Università
Ca'Foscari
Venezia

Master's Degree Programme
in
Global Development and Entrepreneurship

Final Thesis

Neural Networks for Cryptocurrency Price Forecasting

Supervisor
Ch. Prof. Claudio Pizzi

Graduand
Davide Signorile
Matriculation Number 852631

Academic Year
2020/2021

TABLE OF CONTENTS

INTRODUCTION

The ability to predict financial market movements in advance is an issue of considerable importance to the various global players, be they institutional or private investors. The following thesis work is focused on price prediction and consequent price movement of cryptocurrencies and in particular, this is applied to Bitcoin. The advent of the cryptocurrency market dates back to January 2009 when Satoshi Nakamoto introduced Bitcoin, the virtual currency for secure transactions on a decentralized peer-to-peer network that guarantees anonymity and transparency through the use of cryptographic algorithms. In recent years, especially during the current year, interest in the new technology has been growing and several new cryptocurrencies have been launched to the market with the intent to ride the wave of the current trend. The analogy of the world of cryptocurrencies with the financial market is not lacking: just as with equity investments, the investor hopes to make a profit by reselling or repurchasing the assets he owns at a more advantageous price than the previous transaction.

The work focuses on the use of machine learning techniques, specifically neural networks, that use historical data related to the cryptocurrency itself as input data to make forecasts. Several machine learning algorithms have recently been proposed to be used in conjunction with traditional forecasting methods. The great advantage of using the latter comes from their direct application to raw data and their ability to capture the volatility that characterizes the cryptocurrency market, providing good forecasts.

To better understand this elaboration and grasp its objective, it is necessary to have some preliminary knowledge. This is why the first part of the thesis will deal mostly with theoretical aspects. In the first part, cryptocurrencies, in general, will be deepened. Then we will move more specifically to Bitcoin and the Blockchain, the technology on which cryptocurrencies are based. Finally, we'll talk about neural networks, the machine learning technique used in the second part.

This work shows that these algorithms can capture, with a fair and acceptable approximation, past price characteristics to make predictions. Further investigation in this field can take many directions, as it is a complicated discipline made of trial and error, where a unique solution does not exist. One could modify the obtained machine

learning models looking for the best algorithm with the best number of parameters resulting in improved performance or choose a different approach to the problem. However, the fact remains that predicting a time series is always an advantage. These might be suggestions, the analysis in this field is still of wide debate and leaves room for further investigation.

1. INTRODUCTION TO CRYPTOCURRENCIES

1.1 The double-spending problem

The will in recent years has been to develop a payment system that would allow electronic and digital transactions. To better understand this desire to implement digital cash, consider first a simple money transaction, as shown in *Figure 1.1*.

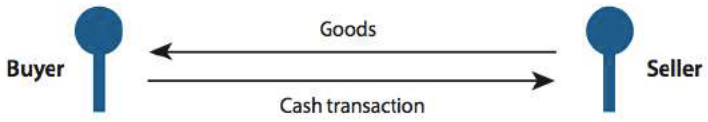


Figure 1.1 - Cash transaction

Money is represented by a physical object, usually a coin or bill. When this object is handed over to another individual, its unit of value is also transferred, without the need to involve a third party. There is therefore a direct relationship, as seen in *Figure 1.1*. No credit relationship arises between the buyer and the seller, and it is for this reason that anonymity can be maintained for both parties.

The advantage of cash lies in the fact that whoever possesses the physical object becomes the owner of the unit of value and therefore the property rights over the units of value circulating in the economy are always clearly established, without the necessary intervention of a central authority. In addition, anyone can participate in a cash payment system, there is free access without authorization, and no one can be excluded.

However, cash has, for example, the disadvantage that the buyer and seller must be physically present at the same place to operate, which in many situations makes its use impractical.

An ideal payment system would be one where monetary value is transferred electronically via data files, as seen in *Figure 1.2*.

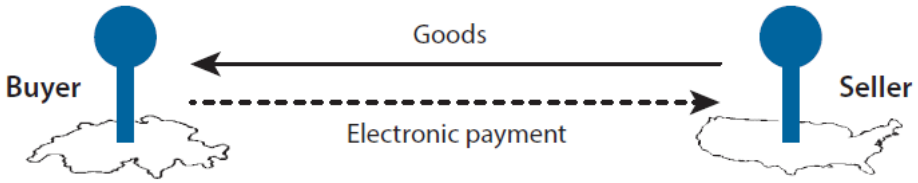


Figure 1.2 - Electronic payment

Such data files retain the advantages of physical money but can circulate freely in electronic networks. Such a file could be sent, for example, via email.

The idea of digital cash goes beyond the limit of cash just mentioned, however, the problem is that electronic data can be copied an infinite number of times and this feature is highly undesirable for money. If data files can be copied and duplicates used as currency, they would not serve as a payment instrument: this drawback is defined as the problem of double-spending[1], that is, the cloning of money which, being virtual, could be duplicated, spent, and used multiple times. E-commerce, for example, requires the use of digital tokens, and in a virtual checkout system, the means of payment could easily be copied and reused to perform the payment. The currency could then be counterfeited and used twice, giving rise to this problem of so-called double-spending.

Traditionally, this inconvenience is solved by relying on a trusted third party to handle the payment and settle the accounts of buyers and sellers. For example, PayPal works and is accepted by users precisely because they trust the third party to avoid double-spending (Chiu and Koeppl, 2017).

This undesirable feature does not arise when electronic payment systems are based on a central authority. This defect is solved if an authority (usually a bank) steps in, verifies that payments are legitimate, tracks the movement of money, and manages the accounts of the buyer and seller. The buyer initiates a payment by placing an order, the

[1]"Double-spending is a potential flaw in a virtual cash system where the same digital currency can be spent more than once when there is a lack of transaction history records, which allows for the digital file to be duplicated or forged" (Cong & Heng, 2018).

central authority ensures that the buyer has the necessary funds and adjusts balances, accordingly, as can be seen in *Figure 1.3* below.

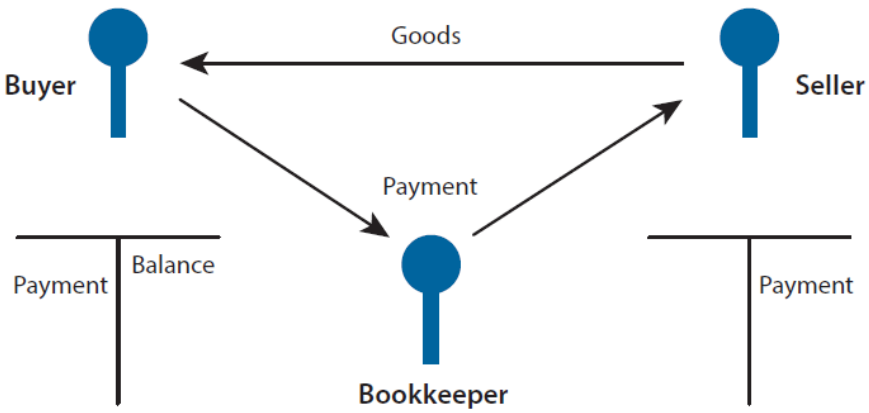


Figure 1.3 - Payment system with a central authority

This centralized system requires the intervention of a third party that the agents must trust, and in this way, the problem of double-spending is solved. However, centralized systems are vulnerable to hackers, who would have a well-defined target to hit and the bank, if not sufficiently protected, could allow access to sensitive information. The result would be very serious, as the trust placed in the third party would no longer be valid and transactions no longer deemed secure.

Cryptocurrencies like bitcoin[2] go a step further and remove the need for a third party. Thanks to a distributed network, they eliminate both the risk of "double-spending" and the risk of there being a single vulnerable target. However, before seeing in detail how these currencies work, it is necessary to introduce them.

[2] Note that, in this work, Bitcoin with the upper case letter indicates the system and the network, while the lower case letter indicates the cryptocurrency.

1.2 Cryptocurrencies and cryptography

Thanks to the development of the internet and technology, alternative payment systems have been developed. It is at this point that cryptocurrencies, also called virtual or digital currencies, were born. Their name derives from the fact that they are not issued by any recognized institution (such as the Central Bank), but are issued, thanks to technology and cryptography, by individuals operating on the web.

By definition, a cryptocurrency, or virtual currency, is "the digital representation of value, not issued by a central bank or public authority, not necessarily linked to a legal tender, used as a means of exchange for the purchase of goods and services and transferred, stored and traded electronically"[3].

An important role in the world of cryptocurrencies, as the name reminds us, is played by cryptography. It is used for various purposes such as signing transactions, verifying them, controlling money creation, and guaranteeing users privacy. Cryptography is defined as "the technique of representation of a message in a form such that the information contained in it can be received only by the addressee; this can be achieved by two different methods: concealing the very existence of the message or subjecting the text of the message to transformations that make it incomprehensible".[4]

Cryptocurrencies were created in conjunction with the financial crisis of 2007-2009, as a "nonconformist" instrument opposed to common currencies. Taking the euro as an example, each euro in circulation is printed by the European Central Bank (ECB), which decides to increase or decrease the amount of money in circulation according to the economic stage. Cryptocurrencies go exactly in the opposite direction: a decentralized system where no authority establishes the amount of money that needs to be in

[3]Directive (EU) 2018/843 of the European Parliament and of the Council of 30 May 2018, amending Directive (EU) 2015/849 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing, and amending Directives 2009/138/EC and 2013/36/EU.

[4]https://www.treccani.it/enciclopedia/crittografia/#:~:text=crittografia%20Tecnica%20di%20rappresentazione%20di,a%20trasformazioni%20che%20lo%20rendano

circulation and then introduced just as an instrument to "counter" the total control over the currency in circulation by the state. Therefore, a sort of "parallel economy" is born, where everyone can carry out their transactions in total freedom, without the need for a central authority and/or some intermediary.

This lack of central authorities could suggest a possible problem linked to "double-spending" cited before. The drawback is solved, even without the need for a central authority intervention, thanks to the system on which cryptocurrencies are based: a distributed network where transactions take place peer-to-peer[5] without a central authority, where everyone can access and interact directly with all the other nodes[6] in the network. Each transaction is verified and validated by each member of the network and is time-stamped and recorded in an "ongoing chain", called blockchain, to have certain evidence that all transactions made in the past are recorded.

Cryptocurrencies do not exist in physical form because they are digital. They're held in digital/electronic wallets, and it is possible to exchange them via the internet between users or buy goods and services in certain businesses that accept them as a means of payment. They are an asset to the person holding them without being the liability to another entity. To clarify the latter aspect, the money that a person holds in the checking account is an asset, a claim of the owner and, at the same time, a liability for the bank, because it is a debt that the bank agrees to convert into cash in case it is demanded. In this respect, cryptocurrencies resemble gold.

The first cryptocurrency that introduced important innovations and that started the development of this market is bitcoin, the most famous and widespread virtual and

[5] Peer-to-peer systems also referred to as P2P, are systems that allow each user to interact directly with others. In the specific case of cryptocurrencies, the network is built in just this way, without the need for any bank to be present as a third party (unlike, for example, Visa or MasterCard which are centralized systems). Only cryptography allows these decentralized systems to function.

[6] It is a computer network in which there is no difference between the nodes (a node in technical jargon is nothing more than a user or a computer on the platform). In other words, all nodes are considered equivalent (peer means equal) and each node can perform a transaction.

decentralized currency at the moment. Due to its fame, bitcoin is used as a reference point when discussing cryptocurrencies, including in this paper.

1.2.1 Hash function

As mentioned in the previous section, an important role in cryptocurrencies is played by cryptography, where hash functions are used to transform messages into encrypted messages so that they cannot be read by third parties.

A hash function is a non-invertible function that transforms an alphanumeric string of any length (input) into a string, also alphanumeric, of a predetermined length (depending on the type of hash function used), which is called “digest”. If the input string is unchanged, the hash function will always produce the same output. However, if any part of the input is changed even marginally (a letter being changed from a lowercase character to an uppercase one) the code will change to a new, completely different hash. Bitcoin uses the SHA-256 hash function, the most secure cryptographic hash function available at the time, which transforms input messages into a digest consisting of 256 bits.

Figure 1.4 - SHA256 hash function converting three different messages

SHA-256 has an important role in Bitcoin's blockchain, which is to ensure the correctness of transactions. As will be explained in the next chapter, during each transaction the hash of the previous transaction is digitally signed, so who receives the payment can check the various transfers of ownership of the transferred currency by verifying the

signatures on each transaction. Changing a transaction in this way is not possible, as the hashes would no longer match.

Another fundamental use of cryptographic hash functions in the Bitcoin system is in the generation of new coins. In fact, bitcoins are continuously generated and are given to whoever manages to solve the mathematical problem of calculating a counter-image, described above.

1.2.2 Public-key cryptography

Every transaction recorded on the blockchain is encrypted and only the recipient can decrypt it. In this way, the blockchain does not need to implement particular security systems to "protect" data, as data are rendered indecipherable by all those who are not authorized to do so. This is possible by using asymmetric encryption.

Asymmetric encryption is based on the use of a key pair: a public key and a private key. The key pair is mathematically linked by a function, which ensures that a message encrypted with one of the two keys can only be decrypted by the other.

Here is an example to better understand it. There are two people: A and B. A wants to send a text document to B and wants to make sure that only B can read the content of that document. A then decides to use asymmetric encryption, so he uses B's public key to encrypt his message (A knows B's public key because, being public, B has made it available to A). The encrypted document is no longer decipherable by A, as it does not have the private key of B (this, unlike the public key, is private: only B possess it). B receives the document and can decrypt it using its private key.

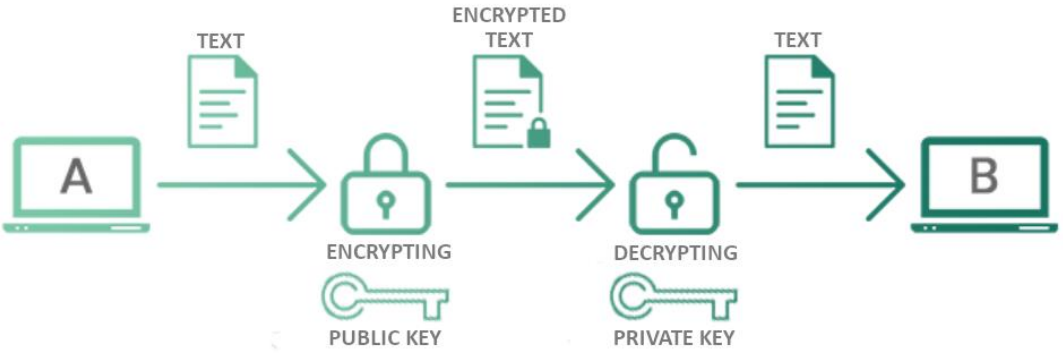


Figure 1.5: Steps involved in public-key cryptography

Once B has received and opened the document, he decides to reply to A using the same encryption method, so that only he can read the contents of his document. It then encrypts the document using A's public key. A receives the document and manages to decrypt it using its private key.

Blockchain uses asymmetric encryption to allow the exchange of goods (such as cryptocurrencies) between one person and another. Every person who owns an asset (of any kind) on the blockchain has a public key (also known as "address") and a private key. To make it easier, here's an example. Think about two identities with two public keys:

- John - address: 0x1234567890
- Simon - address: 0x5678901234

John and Simon are identified within the blockchain with an address, this is public and anyone who wants to send them goods must send them to their respective addresses. Each of them has their private key, which guarantees that the sending of any goods is wanted by the owner of those goods (just as if it were the pin of the credit card), since only those who have the private key can carry out the transfer of an asset. Of course, anyone else in possession of the private key will be able to perform any transaction (just as if someone knew the credentials to access your on-banking), so you need to keep your private keys in secure locations to limit the risk of being robbed. Let's see in detail what is the procedure that would happen if John were to send one bitcoin (BTC) to

Simon. John accesses his assets using his private key; at this point he transfers one bitcoin to Simon's address 0x5678901234, and this is where asymmetric encryption comes into play, as the transaction is encrypted using Simon's public key:

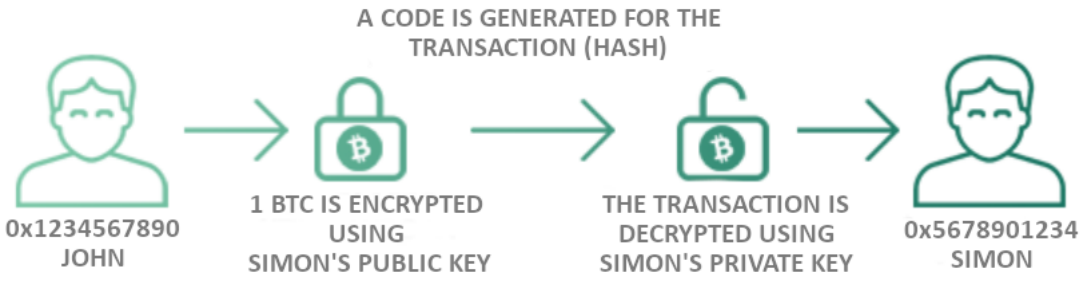


Figure 1.6: Example of a transaction

The transaction takes place securely and legitimately, since John sent the transaction (which is authorized by John's private key) to Simon's address, while only Simon can decrypt the transaction using his private key.

