figure 7.3). This indicates that fewer instances of the UP class were confused with the DOWN class compared to the LR (figure 7.2), 80 against 159.

The confusion matrix allows us to quickly see the dispersion of the results across the classes, but the indicators such as recall, precision or F1-score that are contained in figure 7.2, allow us to bring a more precise point of view and summarize the content of the matrix.

For the SVM, we see that its recall and precision are slightly higher than those of the LR, as is its accuracy which is 3% higher. However, the F1-score of the DOWN class of SVM is slightly lower. Although the SVM performs better than the LR, the SVM is less reliable in detecting price drops.

As our study is based on mining data from 2012 to 2021, 27 features and with more than 3300 instances, our dataset would correspond to a large dataset. As the amount of data increases beyond thousands of instances, the performance of the SVM decreases (Ahmad et al., 2018). (Cervantes et al., 2008) points out that SVMs have to solve the quadratic programming problem and this does not make them a prime candidate for large datasets because of the computational complexity.

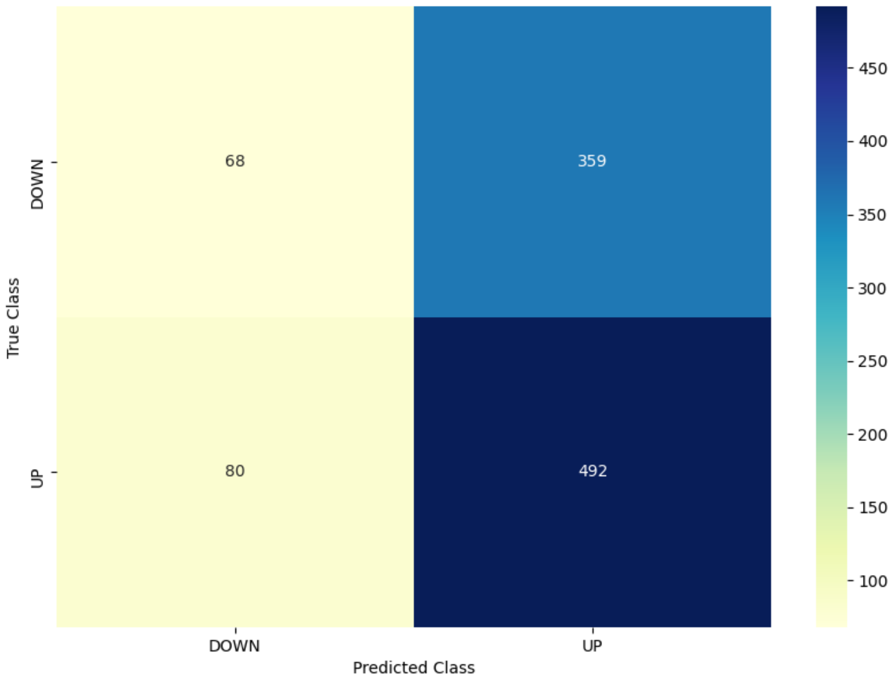

FIGURE 7.3: Support vector machine, confusion matrix

7.1.3 Results: convolutional neural network

The CNN model has a confusion matrix (figure 7.4) close to that of the SVM (figure 7.3).

Table 7.2 shows that accuracy, precision and recall all with in 3%. Moreover, the F1-scores are also very close. Therefore, the CNN and the SVM have similar performance, the SVM has the advantage of having a shorter training time.

Out of 507 DOWN instances, only 13% (66 instances) are correctly classified. In our case, the CNN failed to detect the patterns in the images representing our dataset to predict the decrease in the Bitcoin price.

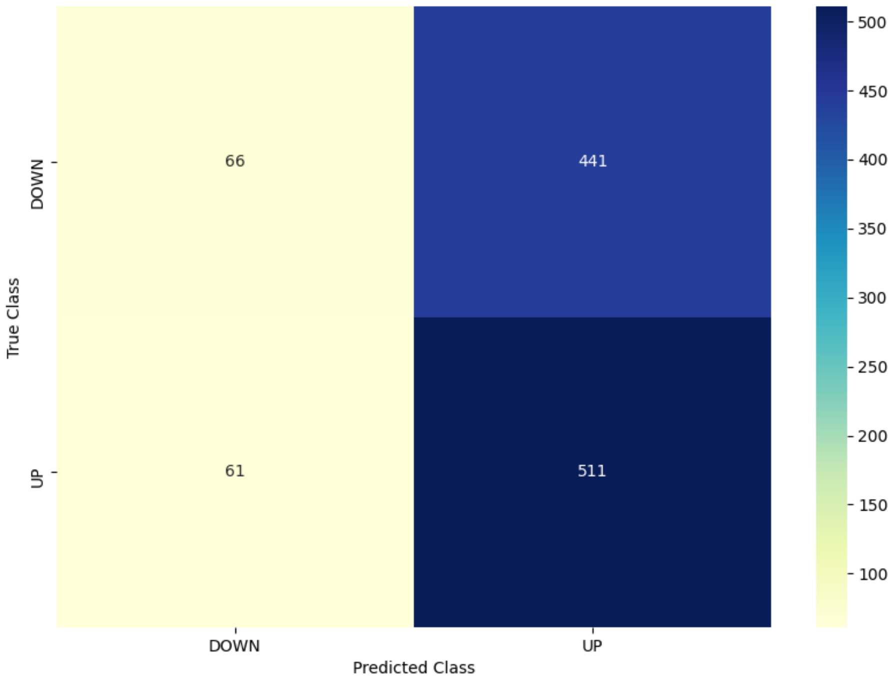

FIGURE 7.4: Convolutional neural network, confusion matrix

7.1.4 Results: long short-term memory

The long short-term memory (LSTM) presents a confusion matrix that shows a clear improvement in the ability of the model to detect price decreases (223). This is twice as high as the LR model with 113 (figure 7.2). The CNN (figure 7.4) and the SVM (figure 7.3) have respectively 66 and 68 price decreases correctly identified.