1.2.3 Where are cryptocurrencies stored?

Commonly on the net, it is said that cryptocurrencies are stored inside special wallets, but this is not correct. Without getting into too many technicalities, cryptocurrencies reside on the blockchain, while wallets keep track of all the transactions that take place within the blockchain. A wallet is therefore a tool that we can use to communicate directly with the blockchain, to send and receive cryptocurrencies.

Opening a wallet is simple and anyone can do it, using special software rather than hardware, or by creating a wallet manually (paper-wallet). The usefulness of this tool is comparable to that of a common bank account, with the difference that the blockchain does not ask us for the tax code or other identifying information to open one.

When a wallet is opened, the owner becomes aware of a public key and a private key. Through the public key, it is possible to receive transactions, while through the private key it is possible to unlock the wallet and carry out any transaction. The private key, metaphorically speaking, represents the "signature" of the wallet owner, so anyone in possession of the wallet could "sign" (authorize) any type of transaction as if they were the rightful owner. In the eyes of everyone, transactions signed in this way would appear to be legitimate transactions. Precisely because of the above, it is important to keep the private key of your wallet in a safe place.

1.3 Are cryptocurrencies really a currency?

Cryptocurrencies, as stated in the previous section, are not currently a currency from a legal point of view, because they are not validly recognised by law for the fulfilment of payment obligations, are not issued by a central body and there is no obligation for the recipient to accept them. In addition, to be considered a currency, they must comply with certain characteristics.

By definition of the European Central Bank "Money, in any form, fulfils three different functions. It is a means of exchange, a means of payment with a value that everyone trusts. Money is also a unit of account that allows the attribution of a price to goods and services. And it is also a store of value."[7]

Unit of account: a primary function traditionally attributed to money is that of being a unit of account. That is to say: the object-currency is a common metric for measuring the value of commercial transactions, for example. This is the function that historians consider to be the oldest. Since cryptocurrencies have frequent fluctuations in value resulting from demand and supply, it is difficult to use them as a unit of account, as a standard numerical unit to measure the value of goods and services to compare them with one another. Moreover, since the value of a single cryptocurrency is relatively high

[7] https://www.ecb.europa.eu/explainers/tell-me-more/html/what_is_money.it.html

compared to the value of many goods (such as food or necessities), in estimating the price difference between them many decimals should be used, making comparisons more complicated.

Means of exchange: the second function of currency is to be a medium of exchange. "It consists in the possibility of accepting an object in exchange for another, with the expectation/confidence of being able to use the same object in other exchanges.". Cryptocurrencies are a means of exchange but only in an intangible dimension, they have no intrinsic value and very few people accept them as a means of payment. In addition, obtaining bitcoin is possible if you are a miner (we will see it later) who owns many powerful computers to produce them independently, otherwise, you have to buy them online from specialized secure sites. For cryptocurrencies, there is no dispute resolution, so fraud cannot be refunded, and this situation is worsened by the fact that users are protected by pseudonyms and encryption, which make users difficult to identify.

Store of value: a currency is defined as a value reserve when the owner holds money, physically at home or in a bank, at a certain point in time, to exchange it in the future for goods and services, assuming that the currency maintains its value and the price of the goods does not change excessively over time. Cryptocurrencies are highly volatile, their value is unstable and changes frequently over time compared to official currencies, or less volatile assets such as gold, which makes them unsuitable for saving, but, at the moment, are more likely speculative assets to invest in.

Figure 1.7: Gold price (oz) in USD, from April 2013

Figure 1.8: Bitcoin price in USD, from April 2013

Figure 1.7 shows the evolution of the price of an ounce of gold, while *Figure 1.8* shows bitcoin's price evolution over time, both are represented by taking a period of time that goes from April 2013 onwards. As can be seen, the price of gold shows a fairly regular pattern with slight price variations from one year to the next, while in the case of bitcoin's price, from 2017, when the cryptocurrency market started to be better known, and especially from the beginning of the COVID-19 pandemic in 2020, market variations can be seen even from month to month over the years.

Gold is considered a haven asset, meaning an asset that tends not to lose value over time due to events in the market. Bitcoin, instead, is still seen by some people as a speculative investment rather than currency and store of value, as its price is very volatile.

1.4 An overview on cryptocurrencies regulation

Cryptocurrencies have gained enormous success in recent years, not remaining a niche phenomenon cultivated only by financial experts, but also entering the lives of ordinary citizens and gaining explosive popularity. In fact, in the last year, virtual currencies have grown exponentially and quickly reached record levels. A striking example is Bitcoin, which in a very short time reached a peak of $64,863 in April 2021.

Governments have taken, or are taking, a wide range of very different approaches to the phenomenon. Many of them have only recently begun to take a stand, although in some countries the legislation is still poor at the moment, with many issues left unresolved.

However, as the phenomenon developed, several important questions have arisen. All countries will sooner or later have to express themselves. So far, the focus is mainly on three crucial elements. First and foremost, there is the fundamental issue of consumer protection[8], who may have little knowledge of these highly volatile instruments and the risks associated with them they could incur by investing money in them. The challenge is not easy, especially when you think that cryptocurrencies are designed to exist outside any form of centralised control, which means that regulation could easily be ignored by anyone with an internet connection.

To date, the solution for most governments has been simply to warn consumers about the risks associated with cryptocurrencies, exchanges and ICOs[9], rather than trying to impose a ban that is difficult to enforce. Other countries have instead legislated to make the use of cryptocurrencies illegal within their borders.

Second, there is another important challenge related to how governments can act to stop or minimize the use of cryptocurrencies to fund illegal activities, such as money laundering and terrorist financing. The issue is indeed thorny since there is a major problem related to the fact that in the network cryptocurrency transactions are anonymous, they can easily cross the borders of a country, making it difficult for the latter to supervise and identify the users involved in buying or selling on the web, among whom could hide criminals or terrorist organizations. In this regard, many governments have yet to address the issue, while other countries have extended money laundering laws to exchanges and entities that come in contact with and operate cryptocurrencies,

[8] Think of the phenomenon of centralized exchange hacking, where millions of cryptocurrency assets have been stolen, or the failure of the exchanges themselves resulting in the loss of all consumers funds.

[9] ICOs are a form of financing used by start-ups or those who want to carry out a particular project, made possible by technology. To obtain financing a project is proposed to the public (usually via a "whitepaper"), which will be realized via Blockchain with the creation of "tokens" to be transferred, against a fee, to the funders.

requiring them to conduct thorough user checks and report suspicious transactions. They have also extended money laundering laws to cryptocurrency exchanges and businesses that deal with cryptocurrencies, requiring them to thoroughly vet users and report suspicious transactions.

Finally, there is also the tax issue, with governments busy figuring out how to classify these currencies for tax purposes. Again, the approach is very different. There are many differences in how cryptocurrencies are classified for tax purposes, with some governments defining them as foreign currencies, others as financial assets, others as securities, and others as commodities. This means that, depending on the jurisdiction, virtual currencies may be subject to VAT, income tax, capital gains tax, etc.

As mentioned above, the dimensions of the phenomenon are large and it is precisely for this reason that each nation has posed, or is posing, the problem of how to deal with it, without finding a common vision or position. As we will see later, on one side some countries welcome cryptocurrencies with open arms seeing in the technology an enormous innovative potential, while on the other side there are governments that have adopted a very hard line due to the risks related to virtual currencies and fearing repercussions on the stability of the financial system. In between are many nuances. Some jurisdictions have recently regulated the phenomenon to some extent or are in the process of issuing new regulations, while in others the position is still not clear.

Speaking specifically about bitcoin, in some countries the currency is not allowed, in others, it can be used as a legally accepted method of payment and, in the only case in the world of El Salvador, as of September 7, 2021, be legal tender.

The panorama is, therefore, very diversified, dynamic and constantly evolving, and the approach of the countries is anything but well-defined. Furthermore, it should be kept in mind that the regulatory framework that seeks to regulate and govern the phenomenon could change very rapidly, given the delicate phase in which everyone is studying and testing possible responses. Taking the end of September 2021, as a reference, however, it is possible to take a global snapshot to take stock of the situation and see what developments there have been so far (if any) at the regulatory and normative level in the cryptocurrency sphere.

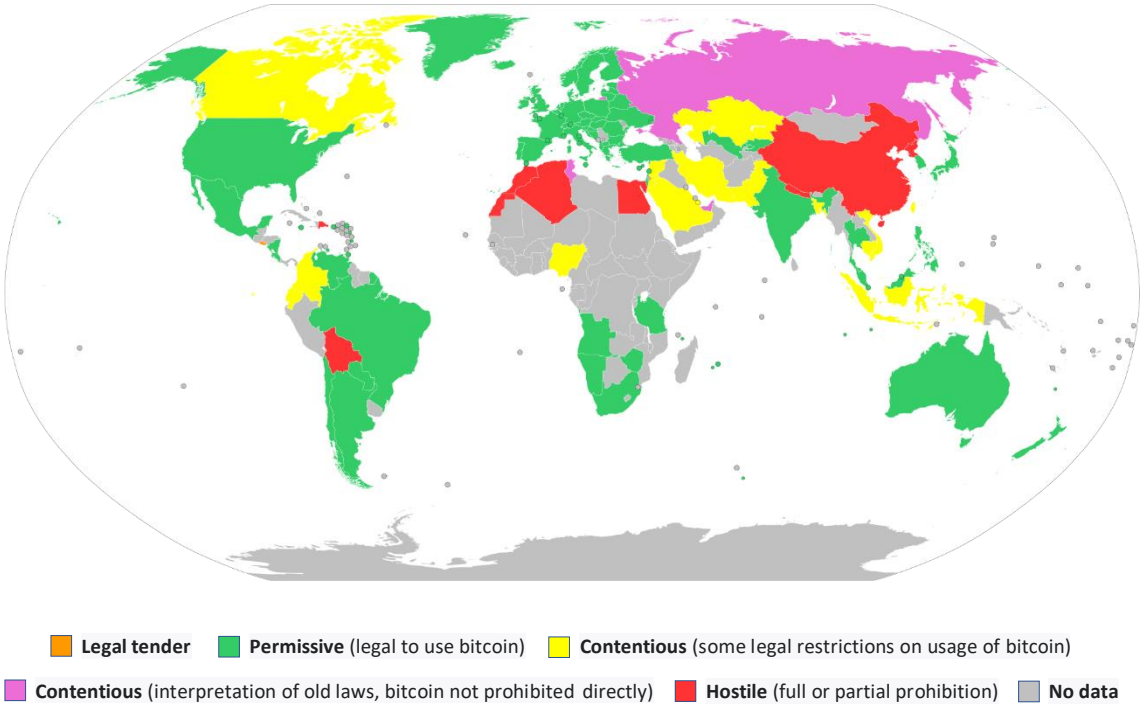


Figure 1.9: legal status of bitcoin (and related other instruments)

It is worth reflecting on what can be seen in *Figure 1.9*, which shows how some jurisdictions impose restrictions on cryptocurrency investments, although in different ways. In particular, the red colour represents the countries characterized by an absolute ban, such as China, Algeria, Bolivia etc., which prohibit all activities that have to do with cryptocurrencies. Yellow and violet represent countries with an implicit ban that, while not prohibiting their citizens from investing in cryptocurrencies, impose indirect restrictions by preventing financial institutions within their borders from facilitating transactions involving virtual currency and participating in this market. In other words, in these territories, legislation has been enacted and makes it difficult to access the cryptocurrency market. This is the case for example of the USA, Russia, Iran, etc.

2. BITCOIN AND THE BLOCKCHAIN

2.1 History of bitcoin

This paper focuses on bitcoin, being the main cryptocurrency and the subject of empirical analysis, but these concepts can also be extended to the larger universe of altcoins, which are all those cryptocurrencies alternative to bitcoin. This is also because many altcoins are descended from bitcoin and are based on its protocol.

As explained in *Chapter 1*, bitcoin is the first cryptocurrency that has introduced important innovations in the payment system and that started the development of this market. The importance of bitcoin is related to the fact that, with a market capitalization of over 800 billion dollars, it is the first virtual currency by market capitalization, with a dominance of over 40% on the total cryptocurrency market capitalization, according to the data provided by *coinmarketcap.com* at the time of September 2021.

Bitcoin was born precisely in 2008 when on October 31st, Satoshi Nakamoto, an anonymous programmer whose true identity nobody knows yet, published the white paper introducing it. In the abstract of that document titled "Bitcoin: A Peer-to-Peer Electronic Cash System", the author argues that "a fully peer-to-peer version of electronic money allows you to send online payments directly from one place to another without going through a financial institution". He also argues that "digital signatures are the solution, but the main benefits would be lost if a third party were still needed to avoid the "double-spending" problem". We propose a solution to the problem of "double-spending" using a peer-to-peer network. The network records transactions by entering them into a continuous chain based on the Proof-of-Work[10], creating data that cannot be changed without redoing the Proof-of-Work. This chain is clear proof of the sequence of events" (Nakamoto, 2008).

[10]The Proof-of-Work, or PoW, is the most common consent model and is an algorithm that confirms the transactions that have taken place and create a new block unambiguously so that a consensus can be reached between all the nodes when it is added to the chain.

Nakamoto, therefore, to overcome the trust problems related to traditional systems of payment and high brokerage costs, introduces bitcoin, explaining that encryption allows two parties to interface directly while protecting users from fraud.

There had been attempts in the past to introduce digital currencies, but they did not solve the problem of “double-spending” properly, there was no mechanism to prevent the currency holder from using it in multiple payments. This problem was solved by Nakamoto with the introduction of an "ongoing chain", as mentioned in section 1.2. The latter is the most important element of the system. It is also known as the blockchain and it is first of all the technology behind bitcoin, but generally also behind all other cryptocurrencies, and allows them to work. It will be discussed in the next section.

In addition, Bitcoin is open-source and its design is public, nobody owns or controls Bitcoin and everyone can take part in the project.

2.2 The blockchain

The blockchain is a public ledger, distributed and shared across devices that are part of Bitcoin's peer-to-peer network, and it is composed of a chain of chronologically ordered blocks.

A block is comparable to a page in a ledger, and it is a file containing data from a series of transactions. All transactions are grouped into blocks that make up the blockchain, and new transactions are placed into a new block that is submitted for review and approval by blockchain participants. Once approved, the block joins the chain, permanently. Blocks form a chain because each block, in addition to containing transactions and other data, has a reference from the previous block, so all blocks, from the very first one, are linked together and it is almost impossible for a block to be modified, especially at the beginning of the chain, since the modification of one block would result in the modification of all subsequent blocks.

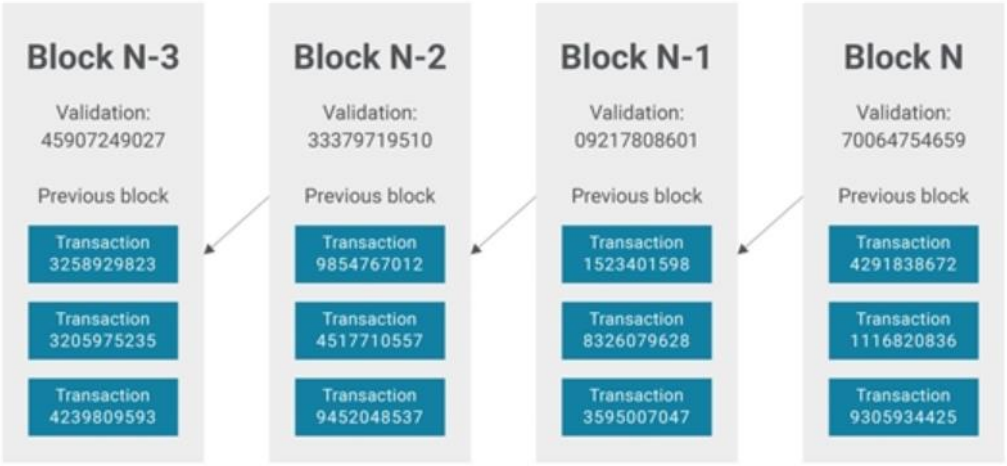


Figure 2.1: A generic blockchain. It shows blocks that contain multiple transactions, pointers to previous blocks in the chain, and data used to validate the block.

The blockchain is a distributed ledger, in the sense that this ledger is not stored in a central point but is present at the same time on all devices connected to the network, each of which is perfectly synchronized on the same documents: each user has a copy, and you can also view it online, in some sites, without downloading software.

The advantages of this system are numerous, the power is distributed among the network users[11] because participation is required to approve blocks, as we will see later. It is not necessary to have a central authority that manages the supply of currency, as its creation is linked to an algorithm, based on a cryptographic system to ensure the security of transactions and privacy. In addition, since there is no central entity, the costs to carry out transactions are reduced and the risk of the centralization of power in a single individual is an unlikely event in practice.

There is trust in the network. The system's reliability and security are based on users consent, whose privacy is guaranteed thanks to pseudonymity and, at the same time, enjoy transparency and traceability because each user can view all transactions

[11] Note that to use bitcoin you do not need to be a node in the Bitcoin network. Nodes allow the network to function and make transactions possible, but whoever uses bitcoin to pay for goods or services (or to send money to other users) is not obliged to participate in the network actively, that is, to be a node and thus have an always up-to-date copy of the ledger. To this subject, it is enough just to create an address to be able to perform these operations.

recorded and those under validation. It is a robust system, thanks to cryptography and irrevocability of transactions, which incentivizes users to participate in the confirmation of transactions and validation of blocks because you are rewarded with newly minted bitcoins along with the commissions included in the transactions.
We talk about the “immutability” of blockchain technology, and DeRose (2015) argues that the concept of immutability, or resistance to tampering, is what gives intrinsic value to cryptocurrencies because of a revolutionary feature, namely “the ability to declare the truth, globally and without a central authority, regardless of what anyone else does to change that truth. On this resistance to tampering rests the value of a bitcoin.”.
Some characteristics might indeed be different depending on the type of blockchain but not immutability, which is what contributes to the reliability of transactions, and which ultimately makes a cryptocurrency tradable.

2.2.1 How a transaction takes place

As already mentioned in Section 1.2.2, the system is based on cryptography and that is what makes it secure. The public key encrypts outgoing messages and decrypts its signature, while the private key decrypts incoming messages and signs outgoing messages so that the recipient is sure where the message came from and that it has not been modified. Only the public key is needed to receive, while both keys are needed to send.

Example: A wants to make a transaction to B.
1. A creates the transaction (containing exchange information, price, affordability) and applies a hash function to it, thus creating the digest.

2. A signs the digest with his private key and adds B's public key.

3. A sends the transaction to the other nodes in the network for verification of A's availability of the amount of cryptocurrency to be sent through analysis of the history of transfers already made.

4. The nodes begin the construction of a block, in which the transaction and other subsequent ones are recorded, which will then undergo mining, its validation. Once the block is validated, it is added to the blockchain.

5. B receives the transaction, with the public key of A decrypts the digital signature and acquires the digest.

6. since B's public key is now inserted in the transaction, only B through his private key can unlock the availability and start new transactions by repeating the process.

By transferring the ownership of cryptocurrencies, with the digital signature of the previous transaction and adding the public key of the new recipient each time, a chain of possession is created: this mode cancels the possibility for a user to spend the same amount of bitcoin twice.

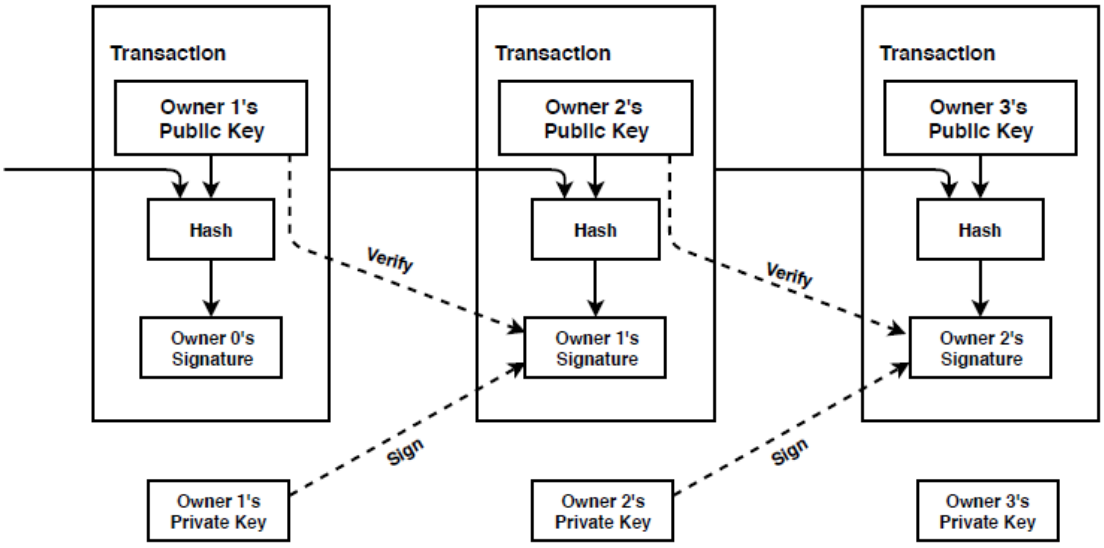


Figure 2.2: Transaction between users of the Bitcoin network

2.3 Miners and mining

While transactions are made by users, miners are the ones who validate transactions, verify that the user has the bitcoins he wants to spend (i.e. that he has not spent them before) and thanks to their activity the security level of the network is increased. They have the task of creating new blocks, which are not automatically added to the blockchain but are published through a process called mining[12], performed by miners or mining pools. More precisely, after a node initiates a transaction, it is reviewed and validated by other nodes and then waits in a queue with other pending transactions until a mining node validates a group of pending transactions and adds them to a block; then it publishes it on the blockchain.

Miners, thanks to their computational resources, solve a mathematical problem of high difficulty (or cryptographic puzzle). The problem consists in tracing, through attempts, the original content that has been encrypted through a hash function (see Section 1.2.1). This process, competitive and adopted by most platforms, represents the Proof-of-Work underlying the Bitcoin system, the distributed consensus mechanism.

2.3.1 Block structure

As you can see in the figure below, a block consists of a header and a body. The header mainly consists of:

- hash from previous block: the alphanumeric character set referring to the previous block.
- metadata: composed of the block version number, timestamp and difficulty; the block version number is a number to track software/protocol upgrades, the

[12] To be precise, the term mining is used for blockchain implementations that use the Proof-of-Work consensus model, as in the case of the Bitcoin network. The name of the process for publication of new blocks varies depending on the consensus model, but for simplicity, we use the term mining regardless of the consensus model used

timestamp is the approximate creation time of this block and the difficulty is the proof-of-work algorithm difficulty target for this block.

- nonce (or “number only used once”): a unique number added to the block encrypted by the hash.
- merkel root: the hash of all hashes of all transactions in the block.

The body, on the other hand, contains the transaction data for the block.

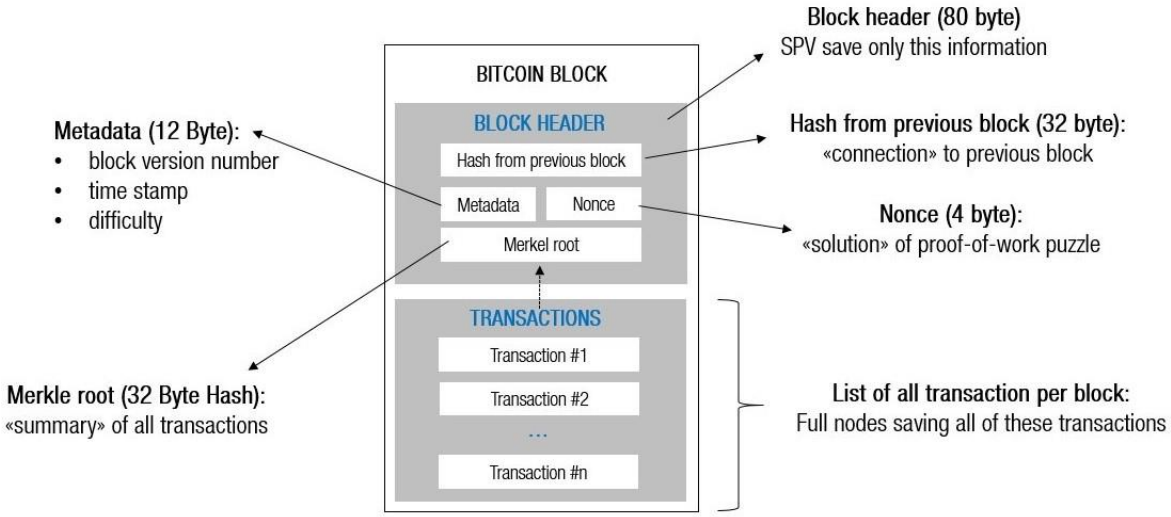


Figure 2.3: structure of a blockchain block

2.3.2 Mining process

Once connected to the network, the miner can begin the mining process. The miner, after requesting the blockchain history, collects some pending transactions and makes sure they are valid by checking digital signatures and that users have not already spent the bitcoins. Once this is done, the miner can start building its block, a candidate for the network as the next in the chain. To build it, the miner groups and then add some transactions that he had verified, within the size limits of the block. Once built, he has to solve it, because he has to calculate the last brick of the block, he has to find a nonce that makes it valid.

The problem to solve is to find the value of the nonce such that when the hash function is applied to the block header, the output of the function must be a number that is less than or equal to a target value, which changes over time, based on the current difficulty.

$$hash\text{-}f\ (header) \leq targetvalue$$

A hypothetical nonce value is put into the function, which is incremented by one until the condition is satisfied whereby the output of the function is less than or equal to the target.
If this condition is true, the block is valid and the hash value uniquely and securely identifies each block as if it were a fingerprint, without the possibility of tracing the generated content. Moreover, the fact that each block contains the reference to the hash of the previous block creates the chain data structure and allows us to verify that its value, which remains unknown, has not changed. Below is an example to better understand.

Example:

- header = (Hash from previous block, Metadata, Merkel root) = “Hello, world”
- nonce = unknown
- target value = 000d4fg5t698...
- hash function used: SHA256

The miners already know the other elements of the block header, apart from the nonce. The goal is to find a value for the nonce such that:

$$SHA256\ (\text{“Hello, world”}, nonce) \leq 000d4fg5t698...$$

To find the nonce the miners spend a lot of time calculating: they insert in the function a series of values in place of the nonce and calculate the hash function each time.
The series of values to insert in place of the nonce starts from number zero, the hash function is calculated and if the result of the function is less than or equal to the target,

it means that the value of the nonce is correct and the solution has been found. Otherwise, the value of the nonce is incremented by one, then the hash function is recalculated and so on until a correct value is found.

SHA256("Hello, world", 0) =
1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64

SHA256("Hello, world", 1) =
e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8

...

SHA256("Hello, world", 4250) =
0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9

So the calculator has increased the value of the nonce 4250 times to reach the target: the digest found with the nonce 4250 starts with four zeros and is, therefore, less than the target value, because the latter starts with only three zeros.
The nonce is then transmitted to the other network nodes that check its correctness: if the result is negative, the block is rejected, while if it is positive, the lock is added to the chain which updates on all devices.
Miners, or mining pools[13], who contribute their resources to the system by providing computer power, also called hash rate[14], are rewarded for the important service they offer and this is how the Bitcoin protocol issues money.
This reward is provided to miners because finding the correct hash (and thus recording a transaction) requires a lot of effort and energy, so it is a very costly activity.
The miner who first provided the solution to the problem, and then validated the block, receives a reward of 6.25 bitcoins (from May 2020) and also, if the block transactions contain commissions, the miner is entitled to receive them as an additional reward.

[13] https://www.investopedia.com/terms/m/mining-pool.asp

[14] "Hashrate is a measure of the computational power per second used when mining. More simply, it is the speed of mining. It is measured in units of hash/second, meaning how many calculations per second can be performed."

2.3.3 Network difficulty

Initially, mining could be done by any computer or network user, but then more and more people began to exploit this process to make profits, making it more difficult to find the hash to close the block. Precisely because the number of users who want to solve these puzzles is increasing, the puzzles are becoming more and more complicated. The difficulty is the same for all miners, and it's the value that indicates how hard it is to find a hash less than or equal to the target value and regulates how long it takes miners to mine the next block.

To offset the increase in network computing capacity, the difficulty increases every 2016 block, i.e. every two weeks, so that block validation always takes about ten minutes on average. The difficulty increases as the size of the blockchain increases in relation to the activity in the market, for example concerning the number of new miners, or the more efficient hardware that allow a block to be solved faster.

Within each block, there is stored a compressed representation of the target when the block is created, which corresponds to the difficulty field (see *Figure 2.3*).

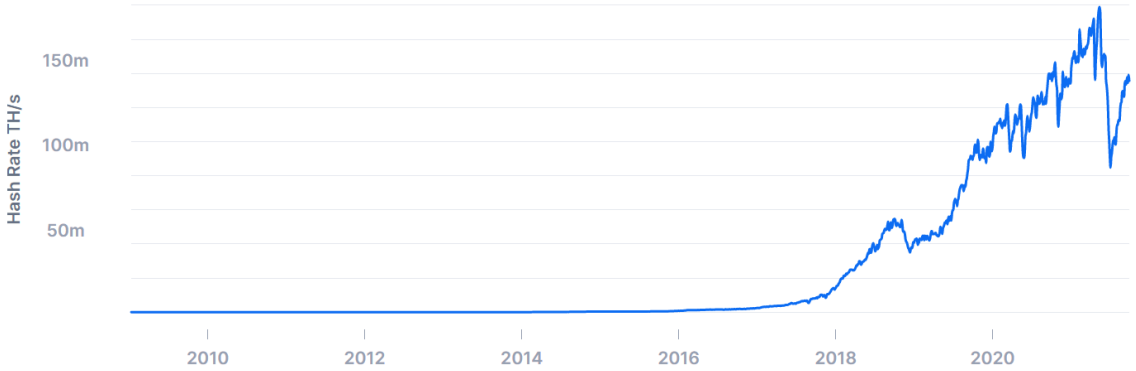


Figure 2.4: Total Hash Rate (TH/s) from 2010

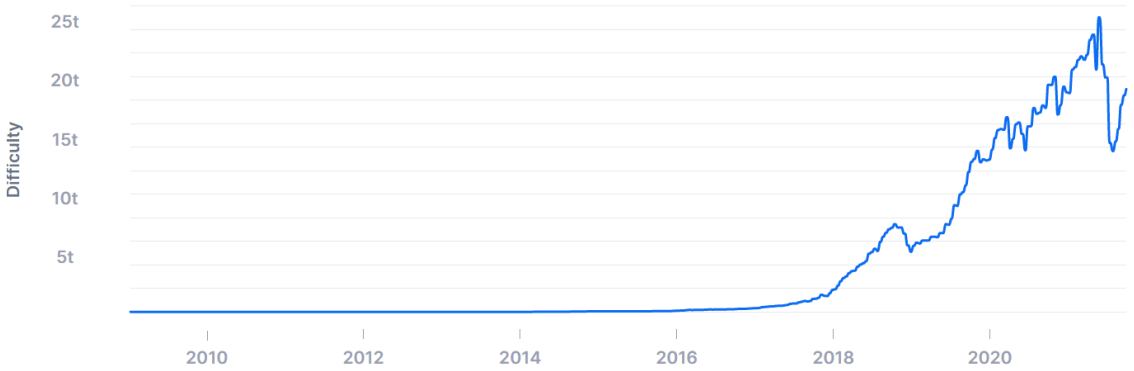


Figure 2.5: Network Difficulty from 2010

Figure 2.4 shows the total hash rate, i.e. the computing power of the Bitcoin network, while *Figure 2.5* shows the level of the network difficulty in mining new blocks, both from 2010.

The two charts show an almost identical pattern over time because if the global hash rate increases, the global difficulty increases accordingly, and if the difficulty increases, the miners will have to increase the computing power to be able to mine the blocks.

2.4 Proof-of-Work consensus mechanism

As explained in the previous sections, mining calculations require a considerable amount of work to be solved and, therefore, different types of costs have to be incurred.

It is a system that encourages users to participate, miners are competing with each other to solve the problem because the first one providing a solution to the calculations is rewarded with new bitcoins.

It is defined as a Proof-of-Work (PoW) system, so named because adding a block to the chain requires a large amount of work (mining), which produces specific data (the proof, or solution to the cryptographic problem) to verify that a large amount of work has been done.

Bitcoin’s Proof-of-Work is a distributed consensus algorithm because with this system a decentralized agreement is reached between the nodes for the addition of new blocks, without the need for a central body to act as a controller.