The F1-score associated with the down class of LSTM is 0.46 which is higher than that of LR (0.32), SVM (0.24) and CNN (0.21). This makes LSTM the most suitable machine learning model for Bitcoin price prediction.

The results of our study, displayed in table 7.2, show that the LSTM obtains the highest accuracy of 59%. This result is confirmed by an F1-score of 0.46-0.66 and the lowest RMSE.

Our research has a methodology similar to (McNally et al., 2018), we used an agile method to create our dataset and we both perform a classification based on daily data, Open, High, Low, Close, Volume of Bitcoin. Their result also highlights the performance of LSTM compared to other models that we did not test such as ARIMA and RNN.

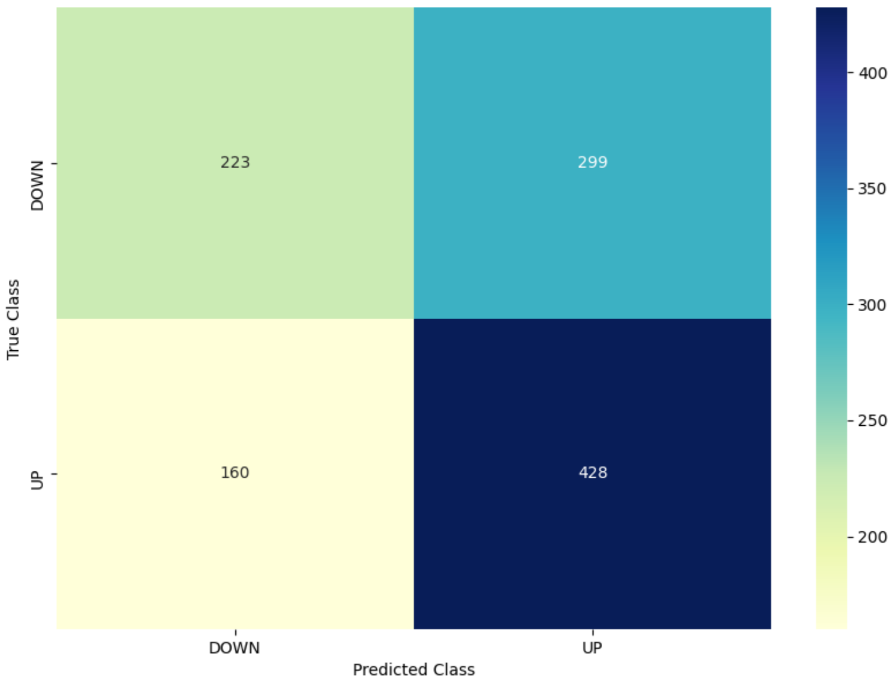

FIGURE 7.5: Long short-term memory, confusion matrix

7.1.5 Results: Bayesian neural network

Following the predictions made with the Bayesian neural network (BNN), its confusion matrix (figure 7.6) highlights the fact that this model has difficulties in identifying price drops. Indeed, of the 532 instances that corresponded to a price decrease, only 17 were correctly classified by the BNN. This observation is confirmed by the F1-score of the DOWN class which is only 0.06. In our context, the BNN is very poorly performing because it classifies almost all instances as UP without making any distinction.

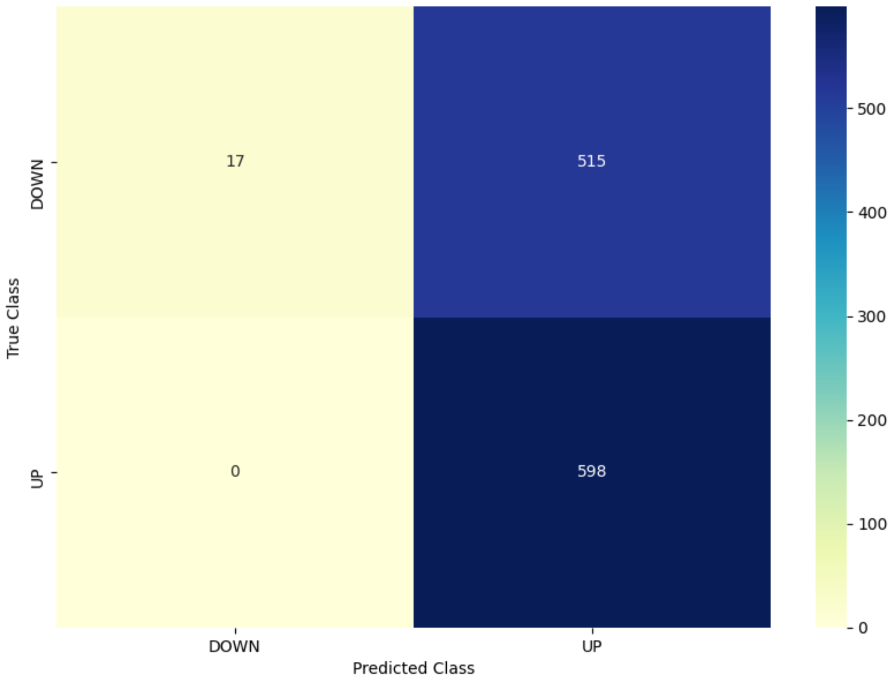

FIGURE 7.6: Bayesian neural network, confusion matrix

We have generated other graphs to provide an alternative method of visualising the results obtained during and after training the machine learning models. This content is available in the appendix A.

7.2 Comparison of results from machine learning models

In this section we will compare different machine learning model results. These models were introduced in section 2.1.3 and their implementation was described in section 6.4.

The tables are constituted in the following way:

- The columns represent the different types of machine learning models.
- The rows contain the statistical metrics discussed in the part 3.2, they allow to compare the machine learning models. Although we aim to improve the accuracy of our models, the most relevant indicators for model comparison are the F1-score and the RMSE. The RMSE allows to put more weight on the large errors because it corresponds to the square root of the errors. And the F1-score, as shown in the equation 3.5, is the harmonic mean of recall and precision. This allows it to better identify incorrectly determined items.

Model	LR	SVM	CNN	LSTM	BNN
Accuracy	0.53	0.56	0.54	0.59	0.54
Precision	0.57	0.58	0.54	0.58	0.54
Recall	0.72	0.86	0.89	0.77	1
F1-score	0.32 - 0.64	0.24 - 0.69	0.21 - 0.67	0.46 - 0.66	0.06 - 0.70
RMSE	0.69	0.66	0.51	0.49	0.68
R2-score	-	-	-0.03	0.05	-

TABLE 7.2: Results of the dataset for the period from 03/06/2012 to 15/07/2021, without BTC1!CME

Model	LR	SVM	CNN	LSTM	BNN
Accuracy	0.54	0.54	0.46	0.56	0.54
Precision	0.54	0.54	0.61	0.62	0.53
Recall	1	1	0.07	0.49	1
F1-score	0.00 - 0.70	0 - 0.70	0.61 - 0.13	0.55 - 0.55	0.08 - 0.70
RMSE	0.66	0.66	0.51	0.52	0.68
R2-score	-	-	-0.06	0.05	-

Table 7.3: Results of the dataset for the period from 17/12/2017 to 15/07/2021, with BTC1!CME

The first thing we observe is the difference between the results obtained via the dataset extended from 2012 to 2021 and the one from 2017 to 2021 containing the BTC1!CME. Indeed, the one with more data (2012-2021), allows all the machine learning models to converge better and thus to obtain better results.

In our dataset which contains 3300 days of data, a majority (UP: 1831, DOWN: 1499) are associated with an increase in the Bitcoin price. Models like BNN, CNN and SVM tend to detect only price increases, as shown by their low F1-score DOWN in the table 7.2. The LR is slightly less subject to this, however its accuracy is only 53% and its error measured by the RMSE is the highest of the models. The LSTM stands out with the highest F1-score and accuracy and the lowest RMSE error.

Our finding is based on daily predictions of the bitcoin price. (Jaquart et al., 2021) also demonstrated the effectiveness of LSTM on short 60 minute predictions showing that LSTM outperforms models such as feedforward neural networks, random forest or LR. However, over the 5min-horizon, the gradient boosting classifiers (GBC) performed better. In our case, we can therefore only conclude that the LSTM is suitable for daily classification.

In the following chapter 8 we will conclude and extrapolate the interpretation of these results.

Chapter 8

Conclusion and future work

8.1 Conclusion

The objective of our study was to use a complete historical dataset to establish a comparative assessment of five machine learning models that had previously shown their effectiveness. To do so, we collected daily data and some of them are directly related to Bitcoin such as OHLCV and others have a correlation with its price, such as the price of gold and crude oil.

To begin with, we have built up a comprehensive dataset over a large period of time, from 2012 to 2021. A large number of sites offer free datasets or APIs giving access to Bitcoin-related data such as (CoinDesk, 2017), (Bartolome del Canto, 2021) or (Garcia, 2021). However, as (Quandl, 2021*b*) points out, the free nature of this information often implies a lack of quality control of the data, poor documentation or a lack of transparency regarding the provenance of the data. We were confronted with these problems during the creation of our dataset. The documentation of the (CoinDesk, 2017) was minimalist, the Investing.com python package (Bartolome del Canto, 2021) would generate errors in the data collection (decalibration of values over one day) or (Garcia, 2021) aggregated daily OHLC data from several exchanges without weighing them by transaction volume.

We implemented five machine learning models, LR, SVM, CNN, LSTM and BNN to predict the increase or decrease of the Bitcoin price over a daily time interval. Two datasets were constructed based on (BraveNewCoin, 2015) the BNC Bitcoin Liquid Index (BLX), a source that is quarterly reviewed and has an accuracy of eight decimal places. One dataset with all attributes from 2012 to 2021, a second one from 2017 to 2021 with the twenty attributes selected by the correlation analysis on Weka plus the BTC1!CME.

The results obtained in the table 7.2 show that the overall performance of all models on the dataset is better than 50% accuracy. However, some models such as SVM, CNN and BNN have shown poor ability to detect Bitcoin price declines. Over the last ten years, the Bitcoin price has been on a rising phase. If the models detect this rising phase and only predict price increases, their accuracy will necessarily be better than 50%, but this does not make them efficient models.
Based on accuracy alone (7.2), LR and BNN have the worst results with respectively 53% and 54% accuracy. Contrary to LSTM and SVM which have the best accuracy with 59% and 56%. Nevertheless, the F1-scores (table 7.2) nuance the capacity of the models to correctly detect the UP and DOWN classes. The F1-score of the DOWN class of the SVM shows a difficulty in detecting price decreases whereas the LR does better.

According to our results, BNN, CNN and SVM have too low F1-score on the DOWN class to be considered relevant for Bitcoin price prediction. These models seem to detect the overall trend of the Bitcoin price and lose their ability to detect price declines. The LR is slightly less affected by this phenomenon than the DOWN class, however the error shown by its RMSE is high, resulting in low accuracy.
The LSTM stands out and presents the best performance as it has the best accuracy (59%), high F1-score especially on the DOWN class and the lowest RMSE. Taking into account the theoretical explanations of the five machine learning models in section 2.2, the results were partly predictable. Indeed, due to its architecture

described in section 2.1.3.4, the LSTM is supposed to perform well on time prediction, unlike the CNN which uses images as input and is traditionally used for image classification.