The PoW system has both advantages and disadvantages. The advantages are first that the number of coins owned by miners or mining nodes does not affect the mining process, and second the system protects from attacks on the network and ensures that transactions cannot be changed, as this would require the malicious party to have computing power that can compete with the entire network. The disadvantages are that PoW is expensive because it requires high computational resources, time, electricity and the computers used must be updated over time to compensate the increase in mining difficulty.

2.4.1 The energy consumption problem

The main problem related to the PoW system, which has emerged and is increasingly being debated in a particular way in 2021, is the energy consumption issue. From an environmental point of view, the PoW system has quite a negative impact. Unfortunately, it is not possible to know exactly how much energy Bitcoin consumes, because each miner has its equipment, which can be more or less efficient depending on the machines used for mining or the cooling system, which also consumes energy.

According to Digiconomist, Bitcoin's energy consumption more than doubled from January to September 2021, from nearly 78 to over 160 Terawatt-hours of energy consumed[15]. That's a number that varies over time, and currently, if we were to consider Bitcoin as a nation, it would be ranked 24th most energy-consuming country in the world, slightly less than countries like Thailand and more than countries like Poland.

We can also think about comparing a Bitcoin transaction with a VISA card transaction, and, according to Statista's estimates as of September 2021, the energy consumed by a Bitcoin transaction is equivalent to several hundred thousand VISA card transactions: a

[15] https://digiconomist.net/bitcoin-energy-consumption

Bitcoin transaction consumes 1,810.74 kilowatt-hours of energy, while 100,000 VISA transactions consume 148.63 kilowatt-hours[16].

The comparison is scary, even if for the bank transaction many other costs of different nature have not been considered, such as heating oil, gasoline consumed by employees, the material used such as paper, plastic containers, stationery, IT equipment, waste, etc. So, considering the number of banks in the world, Bitcoin probably has lower consumption since it doesn't have infrastructure like banking and that the number of users is lower than bank users.

Clearly, as the difficulty increases, more computational power will be necessary to solve the blocks and consequently, more electricity will be needed to run the machines, for cooling and other services.

In the studies mentioned before, there is only an estimate of energy consumption and no reference to how the energy is produced. Bitcoin and other cryptocurrencies certainly require a lot of energy, but it is important to broaden the comparison by looking at the issue from multiple points of view, considering for example all those online and offline services, that can be useful and less useful, used by users every day and that also consume a lot of energy. In addition, it is not only important how much energy is consumed, but how much of it has been produced by processes that do not impact the environment, such as renewable energy.

2.4.2 51% Attack

The risk in this type of system is the so-called 51% attack, a situation in which one or more users have accumulated more computing power, expressed in hash rate (see *Section 2.3.2*), equal to at least 51% of the power compared to the other members present in the Bitcoin network.

[16] https://www.statista.com/statistics/881541/bitcoin-energy-consumption-transaction-comparison-visa/

In this case, the attacker, having the majority of the computing power, would have control over the blockchain and could intentionally exclude transactions by preventing confirmation or modifying the order: for example, he could reverse some personal transactions made, thus creating situations where it would potentially be possible to spend the same bitcoins twice ("double-spending"). So, he could build the blocks as he pleases, monopolizing the mining and receiving the rewards.

However, this is a difficult situation to implement, as modifying previously confirmed blocks becomes increasingly difficult as the size of the blockchain increases because the blocks are linked together and only the last blocks would be modifiable. In addition, the larger the network, the more protected it is from this type of attack thanks to the large number of miners competing with each other to solve the blocks.

In any case, it would be very expensive, from the point of view of hardware resources, to have such computing power available in such a vast global network.

2.4.3 Proof-of-Stake: an alternative

The Proof-of-Stake (PoS) system is based on the concept that users with the most participation, that is the amount of cryptocurrency the individual has on the platform, is the key determinant used by the system to decide which entity has priority in adding new blocks to the chain. The distributed consensus algorithm is called forging (or minting), the block is forged instead of being mined and the person who validates transactions and forges new blocks is called forger (or validator).

For example, if a user owns 10% of the total coins in the network, he has a 10% chance of being chosen by an algorithm, which randomly chooses between validators, to forge a new block. The validator selected is responsible for verifying that the transactions are valid, signing the block and adding it to the chain, and then receiving the commissions associated with each transaction contained in the block as a reward. The basic assumption behind this system is that users with the most involvement in the platform will certainly have a strong desire for the platform to succeed and will therefore make decisions in their best interests. Thus, in the case of PoS, the computational power is not

important, but the amount of currency held by the validator is. As a result, it requires less electricity and is, therefore, more environmentally friendly than PoW.

One disadvantage of this system is that the richer network members become increasingly wealthier because, being easier to mine, they solve more blocks and receive more rewards. This problem, however, is solved by the fact that the PoS algorithms of cryptocurrencies have random selection mechanisms among validators that consider not only the currency owned but also the combination of other factors.

However, this problem is solved by the fact that the PoS algorithms have random selection mechanisms among validators that do not consider only the coin held, but also the combination of other factors. For example, Coin Age Selection combines random selection with the concept of "age" of the coins held; if they are not spent in the last thirty days, they take on more value in the selection mechanism: older and larger quantities of coins have a higher probability of signing the next block.

Also in the case of PoS, a 51% attack on the system is possible. However, the probability of such an attack on the system is low because it is necessary to have at least 51% of the coins in circulation to do it, and if it has a high market capitalization, for example, 100 billion, it becomes difficult to appropriate it.

Below, to summarize the main features of the Proof-of-Work (PoW) and Proof-of-Stake (PoS) systems, a table highlighting the differences.

	Proof of Work (PoW)	Proof of Stake (PoS)
Participants	Called miners; open to everyone on the network	Called forgers; creator of a new block is chosen based on the amount of stake
Requirements	Requires burning an external resource (mining hardware, power)	Requires a high stake on the cryptocurrency to be determined as a block validator
Creation of Cryptocurrencies	New cryptocurrency coins are created each time a transaction is validated; serves as a block reward	Has a set amount of circulating cryptocurrencies; coins were pre-mined in advance
Validation Process	All miners compete with one another to solve a cryptographic puzzle to validate a transaction	Set validators participate in a consensus algorithm to vote on the next block to be forged
Incentivization	Block reward is given	No block reward; the forger takes the block's pooled transaction fees

Figure 2.6: main differences between PoW and PoS

2.5 Bitcoin halving

In the Bitcoin system, there is no central bank, such as the ECB or the FED, that regulates the monetary base. At the time of Bitcoin's birth, an algorithm was set up to regulate it, thus already defining in advance how much currency will be created, how and at what rate. The maximum number of bitcoins is 21 million, contrary to legal tender where its supply can be increased discretionally by the central authority, creatable by 2140 and the creation of new coins occurs when a miner adds a new block to the chain and receives the reward.

As explained in Section 2.3.3, the more users try to mine bitcoins, the more complicated the process becomes, and, in addition, the rewards are lowered. In the case of bitcoins, they are halved about every four years, or more precisely every 210,000 blocks (remember 2016 blocks every two weeks).

As for Bitcoin initially, the reward for the extraction of a block was equal to 50 BTC[17], but today it has dropped drastically and is equal to 6.25 BTC.

What explained above has strong implications on the market value of bitcoin, as it introduces a potential deflationary trend due to the limited supply (Bação et al, 2018).

As new bitcoins are produced, one might think of the phenomenon of inflation, because the issuance of new currency causes bitcoins to devalue. However, as new bitcoins are produced, we get closer and closer to the maximum supply, and it is expected that if this cryptocurrency continues to be used and has market demand, inflation will drop over time to zero as it becomes an increasingly scarce asset, and it is reasonable to expect it to increase in value.

At the time of halvings, when the mining rate halves, the inflation rate halves. When it is no longer possible to produce them, inflation will cease and bitcoin will become a deflationary currency: meaning that once it stops being mined there can be no devaluation and, in theory, it will increase in value. So, in conclusion, bitcoin is an inflationary currency until it reaches the 21 million cap, from then on inflation will cease and it will become deflationary.

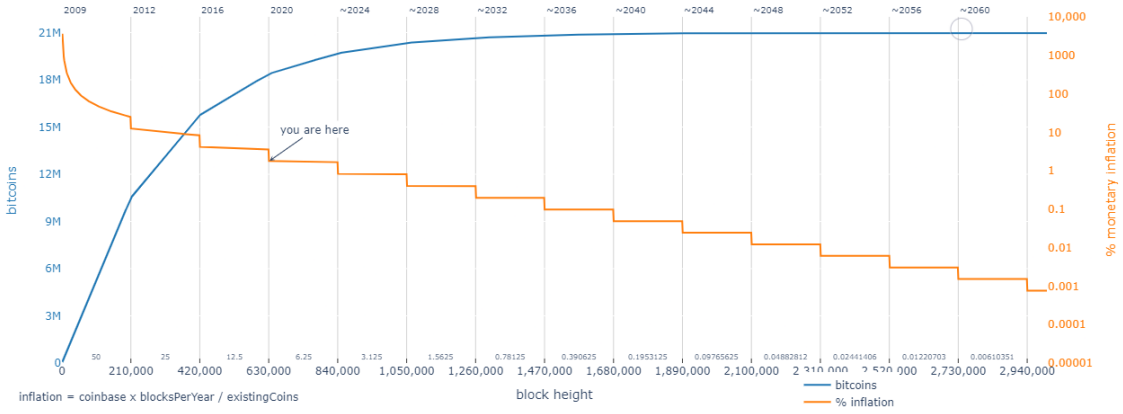


Figure 2.6: Bitcoin monetary inflation

[17] The code with which bitcoin is indicated.

Figure 2.7 shows in blue the total of bitcoins that have already been produced and will be produced in the future: the total amount of new bitcoins depends on the increase in the blocks number and on the reward given to the miners. It has an increasing trend, with a decreasing marginal increase, because miners reward is halved every four years or so; the trend will remain the same until reaching 21 million bitcoins, after which the growth will be zero.

Colour orange highlights inflation: the downward trend of step-by-step inflation is evident, with a downward marginal deceleration that halves when halving occurs. It will maintain this trend until the maximum number of bitcoins are created, at which point inflation will be zero and from that point on it will become a deflationary phenomenon.

3. ARTIFICIAL NEURAL NETWORKS (ANNs)

3.1 Artificial Intelligence, Machine Learning and Deep Learning

Increasingly we hear about Artificial Intelligence, but also about Machine Learning and Deep Learning, with the latter terms sometimes improperly used as synonyms of the former.

The term Artificial Intelligence (AI) was first coined in the '50s and involves all those computational machines capable of performing tasks characteristic of human intelligence. These may include planning, understanding language, recognizing objects and sounds, learning, and problem-solving, for example.

Machine Learning (ML) is essentially a way for implementing Artificial Intelligence. It is a kind of subset of AI that focuses on the ability of machines to receive a set of data and learn on their own, modifying algorithms as they receive more information about what they are processing. The term Machine Learning was coined later than AI, meaning the ability of a machine to learn without being explicitly programmed. Machine Learning is a way to "educate" an algorithm so that it can learn from various environmental situations. Education, or even better training, involves the use of huge amounts of data and an efficient algorithm to adapt (and improve) according to the situations that occur. Machine Learning automates the construction of the analytical model. It uses neural network methods, statistical models, and operations research to find hidden information in the data. A neural network is inspired by how the human brain works. It is a computational system made up of interconnected units (such as neurons) that process information by responding to external inputs, thereby transmitting related information between different units.

Deep Learning (DL) is one of the approaches to Machine Learning that has taken its cue from the structure of the brain, namely the interconnection of various neurons. Deep Learning uses neural networks models with various processing units; it takes advantage of computational advances and training techniques to learn complex patterns through a huge amount of data. Common applications include image and speech recognition.

The concept of deep learning is sometimes referred to simply as *"deep neural network"*[18], referring to the many layers involved.

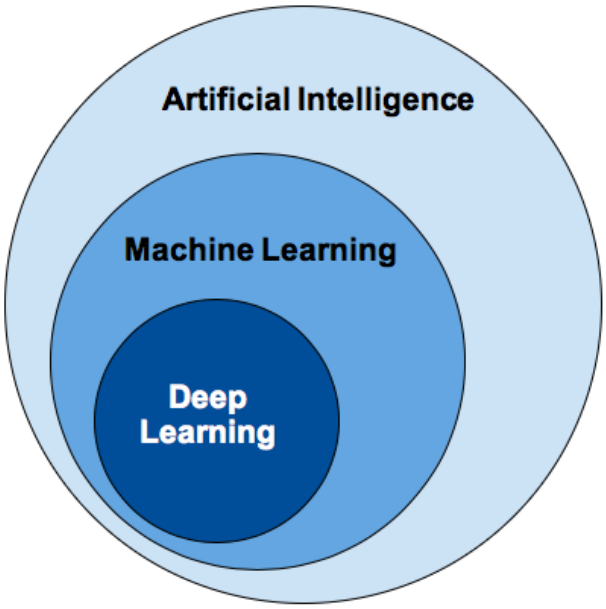


Figure 3.1: hierarchy of AI, ML and DL

3.2 How an Artificial Neural Network works

As already mentioned in the previous section, neural networks are models of machine learning that are born to reproduce typical activities of the human brain. The latter is composed of an intricate network of cells connection that can influence each other and that, working together, try to accomplish a specific task. These nerve cells are the neurons, nerve cells composed by a body, called soma, that is responsible for the processing of input signals and deciding whether transmits them forward or not. Neurons have extensions called dendrites, with many branches, through which the neuron receives electrical signals from other neurons. Each neuron is also formed by a filamentous extension called an axon that branches at the end of the terminals, through

[18] It is called a simple artificial neural network when you have only one hidden layer, while a deep neural network when hidden layers are multiple.

which electrical signals destined to other cells, or dendrites of other cells, are propagated. The connection between a terminal and a dendrite is called a synapse.

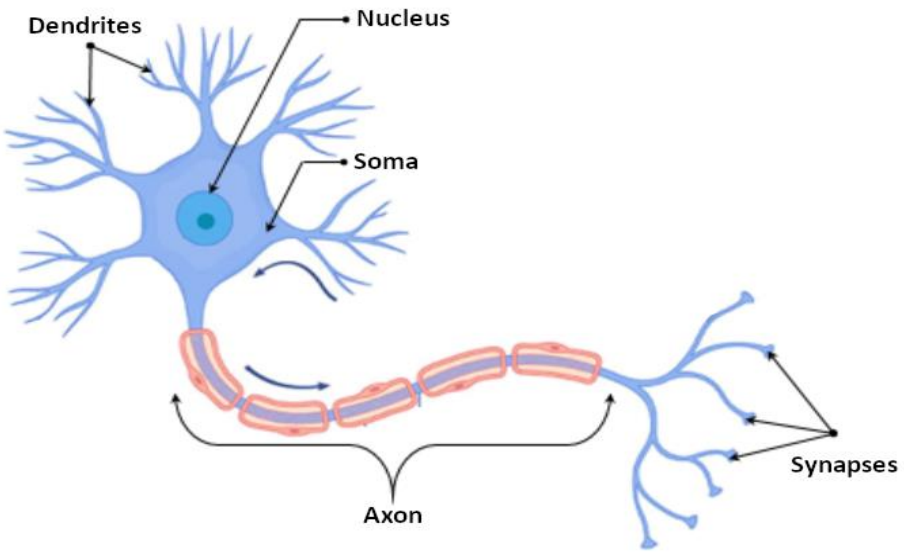


Figure 3.2: a biological neuron representation

The brain is a very complex and powerful computer even though it is made up of very simple processing elements such as neurons. For brain activities, of fundamental importance is the ability of the brain to learn, that is, to modify the connections between neurons based on experience.

An instrument of artificial computation, like artificial neural networks, that wants to reproduce mathematically the brain processing ability, must be realized through a network of elements computationally simple. It must be able to learn and reproduce outputs for unknown inputs, which means acquiring generalization capabilities.

To develop an efficient neural network, like a human brain, you have to make it learn.

The learning process is based on machine learning algorithms that allow you to train the network to improve the accuracy of calculations.

An artificial neuron of a neural network is typically made up of many inputs and a single output. Each input is associated with a weight that determines the conductivity of the

input channel. The output of the neuron is represented by a function of the weighted sum of the inputs.

The training of the neural network is made by feeding it with a series of examples, of which the network knows the nature, which constitutes the training dataset. The responses provided by the network for the training samples are compared with the expected responses by assessing the error between the two. Based on this error, the net weights are changed. This procedure is repeated until the network outputs provide an error below a preset threshold.