Furthermore, as our objective was to evaluate different machine learning models on an equal footing, we did not pre-process the data to optimise a particular model. For example, the multicollinearity phenomenon that negatively impacted the LR could have been reduced by merging features that were too correlated.

To conclude, each of the machine learning models tested in our study has its own architecture. As a result, their performance depends mainly on the context in which they are used. In our case, it is the LSTM that stands out positively because we are evolving on a time series prediction and not on images that would benefit the CNN or a small number of uncorrelated variables that favour the LR.

In the next section 8.2 we will discuss new approaches to broaden the spectrum of this topic.

8.2 Future work

We have tested several machine learning models using data with a quantitative analysis approach in order to perform a classification of the Bitcoin price evolution. A different approach would be to evaluate the ability of the models to predict not a trend but directly the price of Bitcoin. We could also extend the field of analysis with qualitative data. An example of this is using sentiment analysis gathered from different media and communication channels.

From the point of view of optimising results and not benchmarking models as we have done, it would be interesting to make the architecture of machine learning models more complex by combining them. Indeed (Cocco et al., 2021) has demonstrated that the combination of a support vector regression (SVR) and a LSTM model has a lower mean absolute percentage error (MAPE) than its individual counterparts.

Appendix A

Results

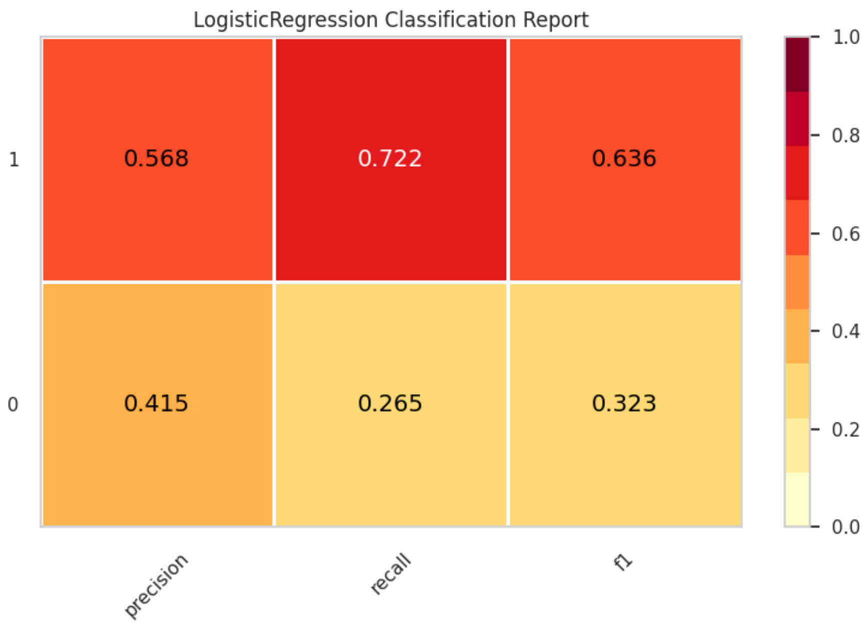

FIGURE A.1: Logistic regression, classification report

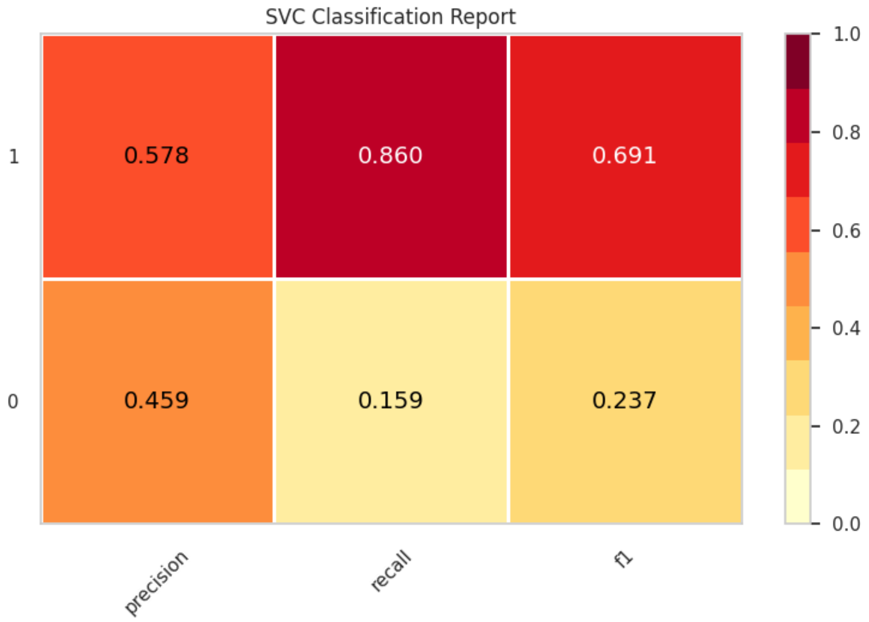

FIGURE A.2: Support vector machine, classification report

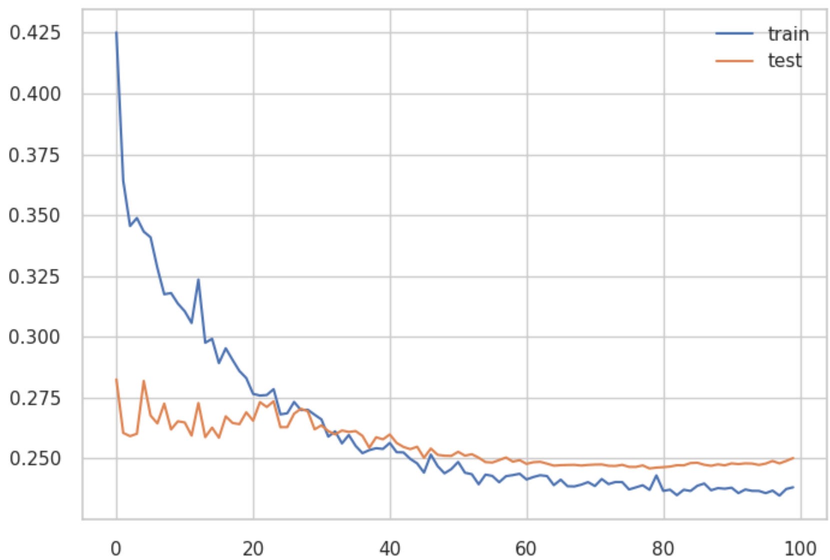

FIGURE A.3: Convolutional neural network, validation-loss evolution curve during training and testing

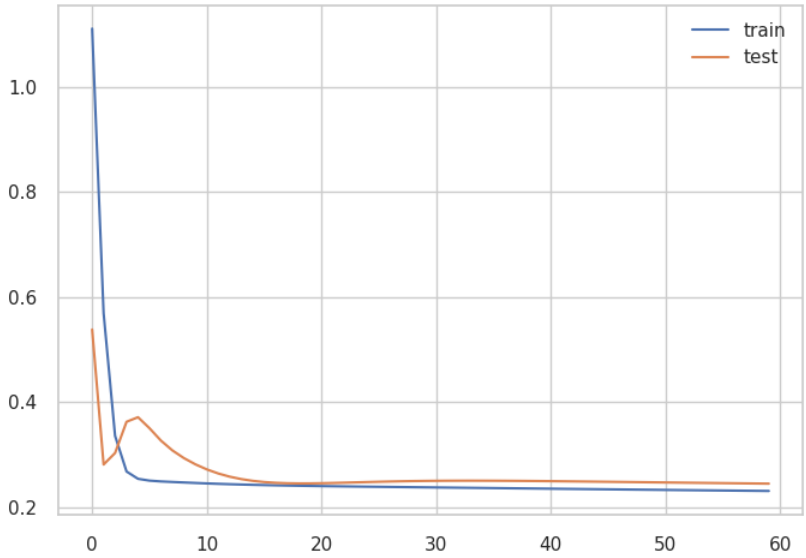

Figure A.4: Long short term memory, validation-loss evolution curve during training and testing

Bibliography

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A. and Tan, R. S. (2017), 'A Deep Convolutional Neural Network Model to Classify Heartbeats', *Computers in Biology and Medicine* **89**.
URL: *https://doi.org/10.1016/j.compbiomed.2017.08.022*

Ahmad, I., Basheri, M., Iqbal, M. J. and Rahim, A. (2018), 'Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection', *IEEE Access* **6**, 33789–33795. Conference Name: IEEE Access.
URL: *https://ieeexplore.ieee.org/document/8369054*

Aivaliotis, P., Zampetis, A., Michalos, G. and Makris, S. (2017), 'A Machine Learning Approach for Visual Recognition of Complex Parts in Robotic Manipulation', *Procedia Manufacturing* **11**, 423–430.
URL: *https://www.sciencedirect.com/science/article/pii/S2351978917303347*

Alessandretti, L., ElBahrawy, A., Aiello, L. M. and Baronchelli, A. (2018), 'Anticipating Cryptocurrency Prices Using Machine Learning', *Complexity* **2018**, e8983590. Publisher: Hindawi.
URL: *https://www.hindawi.com/journals/complexity/2018/8983590/*

Ante, L. (2021), How Elon Musk's Twitter Activity Moves Cryptocurrency Markets, SSRN Scholarly Paper ID 3778844, Social Science Research Network, Rochester, NY.
URL: *https://papers.ssrn.com/abstract=3778844*

Back, A. (2002), 'Hashcash - A Denial of Service Counter-Measure', p. 10.
URL: *http://www.hashcash.org/papers/hashcash.pdf*

Baratloo, A., Hosseini, M., Negida, A. and El Ashal, G. (2015), 'Part 1: Simple Definition and Calculation of Accuracy, Sensitivity and Specificity', *Emergency* **3**(2), 48–49.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4614595/*

Bartolome del Canto, A. (2021), 'Investpy: Financial Data Extraction from Investing.com with Python'.
URL: *https://investpy.readthedocs.io/*

Benediktsson, J. (2021), 'mrjbq7/ta-lib'. original-date: 2012-03-23T17:14:15Z.
URL: *https://github.com/mrjbq7/ta-lib*

Bengfort, B., Bilbro, R., Johnson, P., Billet, P., Roman, P., Deziel, P., McIntyre, K., Gray, L., Ojeda, A., Schmierer, E., Morris, A. and Morrison, M. (2021), 'Yellowbrick v1.3'. Language: en.
URL: *https://zenodo.org/record/1206239*

Berg, C., Davidson, S. and Potts, J. (2020), 'Proof of Work as a Three-Sided Market', *Frontiers in Blockchain* **3**. Publisher: Frontiers.
URL: *https://www.frontiersin.org/articles/10.3389/fbloc.2020.00002/full*

Bhamare, D. and Suryawanshi, P. (2018), 'Review on Reliable Pattern Recognition with Machine Learning Techniques', *Fuzzy Information and Engineering* **10**(3), 362–377. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/16168658.2019.1611030.
URL: *https://doi.org/10.1080/16168658.2019.1611030*

Bhandari, A. (2020), 'Multicollinearity | Detecting Multicollinearity with VIF'.
URL: *https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/*

Binance (2021), 'BTCUSDT Binance Spot'.
URL: *https://www.binance.com*

Bitcoincharts (2021), 'Bitcoincharts | Markets API'.
URL: *https://bitcoincharts.com/about/markets-api/*