Below is a more qualitative representation of how a neural network is structured. Given a neuron with *n* input channels $x_1, x_2, ..., x_n$; to each input is associated a w_i weight, consisting of a real number, which reproduces the neural synapses. It is possible to introduce a threshold called bias (*b*) with the effect of changing the input value of the transfer function, useful for a successful learning. The output signal y_{out} is obtained through the activation function of the weighted sum of inputs plus a bias, which is the said activation level $a = \sum_{i=1}^{n} w_i\, x_i + b$:

$$y_{out} = f_{act}(a) = \sum_{i=1}^{n} w_i\, x_i + b \qquad (3.\ 1)$$

where f_{act} is the neuron activation function, also called the transfer function.

To better understand the above, here is a graphical representation of how an artificial neuron works.

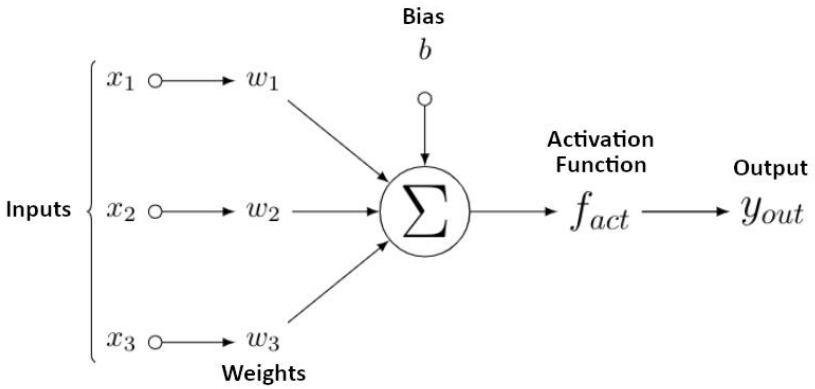


Figure 3.3: an artificial neuron representation

So, after explaining the composition of the biological neuron and the artificial neuron, analogies between them can be highlighted in the following table.

Biological Neuron	Artificial Neuron
Dendrites	Inputs
Soma	Node
Axons	Outputs
Synapses	Interconnections

Figure 3.4: analogies between biological and artificial neuron

A concept often associated with neural networks is *"black box"* because, unlike an algorithmic system where it is possible to examine in detail the process that generates the output from the input, the ANNs can generate a valid result, or at least with a high probability of being acceptable, but it is not possible to define analytically the path that has led to the generation of that result.

3.2.1 Activation and Loss Functions

As mentioned in the previous section, the artificial neurons are given inputs, to which weights are associated. Then a weighted sum of the input is calculated and, subsequently, is given to an activation function that converts it into output. So basically an activation function is used to map the input to the output. This activation function helps a neural network learn complex relationships and patterns in data. These functions are selected according to the structure used and, within the same ANN, different activation functions can be adopted for each layer.

One of the functions most commonly used is the logistic function, or sigmoid, defined by equation *(3.2).*

$$f_{act}(x) = \frac{1}{1 + e^{-x}}$$

(3.2)

The advantage of this function, in addition to being differentiable[19], is to compress the values in a range between 0 and 1 and thus to be very stable even for large changes in the values. The function has been used a lot for a long time, but it still has its problems. Sigmoid function has a very slow convergence, since for large input values the curve is almost flat, with the result that the derivative tends to zero. This lack of responsiveness towards the ends of the curve tends to cause vanishing gradient[20] problems. Moreover, since it is not zero-centred, the values in each learning step can be either all positive or all negative, which slows down the network training process. This is a function that is not much used in intermediate layers anymore but is still very valid in output for classification tasks.

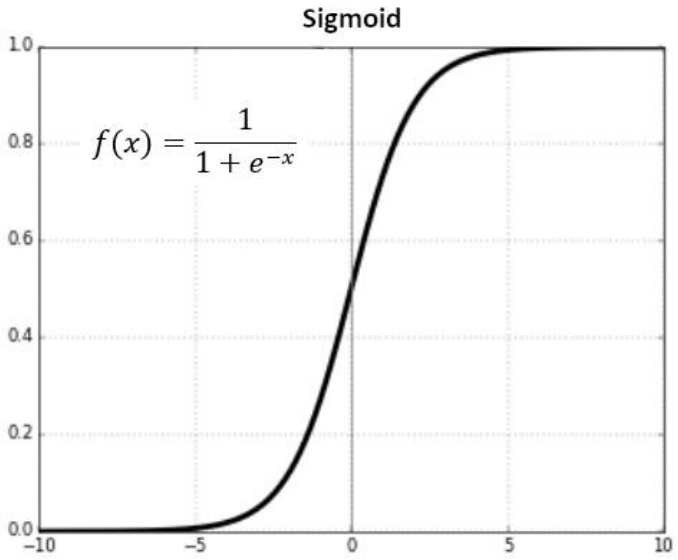


Figure 3.5: Sigmoid activation function chart

[19] Differentiable means that it is possible to find the slope of the sigmoid curve at any point.
[20] The problem is that the derivative of the function shrinks with each step, so networks with many layers tend to fade the gradient, slowing down convergence a lot.

Another important activation function is the ReLU (Rectified Linear Unit), defined by equation *(3.3)*.

$$f_{act}(x) = argmax(0, x) \tag{3.3}$$

It is a function that has become widely used, especially in intermediate layers. The reason is that it is a simple function to calculate: it flattens the response to all negative values to zero while leaving everything unchanged for values equal to or greater than zero.

This simplicity, combined with the fact that the vanishing gradient problem is drastically reduced, makes it a particularly attractive function in intermediate layers, where the number of steps and calculations is large. Calculating the derivative is very simple: for all negative values, it is equal to zero, while for positive values it is equal to 1. At the origin, the derivative is undefined but is set to zero by convention.

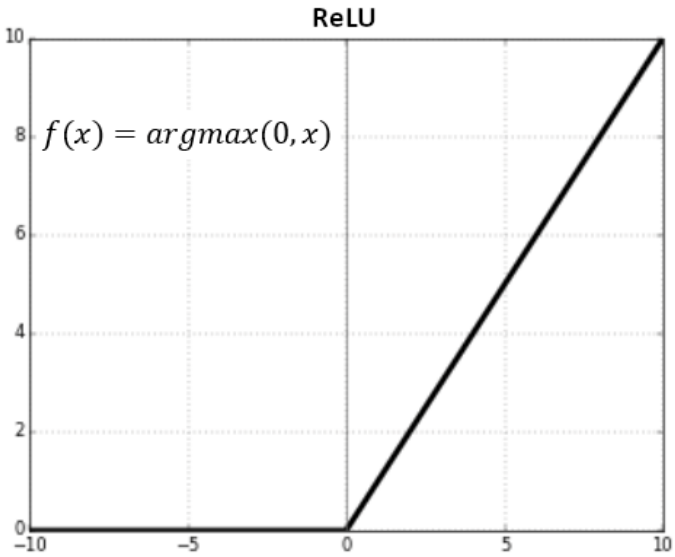


Figure 3.6: ReLU activation function chart

In artificial neural networks, the learning algorithms work iteratively and stop according to a stopping rule defined by a cost function. Each type of cost function is based on the difference between predicted and actual values, producing a loss score as output. As a result, the best possible scenario is to have a loss score equal to zero, which means having built a perfect model that can be used to make predictions.

Among the most commonly used metrics, there are: Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, Mean Absolute Percentage Error.

The Mean Absolute Error (MAE), reported in equation *(3.4)* where y_i is the observed value of the output and $\hat{y}_i$ is the estimated value. It consists of the arithmetic mean of the differences of the forecast errors in absolute value.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3.4}$$

The Mean Squared Error (MSE), in *(3.5)*, is the arithmetic mean of the differences in the squared prediction errors. It is used more often in the "rooted" form, given in equation *(3.6)*, as it is expressed in the same units as the variable of interest. This is called Root Mean Squared Error (RMSE).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.5}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{3.6}$$

The Mean Absolute Percentage Error (MAPE) is the arithmetic mean of the ratios between the absolute value of the forecast errors and the original output as a percentage.

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

(3.7)

All the indicators presented in this section have in common the measurement of deviations between estimated and original output. Each of them has some pros and cons depending on the structure of the data to be analysed.

For example, the MAE over the MSE is preferable in the presence of numerous outliers, values whose amount is accentuated by the square elevation. On the contrary, in the presence of complex mathematical calculations in the application of the learning algorithms, the use of the MSE is preferable to the MAE because the use of the absolute value worsens the efficiency of the processing (Chai and Draxler, 2014). The difference would be zero only if all the elements involved and described above (inputs, architecture, activation functions) can fully represent the phenomenon under analysis.

3.2.2 Delta Rule and Backward Propagation of Errors

The Backward Propagation of Errors, usually abbreviated as Backpropagation, is the most widely used algorithm for training artificial neural networks. It can be considered a generalization of the Delta Rule which, in turn, is a rule used to update synaptic weights based on the Gradient Descent method. The latter is an iterative procedure that leads to the identification of a global minimum point of a function based on locally available information.

The Delta Rule evaluates the model based on an input whose corresponding correct output is known. It adjusts each synaptic weight to an amount proportional and opposite to the contribution which the same weight contributes to the previously identified cost function. Assuming a neural network structure with *i* output, the error recorded by the *i-th* unit can be defined as equation *(3.8)*, where $\hat{y}_i$ is the estimated response, while y_i is the desired response (the value assumed by the series).

$$\delta_i = \hat{y}_i - y_i$$

(3.8)

Having defined a generic cost function *E* that can measure the distance between $\hat{y}_i$ and y_i, the learning process for the neural network consists in minimizing *E* by modifying the weights in the opposite direction to the gradient of the function itself (hence the term Gradient Descent), looking for the minimum.

Take, for example, the MSE (Mean Squared Error) function *(3.9)*, which is certainly one of the most used in the field of artificial neural networks. This function is always non-negative and, as with all cost functions, the condition that $E \approx 0$ if $\hat{y}_i \approx y_i$ applies; therefore, the predictions generated by the neural network model will be more accurate the greater the proximity of *E* to 0.

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

(3.9)

Since the value of $\hat{y}_i$ depends directly on the synaptic weights, to minimise *E*, the synaptic weights *w* of the model need to be modified by a certain amount *Δw*, applying the so-called Delta Rule (3.10).

$$\Delta w = -\eta \frac{\partial E}{\partial w_{ij}}$$

(3.10)

Where:

- η is the learning coefficient, or learning rate, represented by a number between zero and one, which determines the learning rate of the neuron. Regarding the latter, there is a trade-off between slow learning (η close to zero) and fast learning (η close to one), which however leads to the detection of sub-optimal points.

- $\frac{\partial E}{\partial w_{ij}}$ is the partial derivative of the cost function relative to the set of synaptic weights, which represents the contribution of each parameter to the model error.

In this context, if the error *E* and the weights increase in a directly proportional way, the derivative of *E* relative to that of *w* assumes a positive value; therefore, considering the negative sign in front of the derivative, the weights *w* are decreased. Conversely, if the error *E* and the weights *w* move in opposite directions, the synaptic weights *w* are increased.

The minimization of a cost function through the technique described in the previous paragraphs is the basis for the implementation of the Backpropagation algorithm. The latter owes its name to the particular technique of propagating the error backwards, from the output layer to the hidden layers. The gradient of the cost function is defined by equation *(3.11)*.

$$\nabla E(\vec{w}) = \left[\frac{\partial E}{\partial w_{ij}}\right]_{ij}$$

(3.11)

The latter represents the rate of change of the cost function *E*, for a specific neuron *i* in a layer of the network *j*, in relation to the change in the weighted sum of inputs related to *i* in the same layer.

Through successive iterations, called epochs, the Backpropagation algorithm aims to minimize the error by updating the weights according to the gradient as per *(3.12).*

$$\vec{w} = \vec{w}_c - \eta \nabla E(\vec{w})$$

(3.12)

Where $\vec{w}$ represents the vector of the updated weights, *c* identifies the vector of the random weights which initialize the network and η is the learning rate defined above.

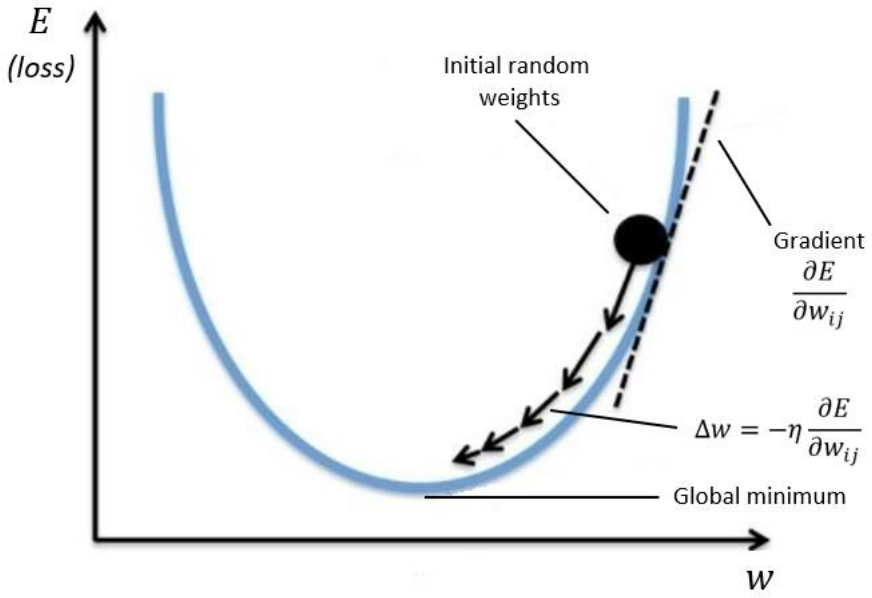


Figure 3.7: Gradient Descent

One of the biggest problems in deep neural networks training is the vanishing gradient (mentioned in section 3.2.1), where the gradient is zero or asymptotic to zero, thus preventing the upgrading of the weights. During the backpropagation phase, the weights near the input layers remain constant or update very slowly, contrary to what happens for the layers close to the output.

One of the main causes of vanishing gradient is the use of activation functions with limited codomain. Therefore, without going into technical details, it can be said that the activation function plays an important role in countering this problem. The use of the ReLU activation function contributed to the resolution (at least partially) of the vanishing gradient.

3.3 Learning paradigms

Learning is the property that distinguishes neural networks from other models. It is based on the ability of the network to learn and improve its functioning through a series of progressive adjustments applied to synaptic weights.

Different types of learning can define the path leading to the modelling of an Artificial Neural Network. The learning methodologies used to train neural networks can be grouped into three distinct learning paradigms which are: supervised learning, unsupervised learning and reinforcement learning.

The type to be used is functional to the scope of the ANN and depends on whether the architecture is feedforward or feedback type.

Feed-forward ANNs allow signals to travel only in one direction: from input to output. There is no feedback, that is, the output of any layer does not affect that layer. Feed-forward ANNs tend to be simple networks that combine input and output. Feedback (or recurring) ANNs can have signals travelling in both directions by introducing loops into the network. They are powerful and can become very complicated. Calculations derived from previous inputs are re-introduced into the network, which gives them some kind of memory. Feedback ANNs are dynamic, their "state" changes continuously until they reach a point of equilibrium.

In supervised learning, inputs are submitted to the network to which known outputs correspond. In this context, the network is trained by also processing the outputs, identifying the relationships that connect them to the inputs and setting the synaptic weights to improve the response of the model.

The application of this model involves a process that breaks down the historical series into two subsets: the training set used to train the network, and the validation set which aims to verify the performance of the algorithm. The algorithm is then applied to the training set and proceeds with a series of adjustments that are made using a set of rules aiming at minimizing the average error. The latter represents the difference between the output produced by the ANN and the actual values recorded in the data set forming the training set. The validation set then aims to measure the degree of generalization of the model built on the training set. If the network is unable to provide adequate responses to a data set external to the one used for training, the training phase must be repeated.

Unsupervised learning, on the other hand, is based on training algorithms based on input neurons alone, which attempt to identify clusters representative of input data.

Since it does not have a target, the network tries to understand some properties of the inputs only according to their layout. Here, too, the learning methodology is dynamic, but it is the neurons themselves that modify the synaptic weights, leaving the same network the task of identifying a logical classification of the input data.

Finally, in reinforcement learning, a particular algorithm performs certain actions on the environment and is guided in learning by the feedback that the environment has on the algorithm itself. Actions are processed using a reinforcement function, which measures the degree of effectiveness of action concerning a pre-established objective. The reinforcement function then assigns a reward (positive real value) or a penalty (negative real value) proportional to the approach or distance of the target.

However, the three categories of learning mentioned above have in common the possibility of formulating the general model as a function of optimizing synaptic weights; particularly in the case of supervised and reinforcement learning, the problem is to minimize an error function given by the difference between estimated values and actual outputs.