Blockchain (2021), 'pools'.
URL: *https://www.blockchain.com/charts/pools*

Blockchain.com (2021), 'Blockchain.com Explorer |'.
URL: *https://www.blockchain.com/explorer*

BraveNewCoin (2015), 'BNC Bitcoin Liquid Index Brave New Coin'.
URL: *https://bravenewcoin.com/enterprise-solutions/indices-program/blx*

Brownlee, J. (2017), 'Multivariate Time Series Forecasting with LSTMs in Keras'.
URL: *https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/*

Brownlee, J. (2019), 'How to Fix the Vanishing Gradients Problem Using the ReLU'.
URL: *https://machinelearningmastery.com/how-to-fix-vanishing-gradients-using-the-rectified-linear-activation-function/*

Cavalli, S. and Amoretti, M. (2021), 'CNN-based multivariate data analysis for bitcoin trend prediction', *Applied Soft Computing* **101**, 107065.
URL: *https://www.sciencedirect.com/science/article/pii/S1568494620310036*

Cervantes, J., Li, X., Yu, W. and Li, K. (2008), 'Support vector machine classification for large data sets via minimum enclosing ball clustering', *Neurocomputing* **71**(4), 611–619.
URL: *https://www.sciencedirect.com/science/article/pii/S0925231207002962*

Chartois, J. (2021), 'Research-ML-BTC'.
URL: *https://gitlab-student.macs.hw.ac.uk/jmmc2000/f21mp_2020-2021*

Chen, S. (2020), 'Stock Closing Price Prediction with CNN'.
URL: *https://kaggle.com/salmanhiro/stock-closing-price-prediction-with-cnn*

Chollet, Francois and Other (2015), 'Keras'.
URL: *https://github.com/fchollet/keras*

CME (2021), 'CME CF Bitcoin Reference Rate & Real Time Index - CME Group'.
URL: *http://www.cmegroup.com/trading/cf-bitcoin-reference-rate.html*

Cocco, L., Tonelli, R. and Marchesi, M. (2021), 'Predictions of bitcoin prices through machine learning based frameworks', *PeerJ Computer Science* **7**, e413.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8022579/*

CoinDesk (2017), 'Bitcoin Price Index API'.
URL: *https://www.coindesk.com/api*

CoinDesk (2021*a*), 'Bitcoin Price Index — CoinDesk 20'.
URL: *https://www.coindesk.com/price/bitcoin*

CoinDesk (2021*b*), 'CoinDesk: Bitcoin, Ethereum, Crypto News and Price Data'.
URL: *https://www.coindesk.com/*

CoinMarketCap (2021), 'Cryptocurrency Prices, Charts And Market Capitalizations'.
URL: *https://coinmarketcap.com/*

Corbet, S., Lucey, B. and Yarovaya, L. (2021), 'Bitcoin-energy markets interrelationships - New evidence', *Resources Policy* **70**, 101916.
URL: *https://www.sciencedirect.com/science/article/pii/S0301420720309478*

de Best, R. (2021*a*), 'Biggest crypto exchanges 2021'.
URL: *https://www.statista.com/statistics/864738/leading-cryptocurrency-exchanges-traders/*

de Best, R. (2021*b*), 'Cryptocurrency market value 2010-2021'.
URL: *https://www.statista.com/statistics/730876/cryptocurrency-maket-value/*

Detzel, A., Liu, H., Strauss, J., Zhou, G. and Zhu, Y. (2021), 'Learning and predictability via technical analysis: Evidence from bitcoin and stocks with hard-to-value fundamentals', *Financial Management* **50**(1), 107–137.
URL: *https://onlinelibrary.wiley.com/doi/10.1111/fima.12310*

Dolphin, R. (2021), 'LSTM Networks | A Detailed Explanation'.
URL: *https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9*

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, J. R. G., Gruber, B., Lafourcade, B., Leitão, P. J., Münkemüller, T., McClean, C., Osborne, P. E., Reineking, B., Schröder, B., Skidmore, A. K., Zurell, D. and Lautenbach, S. (2013), 'Collinearity: a review of methods to deal with it and a simulation study evaluating their performance', *Ecography* **36**(1), 27–46. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1600-0587.2012.07348.x.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0587.2012.07348.x*

Ekelund, S. (2011), 'ROC curves – what are they and how are they used?'.
URL: *https://acutecaretesting.org/en/articles/roc-curves-what-are-they-and-how-are-they-used*

El Naqa, I. and Murphy, M. J. (2015), What Is Machine Learning?, *in* I. El Naqa, R. Li and M. J. Murphy, eds, 'Machine Learning in Radiation Oncology: Theory and Applications', Springer International Publishing, Cham, pp. 3–11.
URL: *https://app.dimensions.ai/details/publication/pub.1011680030*

Elith, J., Graham, C. H., Anderson, R. P., Dudík, M., Ferrier, S., Guisan, A., Hijmans, R. J., Huettmann, F., Leathwick, J. R., Lehmann, A., Li, J., Lohmann, L. G., Loiselle, B. A., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., Overton, J. M. M., Peterson, A. T., Phillips, S. J., Richardson, K., Scachetti-Pereira, R., Schapire, R. E., Soberón, J., Williams, S., Wisz, M. S. and Zimmermann, N. E. (2006), 'Novel methods improve prediction of species' distributions from occurrence data', *Ecography* **29**(2), 129–151. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2006.0906-7590.04596.x.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2006.0906-7590.04596.x*

Frank, E. (2019), 'Micro averages in multi-class classification'.
URL: *https://waikato.github.io/weka-blog/posts/2019-02-16-micro_ average/*

Garcia, V. M. (2021), 'bitcoin-OHLC-generator'. original-date: 2017-12-07T11:19:10Z.
URL: *https://github.com/victormartingarcia/bitcoin-OHLC-generator*