3.4 Deep Neural Networks

In the neural networks field, deep learning has been introduced through the definition of deep neural networks. The principle of operation is the same as that of classic neural networks, with the difference that lies in the high number of hidden layers of intermediate neurons. Like classic neural networks, deep neural networks can model complex relationships between input and output data.

The architecture of deep neural networks represents how they are structured in their entirety; it refers to how many units the network can have and how they can be arranged and connected. Remembering what has been said in previous sections, most neural networks are organized into groups of units called layers. Most neural network architectures organize these layers through a chain structure where each layer becomes a function of the preceding layer.

In this type of architecture, the most important variables to analyze are the choice of the depth of the network (number of layers) and the width of each layer, i.e. the number of neurons on it. A network with even one hidden layer is enough to approximate the data. Deeper and deeper networks often can use more units connected to other layers, thus having the opportunity to improve generalization on the test set. The multi-layered architectures that characterize deep learning often tend to be difficult to optimize, so an ideal architecture for a given task is found experimentally by monitoring the change in classification error committed by the trained neural network, calculated on test data. We then described neural networks as a simple chain of layers characterized by the depth of the model and the width of each layer. By varying these parameters, a considerable number of different architectures can be obtained. Many of these have been developed for specific tasks and in this work, we explore three of the most widely used types: Multi-Layer Perceptron (MLP), Recurrent Neural Networks (RNN) and Long-Short Term Memory (LSTM).

3.4.1 Multi-Layer Perceptron (MLP)

When talking about Artificial Neural Networks, they are often referred to simply as Neural Networks or even Multi-Layer Perceptron (MLP), with the latter being a widely used type of neural network. A perceptron is the equivalent of a single neuron, which is the basis for the construction of larger neural networks.

A Multi-Layer Perceptron is a class of neural networks which falls under the category of feed-forward algorithms. It is composed of at least three layers. Each layer, except the input layer, is a set of neurons using a non-linear activation function. The nodes (or neurons) of the Multi-Layer Perceptron are arranged in layers: an input layer, an output layer and hidden layers (layers between input and output).

The learning algorithm for MLPs is the following: as with the perceptron, inputs are pushed forward through the MLP by taking the sum of the product between the input data and the weights between the input and the hidden layer (the so-called weighted

sum). This sum returns a value to the hidden layer, but it is not pushed forward as it would have been done with a perceptron. MLPs use activation functions at each of the calculated layers. There are many activation functions, some of which are analyzed in section 3.2.1. The computed output is pushed onto the current layer through any of these activation functions. Once the computed output on the hidden layer has been pushed through the activation function, it is passed to the next layer of the MLP by taking its weighted sum. These last two steps are repeated until the output layer is reached.

At the output layer, the calculations will be used for a backpropagation algorithm corresponding to the selected activation function for the MLP (in the case of training), or a decision will be made based on the output (in the case of testing). At each iteration, after the weighted sums have been passed through all layers, the MSE gradient is calculated on all input and output pairs. Then, to propagate it backwards, the weights of the first hidden layer are updated with the gradient value.

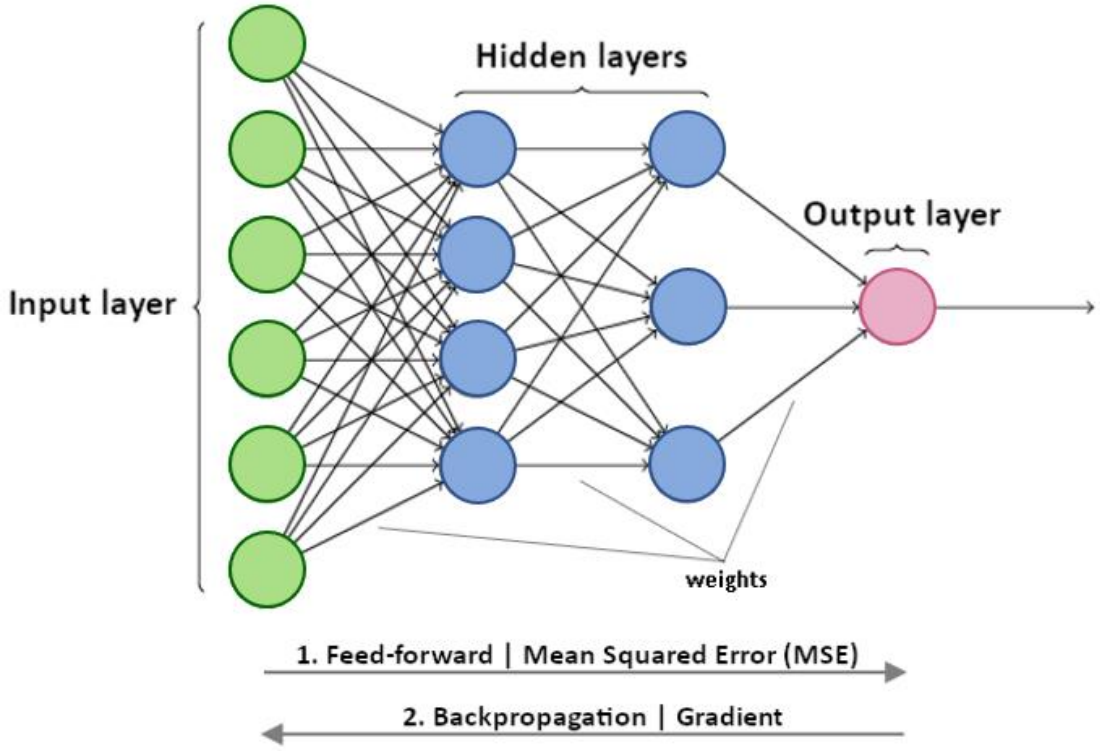


Figure 3.8: example of MLP

3.4.2 Recurrent Neural Network (RNN)

When dealing with sequential data or historical series, traditional feed-forward neural networks, such as MLP, cannot be used for learning and forecasting. A mechanism to store past or historical information is needed to predict future values. Recurrent Neural Networks (RNNs) are a variant of conventional feed-forward artificial neural networks that can handle sequential data and be trained to preserve knowledge about the past. In feed-forward neural networks, as explained in the previous section, the information/signal can only go in one direction and each neuron can be interconnected with one or more neurons of the next layer. Unlike the latter, neurons in RNNs can also accept loops and/or be interconnected to neurons of a previous layer.

RNNs, therefore, provide backward or upward connections. This characteristic makes this kind of neural network very interesting because the concept of recurrence indirectly introduces the concept of network memory. In RNNs the output of a neuron can influence itself (i.e. at time *t*), a subsequent time step (i.e. at time *t+1*), or neurons of the previous layer (i.e. at time *t-1*), which in turn will influence the behaviour of the neuron on which the loop closes.

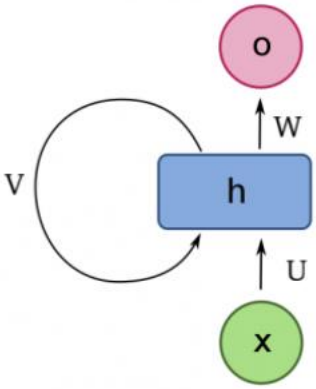


Figure 3.9: a single neuron in RNN

As explained above, the output *O(t)* is a function of the input *X(t)* and in part of the value *V(t)* which, due to the cycle, corresponds to the previous output *O(t-1)*. But the output

O(t-1) in turn depended on the value of the input *X(t-1)* and of *V(t-1)* which, again due to the effect of the cycle, did correspond to the previous output *O(t-2)* and so on. What has just been described is schematized in the following figure, where the feedback loop is unfolded along the time sequence, thus highlighting the correlations between future and past values.

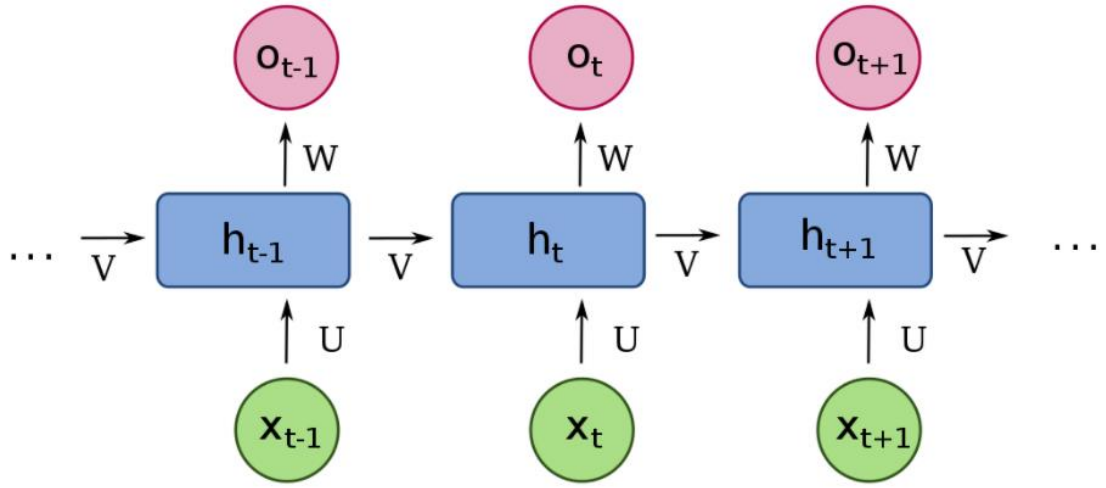


Figure 3.10: example of Recurrent Neural Network

Thus, a simple loop allows the neurons of a recurring network to have a memory of past inputs and thus to correctly handle situations similar to those described above, where it is necessary to maintain a context to process the information correctly.

A typical problem with RNNs is the vanishing or exploding gradient (see 3.2.1). As mentioned in section 3.2.2, during the training of neural networks, the backpropagation method is used. It updates the various parameters of neurons (weights and bias) in proportion to the partial derivative of the loss function compared to the parameter itself.

The problem here is with the activation function. If, for example, a sigmoidal function is used, the values of their gradients remain in the range [0, 1]. Since the backpropagation algorithm requires gradients to be multiplied in a chain along with the layers, it is clear that the product of a large number of values between 0 and 1 causes the total value to

decrease rapidly along the neuron chain (vanishing gradient). If, on the other hand, linear functions such as ReLU are used, the gradient values can be even greater than 1 and thus the total value can increase enormously along the neuron chain (exploding gradient).

3.4.3 Long-Short Term Memory (LSTM)

There are several ways to limit the effect of vanishing/exploding gradient. For example, more complex recurrent neurons than described above can be used. Long-Short Term Memory (LSTM) neurons are a frequent solution to overcome the above problem. As you can see from the picture below, the LSTM neuron is much more complex than the simple recurrent neuron. The latter simply handles an additional output *h(t)* for the current state and an input *h(t-1)* for the previous state.

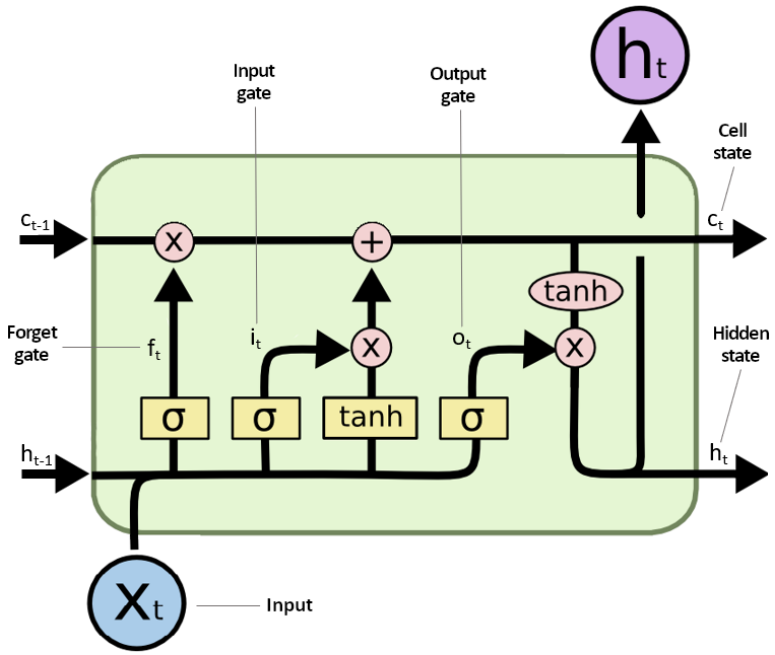


Figure 3.11: a single neuron in LSTM

The LSTM neuron, on the other hand, besides two different activation functions for the *h* state and the *z* output, inside it has several ports (or gates). These gates allow it to

decide autonomously (during the training phase) what is worth remembering or forgetting, if and how to combine the input with the internal state, if and how to return the output. It is essential to define the role of these gates:

- the forget gate decides whether the input information should be discarded or retained. It receives the information of the current input *x(t)* and the previous output in feedback *h(t-1)*. A sigmoid activation function is applied to this information, which will return an output between 0 and 1. This output then will be multiplied by the value of the previous state *c(t-1)*. So if the sigmoid returns a value close to 0, the previous state will tend to be zero (get forgotten), while if it returns a value close to 1, the previous state will tend to remain the same (get stored).
- the input gate similarly decides whether the input values *x(t)* and *h(t-1)* can be processed together with the previous state *c(t-1)* (or what remains after passing through the forget gate).
- the processing of the inputs and the current state (or at least the values that the forget and the input gate let pass) becomes the current state of the neuron, *c(t)*.
- finally, the output gate, similar to the other gates, exploiting the input values *x(t)* and *h(t-1)* decides whether the current state *c(t)* can be presented as output and become the output *z(t)* which also corresponds to the next input *h(t)*.

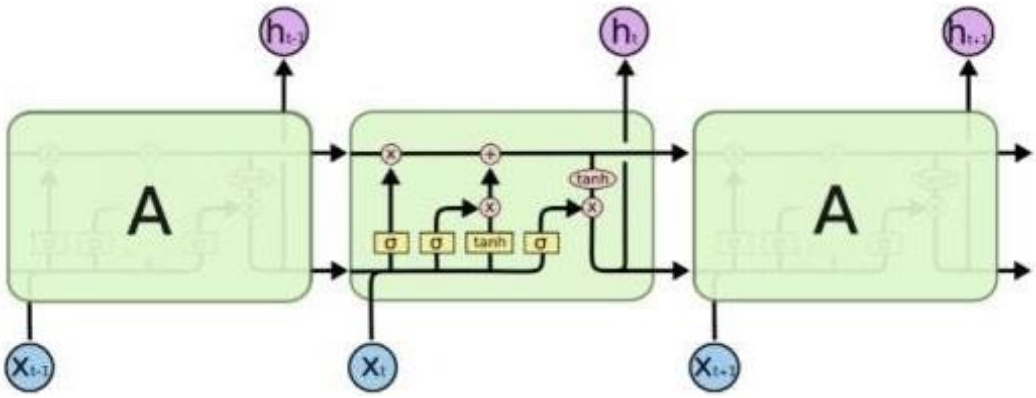


Figure 3.12: example of LSTM

4. NEURAL NETWORK FOR BITCOIN PRICE FORECASTING

After a general analysis of cryptocurrencies and artificial intelligence, in particular neural networks, it's time to move on to empirical analysis. The final part of this paper focuses on artificial intelligence techniques applied to the cryptocurrency market. To do this, as anticipated in the previous sections, Long-Short Term Memory (LSTM) neural networks applied to bitcoin (BTC) are used.

The goal is to use information derived from the history of the digital asset and, with the help of deep learning methodologies, to be able to make predictions about their performance. For this purpose, a neural network model based on a supervised learning algorithm (see section 3.3) will be used, the architecture of which will be described below.

For the implementation of the neural network, "R", a programming language, is used. It is a specific development environment for statistical data analysis. This is also supported by Python's Keras library, specifically used for machine learning and neural networks.

4.1 The Dataset

The variable under study, as mentioned above, is the closing price[21] of bitcoin, called in the model by the abbreviation BTC. Looking at the chart in *Figure 4.1* it is possible to see the price trend in USD (United States Dollar) in the period considered which goes from September 17, 2014, to December 31, 2021. It provides a visual reflection of what has been said above, namely the incredible development that bitcoin has had as well as its volatile performance.

[21] The closing price is the last level at which an instrument is traded on a given day or viewed by many as an indicator of market sentiment.

Figure 4.1: BTC price in USD over time

Until 2017, the price has always fluctuated under $1,000, while in the course of 2017 it has recorded an exponential increase reaching almost $20,000 by the end of the year, with an increase of about 1.900%. Then again undergo a fairly noticeable market correction, followed by a bear market of about three years.

Nothing to do, however, with the growth that occurred between the end of 2020 and the beginning of 2021. This period also corresponds to the beginning of the COVID-19 pandemic during which, precisely, there has been a substantial increase in the demand – and therefore in the value – of cryptocurrencies and the adhesion to online trading services. During this period, the price of bitcoin recovered from its previous All-Time high (ATH)[22] at the end of 2017, and then tripled it, reaching $60,000. He then underwent one of his usual and not slight market corrections, and then set another record by hitting a new ATH at $68,680 on November 10, 2021, ending up with a strong correction once again.

It is important to remark that the price of bitcoin is very volatile and for this reason, its development can only be partially studied based on measurable phenomena. This price volatility is often due to two undesirable and non-measurable characteristics associated with bitcoin, such as regulatory disorientation and cybercrime (Corbet et al, 2018). The authors point out that banning bitcoin in some jurisdictions, but also only announcements of possible hostile legislation, cause substantial price declines. Regulatory loosening, on the contrary, usually generates appreciation. They also argue

[22] Acronym for All-Time High, it indicates the highest value reached by a given asset.

that a focal point is hacker attacks: the growth of cybercrime undermines trust and stability in this market with negative consequences on value. There are therefore several irrational and non-measurable factors that influence the performance of the digital currency. They cannot be included in the template but must be considered.

Speaking more precisely of empirical analysis, the dataset consists of a total of 2663 daily observations, arranged in 5 columns compared to the classic composition "*Open, High, Low, Close*" (also referred to as OHLC), followed by "*Volume*". They shall indicate the opening price, the maximum price, the minimum price, the closing price and the daily volume respectively. The goal of the model is to make predictions about the closing price, therefore "*Close*" will be the dependent variable; all other variables, instead, will be the independent variables.

The data is downloaded from the Yahoo Finance website via the "get.hist.quote" function of the "tseries" library of R, which creates a kind of direct connection between R and the Yahoo Finance website, retrieving the updated data in real-time.

Of course, these are raw data and several manipulations are required. Data pre-processing and adaptation are necessary to feed the neural network and to make their use regular and practical.

	Open	High	Low	Close	Volume
2021-12-12	49354.86	50724.87	48725.85	50098.34	21939223599
2021-12-13	50114.74	50205.00	45894.85	46737.48	32166727776
2021-12-14	46709.82	48431.40	46424.50	46612.63	34638619079
2021-12-15	48379.75	49473.96	46671.96	48896.72	36541828520
2021-12-16	48900.46	49425.57	47529.88	47665.43	27268150947
2021-12-17	47653.73	48004.89	45618.21	46202.14	32902725329
2021-12-18	46219.25	47313.83	45598.44	46848.78	26098292690
2021-12-19	46853.87	48089.66	46502.95	46707.02	25154053861
2021-12-20	46707.06	47401.72	45579.81	46880.28	30961902129
2021-12-21	46985.24	49260.99	46739.78	48677.04	29325985792

Figure 4.2: dataset's structure

4.2 Data Pre-processing

Once collected, the data enter the preparation stage. Data pre-processing is the stage where raw data is cleaned and organised for the next step of network training. During preparation, the raw data is strictly checked for errors. The purpose of this step is to delete bad data (redundant, incomplete or incorrect) and start selecting high-quality data to get the best possible execution.

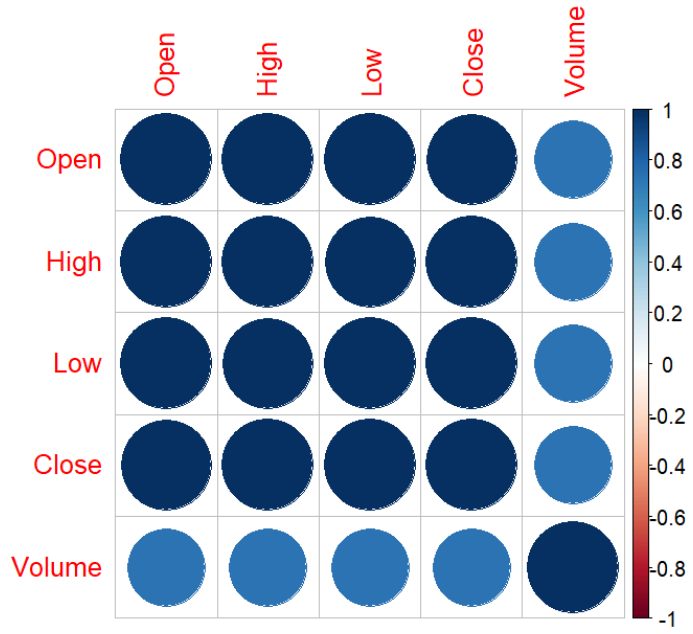
4.2.1 Data Cleaning

The first step in data pre-processing is to clean up the data itself. The first thing we can notice, related to the dataset, is the presence of missing values ("*NA*"). In general, there is no correct way or not to handle missing data. Different solutions are depending on the problem you are facing. In this case, since the missing data are a limited number (4 out of 2663), I decided to delete them. This approach is the so-called "*listwise deletion*" and, in summary, consists of deleting the rows (observations) that contain missing values in at least one of the variables (features).

Another useful tool in preliminary analysis is the *correlation matrix*. It consists of a square table showing the correlation indices between two or more variables. Correlation is a statistical measure that expresses the relationship between two variables and indicates the tendency for two variables (X and Y) to vary together, or "covary".

But what's the problem with the presence of correlation? When variables are correlated (in our case the reference goes to "*Open*", "*High*", "*Low*", "*Close*" *and* "*Volume*"), it indicates that changes in one variable are associated with changes in another variable. The stronger the correlation, the harder it is to change one variable without changing another. It becomes difficult for the model to estimate independently the relationship between each variable because they tend to change together. In my case, you can see from the image below that the variables are all highly correlated. I decided to drop all

variables (by keeping at least one of them) with a correlation index greater than 0.7, so the new dataset will consist only of the variable "*Close*", which is the most important in explaining the phenomenon under analysis.



	Open	High	Low	Close	Volume
Open	1.0000000	0.9990178	0.9985572	0.9980328	0.7485391
High	0.9990178	1.0000000	0.9981537	0.9991463	0.7489595
Low	0.9985572	0.9981537	1.0000000	0.9989919	0.7503264
Close	0.9980328	0.9991463	0.9989919	1.0000000	0.7492187
Volume	0.7485391	0.7489595	0.7503264	0.7492187	1.0000000

Figure 4.3: dataset's correlation matrix

As planned, the next step related to data preparation, before feature scaling, would have been to make the time series stationary through differentiation. This would have allowed work on daily returns and not directly on prices. Unfortunately, this turned out to be not feasible, as in the network training phase the forecasts obtained were not satisfactory. However, taking as reference other works in the literature, working directly on prices can still be considered a valid approach.

4.2.2 Feature Scaling

A common enemy to most machine learning problems is the comparison of data with an unbalanced scale of values. In the dataset introduced earlier, for example, the closing price ranges from $400 to $60,000.

Although the neural network should also be able to adapt to heterogeneous data, the learning phase may be more difficult, triggering larger gradient upgrades and preventing the network from converging. It is, therefore, important to proceed with feature scaling or resizing the data. Batch normalization[23] is performed as a solution to speed up the training phase of deep neural networks, by introducing internal normalization of input values within the neural network level. Generally in machine learning, it is common to normalize input data before passing the data to the input level. One of the reasons the data are normalized is to ensure that our model can generalize appropriately. This is achieved by ensuring that the scale of values is balanced and that the range of values is also maintained and proportional despite the change of scale of values.

There are usually two ways to resize data:

- *z-score standardization:* resizes attributes so that the mean value is 0 and the standard deviation is 1. The formula for standardizing a generic variable is as follows:

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

Where μ is the mean of the samples, σ is the standard deviation of the training data and x is the value to be standardized. This method also considers the anomalous values, standardising the variance of the anomalous values (which would otherwise dominate over the other data). Due to its versatility and minimal information loss, it is widely used for machine learning algorithms.

[23] The term batch refers to the fact that neural networks are usually trained with one set of data collected at a time. This set or group of data is referred to as a batch. The operation within the batch normalization technique occurs on a whole batch of input values instead of a single input value.

- *normalization*: it is the simplest method, data are resized and scaled over a fixed range, usually [0, 1]. This normalization improves the accuracy of the analysis through better distribution of the data. This is the formula for a min-max normalization:

$$z = \frac{x - min(x)}{max(x) - min(x)} \tag{4.2}$$

  Where *z* is the normalized value, *x* is the original value and *min/max* refer to the minimum and maximum values of the dataset. It should be used when there are no anomalous values that are extremely high or extremely low because then the accuracy of the normalized values is affected.

In this work, data was resized using standardization, the most commonly used approach to resize data in machine learning.

4.2.3 Sliding Window and Data Splitting

Three models will be proposed, sharing the same input data and different in the time windows considered or the number of inputs used, based on the autocorrelation between them. These will be in incremental steps, equal to the number of days or which days you want to look back in time, and then use this data in the forecast of future closing price. Specifically, these time windows will include, in order, the previous 7 and 10 days; in the latter case, some days will be removed as the autocorrelation between the time series and its delayed values is evident.

Considering the first case, for example, with a time window of 7 days, the dataset is constructed in such a way that the input variables of the neural network are seven, while the output is only one variable, i.e. the closing price of the next day (the first element of any observation). Specifically, prices at time *t, t-1, ..., t-6* are considered to predict the closing price at time *t+1*. To do this, the original matrix containing the historical series

of 2663 values starts to create a sliding window so that the first element of the first observation is P_{t+1}, the second is P_t, the third is P_{t-1}, and the last is P_{t-6}, where P is the closing price. Then the window starts rolling and all observations will be shifted by one position. The second row of the dataset is as follows: the first element is P_{t+2}, the second is P_{t+1}, the third is P_t and the last is P_{t-5}. And so on until the dataset is complete.

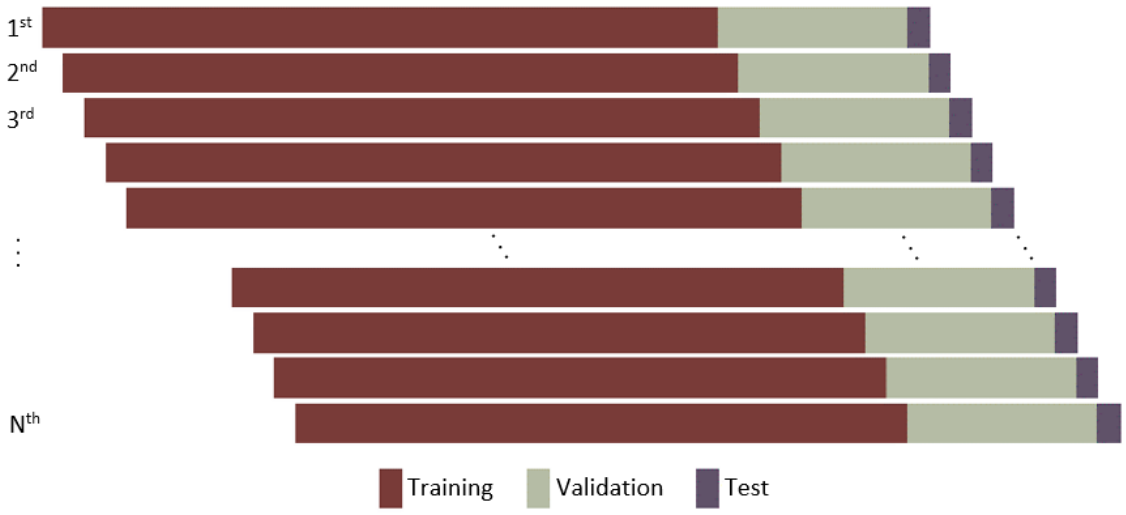


Figure 4.4: sliding window technique

Before the network training phase is carried out, the data shall be subjected to a final process. It consists of dividing the data into three parts, called *training*, *validation* and *test* sets.

The data of the *training set* will be used exclusively to train the model. It means that our model will learn the relationships between our X (the input variables) and Y (the output). About the latter part, there could be a phenomenon called overfitting (discussed in more detail in the next sections). Overfitting means being able to predict perfectly the data used during training, but not being able to generalize on new data. Our objective is not to guess, but to be able to make predictions on data never seen before. This is where the *validation set* comes into play.

Thanks to the training set, the model learned the relationships between input and output. To avoid overfitting, i.e. to have a real predictive capacity, we give our model data it has never seen and we tell it to make a prediction. These data must be labelled,

i.e. they must contain information exactly like that of the training set. At this point, we'll have a predicted Y and a real Y. It will then be enough to compare the predicted Y with the real Y to see how much our model can predict with a good approximation of our output variable. It is called a *validation set* because it is concerned with validating the results obtained in the *training set*. If the performance is poor, we should go modify the model's hyperparameters and start training again, until the *validation set* result is satisfactory.

Once our model can perform well on the *training set* and especially on the *validation set*, we can test our model on other observations that the model has never seen, contained in the *test set*. This is a set that is sometimes even omitted, which serves almost exclusively to evaluate and visualize the performance and functioning of the model. The most robust data is usually from the *training set* because the algorithm needs to learn. Immediately after that comes the *validation set*: a *validation set* that is too large may take data away from learning, but a too-small one may insert a bias that is too strong during the evaluation and thus invalidate the predictive capabilities of the model. Finally, a small part is dedicated to the *test set*.

The dataset under consideration is then divided into 3 sets, maintaining the chronological order of the data. The test set corresponds to all available data for the current year, from January 1st to December 31st, the year for which forecasts will then be made. The remaining data are splitted between the training and validation sets, with proportions of 75% and 25%, respectively.

4.3 Hyperparameter Optimization

The configuration of the neural network consists of the selection of certain variables, also called hyperparameters, a set of parameters that determine the way the neural network is trained and the structure of the neural network itself. This is certainly the most important part of the process. The choice of the number of layers (network depth) and the number of neurons has an important influence on the adaptability of the model and on the processing times. As mentioned in the previous chapter, a complex model

tends to have good performance in the training data, but it is likely to generalize in an inefficient way when it finds itself to elaborate observations not present in the training phase; conversely, a model that is too simple tends not to capture the non-linear components within the data and to lose its effectiveness in terms of forecasting. The configuration of the architecture, therefore, takes place in progressive steps, starting from simple networks with a single hidden layer and a limited number of neurons, increasing progressively the complexity.

We then proceed with the construction of a neural network with three total layers: an input layer, a hidden layer and an output layer. The number of neurons to assign to the output layer is a mandatory choice, as you have to assign one neuron to each of the outputs, in this case only one (the closing price). For the size of the other layers, however, the number of hidden neurons must be modulated according to the number of inputs that are provided (in this case corresponding to which days you want to look back) and by keeping in mind that the total number of parameters in the neural network must be about equal to 1 for every 10 observations, a number chosen by me as a reference. Relative to the input layer, as mentioned earlier, the number of neurons is variable and equal to the days you want to look back. The difficulty lies in determining the number of neurons in the hidden layers. In this case, they are determined by trial and error, considering the rule stated above regarding the number of parameters and according to the best result in terms of MSE, the metric used for model evaluation.

When choosing the model, two particular phases need to be monitored: optimization, which refers to the model's ability to adapt to training data, and generalization, which refers to the model's response to data not included in the training phase.

To this end, three key elements need to be carefully assessed and selected:

- the *activation function*, since it determines the transformation of the data in the passages from one layer to the next;
- the *cost function*, or *loss function*, since it determines the measure of network success;
- the *optimizer*, which determines how neural network weights should be modified during training according to the cost function.

Concerning the activation function, the ReLU is used, considered the standard activation function in deep learning for two main reasons: its simplicity, which translates into shorter computational times, and the best response to the vanishing gradient problem, as explained in Chapter 3.2.1. As for the loss function, the Mean Squared Error (MSE) function is used, which is also discussed in the above chapter. The optimizer chosen is ADAM, an acronym for ADAptive Moment estimation, which is an update of RMSprop. It is a commonly used method, it seems to be efficient in managing problems involving a lot of data or parameters, and it also requires less memory.

The ones listed above are only a part of the hyperparameters to be set for neural network training, also others play an important role. One of these is the **number of epochs**. The number of epochs is the number of times the learning algorithm will go through the entire training data set. You might think that a larger number of epochs will result in a more accurate model, which is not always true. As the number of epochs increases, both the training error and the validation error decrease, but only to a certain extent. After that point, the error continues to increase due to overfitting. Therefore, for the model to work well with new data, the training should end at the point where both the training error and the validation error are at the minimum and the number of epochs at that point is the ideal number. Thanks to the TensorFlow library, you can write code so that when the validation error starts to increase, the training will be stopped.