Georgiadis, E. (2019), 'How many transactions per second can bitcoin really handle ? Theoretically.'.
URL: *http://eprint.iacr.org/2019/416*

Georgoula, I., Pournarakis, D., Bilanakos, C., Sotiropoulos, D. N. and Giaglis, G. M. (2015), 'Using Time-Series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices', *SSRN Electronic Journal* .
URL: *http://www.ssrn.com/abstract=2607167*

GNU (2007), 'GNU AGPL'.
URL: *https://www.gnu.org/licenses/agpl-3.0.txt*

Google (2020), 'Logistic Regression: Calculating a Probability'.
URL: *https://developers.google.com/machine-learning/crash-course/logistic-regression/calculating-a-probability*

Goutte, C. and Gaussier, E. (2005), A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation, *in* D. E. Losada and J. M. Fernández-Luna, eds, 'Advances in Information Retrieval', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 345–359.
URL: *http://dx.doi.org/10.1007/978-3-540-31865-1_ 25*

Gyan, M. K. (2015), 'FACTORS INFLUENCING THE PATRONAGE OF STOCKS ON THE GHANA STOCK MARKET', p. 57.
URL: *http://dspace.knust.edu.gh/bitstream/123456789/8712/1/MARTIN%20KWAKU%20G*

Hall, M., Frank, E., Holmes, G., Reutemann, P. and H. Witten, I. (2009), 'The {WEKA} data mining software: an update'.
URL: *https://www.cs.waikato.ac.nz/ml/weka/*

Hayes, A. (2020), ‘Technical Analysis’.
URL: *https://www.investopedia.com/terms/t/technicalanalysis.asp*

Hertig, A. (2019), ‘Bitcoin Cash Miners Undo Attacker’s Transactions With ’51% Attack”.
URL: *https://www.coindesk.com/bitcoin-cash-miners-undo-attackers-transactions-with-51-attack*

Hochreiter, S. and Jurgen, S. (1997), *Long Short-Term Memory*, Neural Computation.
URL: *https://doi.org/10.1162/neco.1997.9.8.1735*

Hodgson, B. (2020), ‘Statistical release: OTC derivatives statistics at end-June 2020’, p. 8.
URL: *https://www.bis.org/publ/otc_hy2011.pdf*

Huang, J.-Z., Huang, W. and Ni, J. (2019), ‘Predicting bitcoin returns using high-dimensional technical indicators’, *The Journal of Finance and Data Science* **5**(3), 140–155.
URL: *https://www.sciencedirect.com/science/article/pii/S2405918818300928*

Hussey, M. (2021), ‘How Hedge Funds Are Changing the Crypto Landscape’. Section: Features.
URL: *https://decrypt.co/70153/how-hedge-funds-are-changing-the-crypto-landscape*

Investing.com (2021), ‘Investing.com BTC/USD Historical Rates’.
URL: *https://uk.investing.com/indices/investing.com-btc-usd-historical-data*

Jang, H. and Lee, J. (2018), ‘An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information’, *IEEE Access* **6**, 5427–5437. Conference Name: IEEE Access.
URL: *https://doi.org/10.1109/ACCESS.2017.2779181*

Jaquart, P., Dann, D. and Weinhardt, C. (2021), 'Short-term bitcoin market prediction via machine learning', *The Journal of Finance and Data Science* **7**, 45–66.
URL: *https://www.sciencedirect.com/science/article/pii/S2405918821000027*

Kaminski, J. (2016), 'Nowcasting the Bitcoin Market with Twitter Signals', *arXiv:1406.7577 [cs]* . arXiv: 1406.7577.
URL: *http://arxiv.org/abs/1406.7577*

Kim, H. (2020), 'torchbnn: Bayesian Neural Network for PyTorch'.
URL: *https://github.com/Harry24k/bayesian-neural-network-pytorch*

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017), 'ImageNet classification with deep convolutional neural networks', *Communications of the ACM* **60**(6), 84–90.
URL: *https://dl.acm.org/doi/10.1145/3065386*

Mallqui, D. C. A. and Fernandes, R. A. S. (2019), 'Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques', *Applied Soft Computing* **75**, 596–606.
URL: *https://www.sciencedirect.com/science/article/pii/S1568494618306707*

Manoharan, P., Sharma, A., Vijayakumar, V., Bhardwaj, V., Sharma, A. P., Iqbal, R. and Kumar, R. (2020), 'Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system', *Computers & Electrical Engineering* **81**, 106527.
URL: *https://www.sciencedirect.com/science/article/pii/S0045790618331343*

Matta, M., Lunesu, M. I. and Marchesi, M. (2015), 'Bitcoin Spread Prediction Using Social And Web Search Media'.
URL: *https://www.researchgate.net/publication/279917417_Bitcoin_Spread_Prediction_Us*

Mckinney, W. (2011), 'pandas: a Foundational Python Library for Data Analysis and Statistics', *Python High Performance Science Computer* .
URL: *https://www.researchgate.net/publication/265194455_pandas_a_Foundational_Pytho*

McNally, S., Roche, J. and Caton, S. (2018), Predicting the Price of Bitcoin Using Machine Learning, *in* '2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)', IEEE, Cambridge, pp. 339–343.
URL: *https://ieeexplore.ieee.org/document/8374483/*

Mern, J., Anderson, S. and Poothokaran, J. (2017), 'Using Bitcoin Ledger Network Data to Predict the Price of Bitcoin', p. 6.
URL: *http://cs229.stanford.edu/proj2017/final-reports/5228421.pdf*

Mittal, A., Dhiman, V., Singh, A. and Prakash, C. (2019), Short-Term Bitcoin Price Fluctuation Prediction Using Social Media and Web Search Data, *in* '2019 Twelfth International Conference on Contemporary Computing (IC3)', pp. 1–6. ISSN: 2572-6129.
URL: *https://ieeexplore.ieee.org/document/8844899*

Nakamoto, S. (2008), 'Bitcoin: A Peer-to-Peer Electronic Cash System', p. 9.
URL: *https://bitcoin.org/bitcoin.pdf*

Nicolas, H. (2014), 'It Will Cost You Nothing to 'Kill' a Proof-of-Stake Crypto-Currency', *SSRN Electronic Journal* .
URL: *http://www.ssrn.com/abstract=2393940*

Noble, W. S. (2006), 'What is a support vector machine?', *Nature Biotechnology* **24**(12), 1565–1567.
URL: *http://www.nature.com/articles/nbt1206-1565*

Nti, I. K., Adekoya, A. F. and Weyori, B. A. (2020), 'A systematic review of fundamental and technical analysis of stock market predictions', *Artificial Intelligence Review* **53**(4), 3007–3057.
URL: *https://doi.org/10.1007/s10462-019-09754-z*

Pandas (2021), 'pandas - Python Data Analysis Library'.
URL: *https://pandas.pydata.org/about/*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), 'Scikit-learn: Machine Learning in {P}ython', *Journal of Machine Learning Research* **12**, 2825–2830.
URL: *https://scikit-learn.org/stable/about.html*

Plotly (2015), 'Collaborative data science'.
URL: *https://plot.ly*

Quandl (2021*a*), 'Quandl Blockchain'.
URL: *https://www.quandl.com*

Quandl (2021*b*), 'Why should I choose premium data over free data? - Quandl Help Center'.
URL: *https://help.quandl.com/article/285-why-should-i-choose-premium-data-over-free-data*

Redman, J. (2019), 'Close to 14,000 Google Scholar Articles Mentioned Bitcoin in 2019 – Featured Bitcoin News'.
URL: *https://news.bitcoin.com/close-to-14000-google-scholar-articles-mentioned-bitcoin-in-2019/*

Salem, A. M. and Darter, M. O. (2006), 'Requirements analysis: A practical object oriented approach', *Journal of Computational Methods in Sciences and Engineering* **6**(5,6 Supplement 1), 191–204.
URL: *https://www.researchgate.net/publication/262362470_Requirements_analysis_A_prac*

Sankar, C. P., Vidyaraj, R. and Kumar, K. S. (2015), 'Trust Based Stock Recommendation System – A Social Network Analysis Approach', *Procedia Computer Science* **46**, 299–305.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050915000885*

SantanderBank (2021), 'BillPay | Quick Pay | Online Bill Pay | Santander Bank'.
URL: *https://www.santanderbank.com/us/personal/resources/mobile/online-bill-pay*

Sarstedt, M., Ringle, C. M., Smith, D., Reams, R. and Hair, J. F. (2014), 'Partial least squares structural equation modeling (PLS-SEM): A useful tool for family business researchers', *Journal of Family Business Strategy* **5**(1), 105–115.
URL: *https://www.sciencedirect.com/science/article/pii/S1877858514000060*

Segal, T. and Boyle, M. (2021), 'Fundamental Analysis'.
URL: *https://www.investopedia.com/terms/f/fundamentalanalysis.asp*

Sezer, O. and Ozbayoglu, M. (2018), 'Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach', *Applied Soft Computing* **70**.
URL: *https://www.researchgate.net/publication/324802031_Algorithmic_Financial_Trading*

Shah, D. and Zhang, K. (2014), 'Bayesian regression and Bitcoin', *arXiv:1410.1231 [cs, math, stat]* . arXiv: 1410.1231.
URL: *http://arxiv.org/abs/1410.1231*

Sharma, N., Jain, V. and Mishra, A. (2018), 'An Analysis Of Convolutional Neural Networks For Image Classification', *Procedia Computer Science* **132**, 377–384.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050918309335*

Shrestha, N. (2020), 'Detecting Multicollinearity in Regression Analysis', *American Journal of Applied Mathematics and Statistics* **8**, 39–42.

SimilarWeb (2021), 'Top Investing Websites Ranking by Traffic in July 2021'.
URL: *http://similarweb.com/top-websites/category/finance/investing/*

Tiwari, A. (2021), 'Crypto derivatives gained steam in 2020, but 2021 may see true growth'.
URL: *https://cointelegraph.com/news/crypto-derivatives-gained-steam-in-2020-but-2021-may-see-true-growth*

TradingView (2021), 'TradingView – Track All Markets'.
URL: *https://www.tradingview.com/*

Turk, D., France, R. and Rumpe, B. (2005), 'Assumptions Underlying Agile Software-Development Processes', *J. Database Manag.* **16**, 62–87.
URL: *https://arxiv.org/abs/1409.6610*

Vassiliadis, S. (2017), 'BITCOIN VALUE ANALYSIS BASED ON CROSS-CORRELATIONS', **22**, 12.
URL: *https://www.icommercecentral.com/open-access/bitcoin-value-analysis-based-on-crosscorrelations.pdf*

Yamashita, R., Nishio, M., Do, R. K. G. and Togashi, K. (2018), 'Convolutional neural networks: an overview and application in radiology', *Insights into Imaging* **9**(4), 611–629. Number: 4 Publisher: SpringerOpen.
URL: *https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9*

Yang, S. (2019), 'Maximum Likelihood Estimation VS Maximum A Posterior'.
URL: *https://towardsdatascience.com/mle-vs-map-a989f423ae5c*

Yu, Y., Si, X., Hu, C. and Zhang, J. (2019), 'A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures', *Neural Computation* **31**(7), 1235–1270.
URL: *https://direct.mit.edu/neco/article/31/7/1235/8500/A-Review-of-Recurrent-Neural-Networks-LSTM-Cells*