Another important element is the **batch size**. It can be defined as the number of data (samples) that will be trained by the neural network at once. To understand better: Assuming you have 1000 samples that need to be trained, you can propagate all 1000 samples simultaneously through the neural network or in mini-batch, say 100 at once. The neural network will initially take the first 100 samples and train the network and then the next 100 samples and continue until it is propagated to all the samples. The advantages of using mini-batch are that they require less memory and training is faster than the full batch. In this case, a batch size equal to 128 is used.

Also the **learning rate** is an important concept. It is a parameter with a small positive value (often between 0.0 and 1.0) that controls the speed with which the model adapts to the problem. In other words, the learning rates decide how far the weights must be

in the direction of the gradient to meet the minimum overall. If the learning rate is too low, the training will progress very slowly. If the learning rate is set too high, as shown in the figure, the training may not converge to the global minimum, but instead exceeds it and continues to worsen the loss of the model. The value chosen for my neural network, after several attempts, was set at 0.0001.

Directly linked to the learning rate, there is the **dropout** technique. It is a regularization technique used to reduce validation error, in other words, to avoid overfitting. Simply put, dropout means randomly ignoring neurons, and by doing so they will be ignored in the forward and backward propagation of the neural network. During training, neurons start to depend on each other, which limits the exploitation of the power of an independent neuron and the random elimination of several neurons can prevent this. Dropouts are given as percentages. In my case, a dropout value of 20% is set, which means that 1 in 5 neurons will be randomly removed and ignored during training. Dropouts increase the iterations that need to be performed to converge to the global minimum, but the training time for each iteration will be reduced.

Another useful technique, concerning the problem of overfitting, is that of **regularization**. This technique, acting on the cost function, helps to reduce the effects of the overtraining of a neural network. It provides for the addition of a factor, dependent on the weights, after the expression of the cost function and can be seen as a trade-off between finding small weights and minimizing the cost function. One of the techniques most used and applied also in this work is called *L2 regularization* or decay of weights: it uses the sum of the squares of the weights as a regularization function. The value of the factor chosen for L2 regularization is equal to 0.001.

To sum up, hyperparameters play a crucial role in deciding whether the neural network you've trained applies to the problem you're trying to solve or not. Some of the elements listed above, especially the number of epochs, dropout and regularization, are very important to avoid the overfitting problem. During the training phase of the neural network, thanks to the setting of the aforementioned parameters, overfitting does not seem to have occurred in a problematic way.

4.4 Training and Testing Results

Once the starting model of the neural network and the initial setting of the hyperparameters have been defined, we proceed with the implementation of the model. The subdivision of the dataset takes place as previously mentioned in Chapter 4.2.3. Data from 17 September 2014 to 31 December 2020 are used for network training. They consist of a total of 2298 observations, of which the first 75% of the data represents the training set and the last 25% the validation set. The data from 1 January 2021 to 31 December 2021 (365 observations) represents the test set, for a total of 2663 observations, on which the model is tested.

The neural network is then trained, it has three layers (one input layer, one hidden layer and one output layer) having different structures depending on the model under consideration. The important thing I want to stress on is that network training is interrupted when the *val_loss* does not decrease (i.e. improves) from the minimum point reached, after 5 epochs. This is set to prevent overfitting problems from occurring, thanks to Keras library's *callback* function, which stops training when certain conditions are satisfied, the so-called *early stopping*. During training, the perfect scenario is achieved when the validation loss and the training loss follow a similar pattern, converging; whereas for ideal training, the validation loss should be as small as possible. Each model will be discussed separately below, as it has different features from the others.

4.4.1 Training Phase

The **first model** is based on the fact that to predict the closing price of the next day (P_{t+1}), the neural network is trained by inputting the closing price of today (P_t) and the previous six days (P_{t-1}, P_{t-2}, P_{t-3}, P_{t-4}, P_{t-5}, P_{t-6}). The neural network has a structure 7-5-1 where, respectively, 7 are the neurons of the input layer, 5 are the neurons of the hidden layer, 1 is the neuron of the output layer. Training is initially set for a total of 500 epochs, but due to the early stopping technique, it is stopped after 465 epochs as the validation loss

shows no signs of improvement since its last minimum peak reached for 5 consecutive epochs. The total number of parameters in the neural network amounts to 146, in line with what was established at the beginning of the work, i.e., a maximum ratio of 1 to 10 between parameters and total observations.

Below you can see graphically the trend of *loss* and *validation_loss* during training. As mentioned in the previous sections, a fundamental factor is the convergence of the two lines. The smaller is the distance between the two, the better is the training of the neural network. In this case, the network is trained in a good way and there are no overfitting problems, where otherwise we would have noticed a divergence between the lines and not a convergence between them.

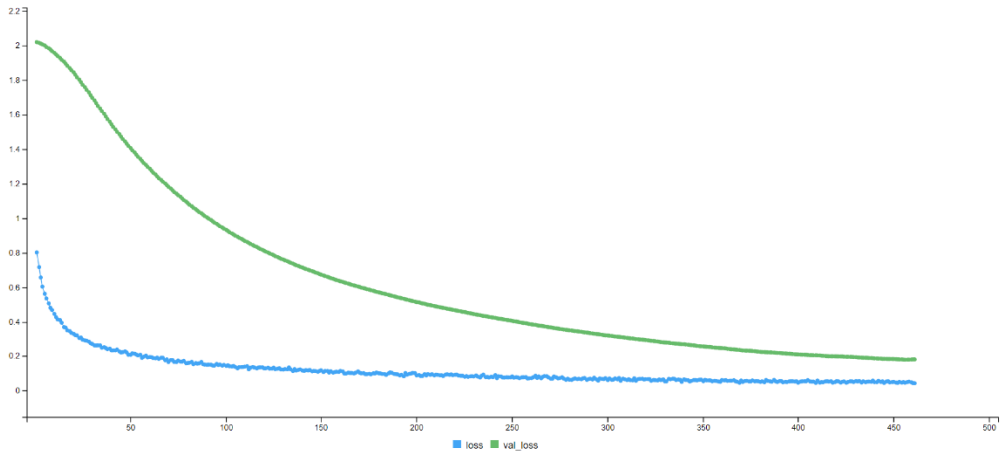


Figure 4.5: 1st model loss vs val_loss

The result of the neural network training on the training set can be seen graphically below. It is to be evaluated by comparing the original behaviour of the historical series with that predicted by the network itself. The length of the original historical series and the forecast series is the same, equal to 1678 values, just over four and a half years of forecasts. *Figure 4.5* below shows the original historical series coloured in blue and the forecast series in red. This way it is easily highlighted where some internal elements are not captured by the model. The results do not include data before 2017, as they have

very low values compared to the current values and would not capture the training trend well graphically.

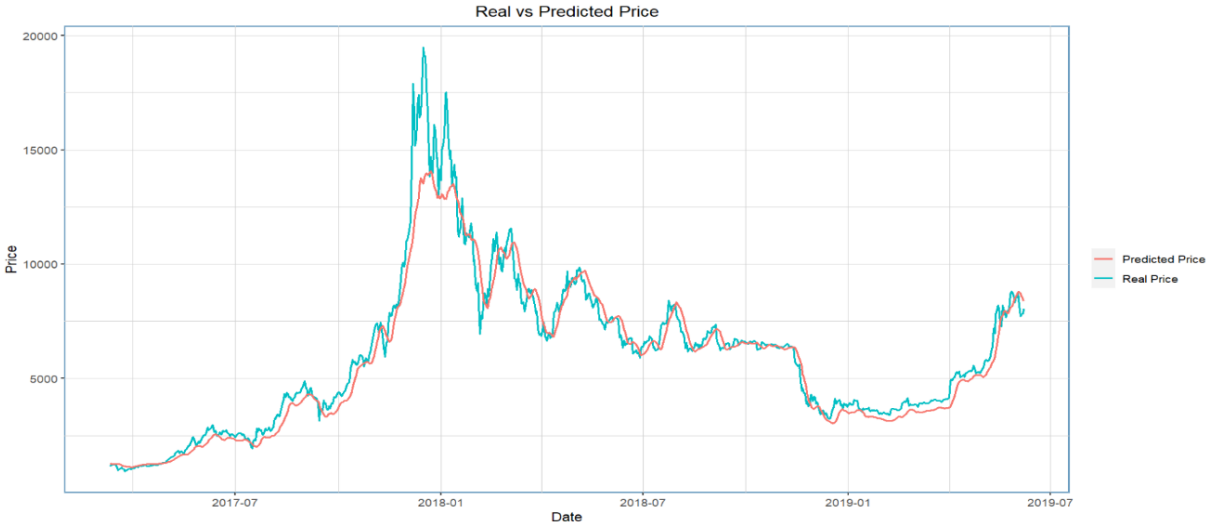


Figure 4.6: 1st model training results

The fitted values (red line) seem to fit relatively well and follow the observed data (blue line) for almost all the entire period considered. Some evident deviations from real values can be seen especially in the most volatile periods when the closing price has experienced sudden increases or decreases, especially between the end of 2017 and the beginning of 2018. Moreover, it is possible to notice that in the first and last part of the graph, the gap between fitted and observed data is greater. As far as the training of the second and third model is concerned, the descriptive procedure will be the same as the first one, so I will not specify any more some of the characteristics listed above. The only difference lies in how data are prepared, the structure of the network and the duration of the training.

The **second model** is based on the fact that to predict the closing price of the next day (P_{t+1}), the neural network is trained by taking into consideration the closing price of the previous ten days and selecting among these only some of them. Specifically, with the help of the Autocorrelation Function and Partial Autocorrelation Function plots (ACF and PACF), the closing price P_t, P_{t-5}, P_{t-6}, P_{t-7}, P_{t-8} and P_{t-9} are selected. The neural network has a structure 6-4-1 where, respectively, 6 are the neurons of the input layer, 4 are the neurons for the hidden layer, 1 is the neuron of the output layer. Training is interrupted

after 500 epochs and again, good network training can be seen. The total number of parameters amounts to 101.

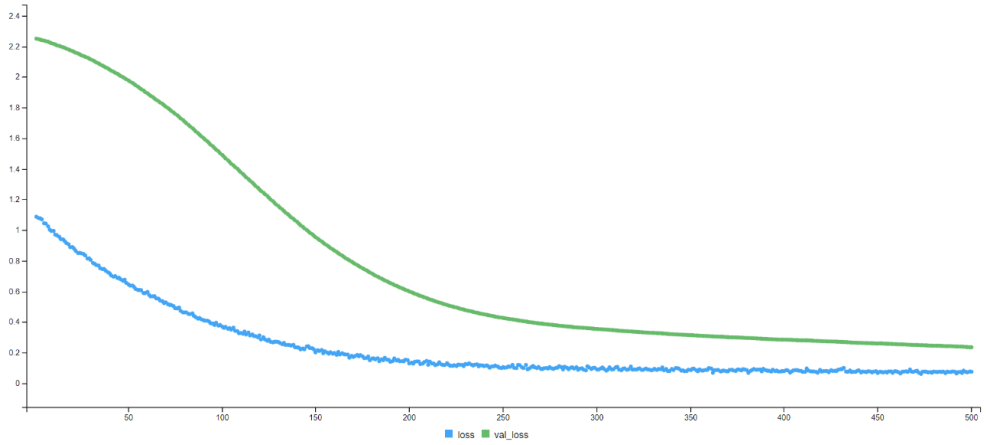


Figure 4.7: 2nd model loss vs val_loss

Figure 4.8: 2nd model training results

The fitted values (red line) seem to fit quite well the observed data (blue line) for the entire period considered. Here, deviations are more pronounced than in the previous model, especially between late 2017 and early 2018.

The **third model** is slightly different from the previous one, but its content changes significantly. The goal is to see if it is possible to make price predictions considering a time frame larger than just one day. It takes into consideration the same days of the

second model, except for price P_t (today's price) as it is not significant in the analysis of ACF and PACF plots. Consequently, the price forecast will no longer be one day but six days in the future. This leaves the closing price P_{t-5}, P_{t-6}, P_{t-7}, P_{t-8} and P_{t-9}.

The neural network has a structure 5-5-1 where, respectively, 5 are the neurons of the input layer, 5 are the neurons for the hidden layer, 1 is the neuron of the output layer. Training is stopped after 500 epochs. There would certainly be room for further training as in previous cases, but to not overtrain the network, it is interrupted early as the training is already good. The total number of parameters amounts to 146.

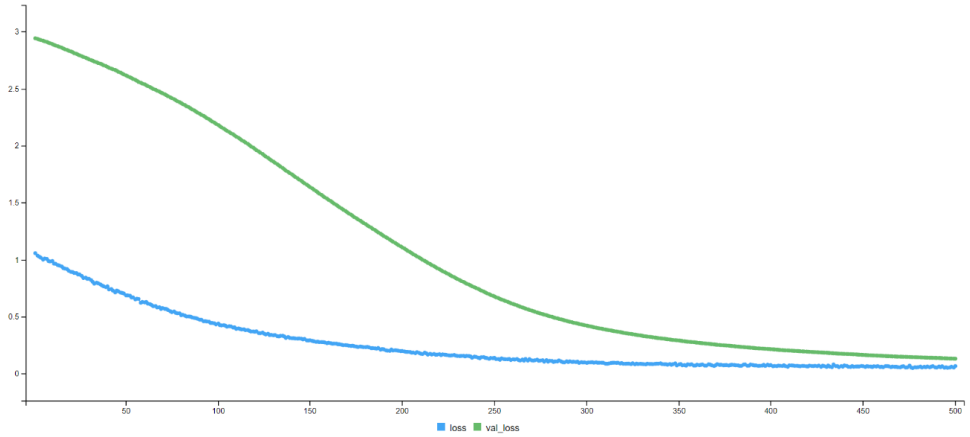


Figure 4.9: 3rd model loss vs val_loss

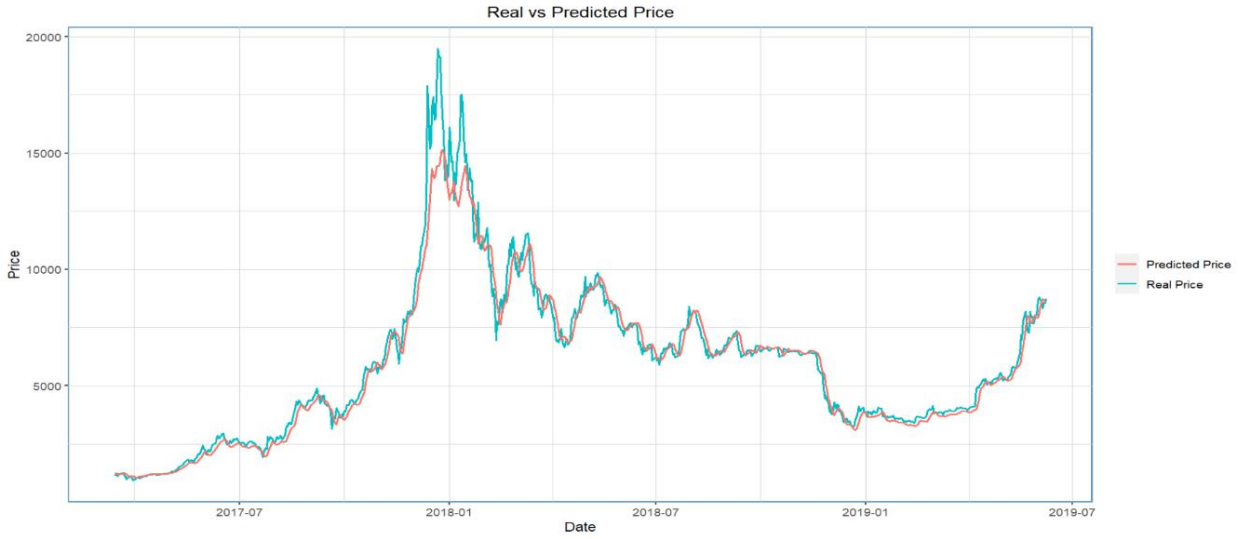


Figure 4.10: 3rd model training results

Also here, the fitted values (red line) fit well and follow the observed data (blue line) for the entire period considered. Considerations are the same as the first model. Some deviation can be seen in the most volatile periods, but the gap between real and predicted values seems smaller if compared to previous models.

Trying to give an initial assessment of the models previously trained, it would seem that the third model performs better than the others. This latter conjecture is based, for the moment, purely on the graphical aspect of the data. A more appropriate verification using reference metrics, such as those listed in Chapter 3.2.1, will be carried out at a later stage. Now take a look at the results of the models when applied to the test set.

4.4.2 Testing Phase

Once the neural network has been trained, the next step is to verify that the model created can also be applied to previously unseen data. For this purpose, the test set has been set aside, which, let's repeat, will be used to forecast prices for the last year after the neural network has been trained using historical data from the previous nearly six years. Below, then, are the results obtained from each model applied to the test set followed by a comment.

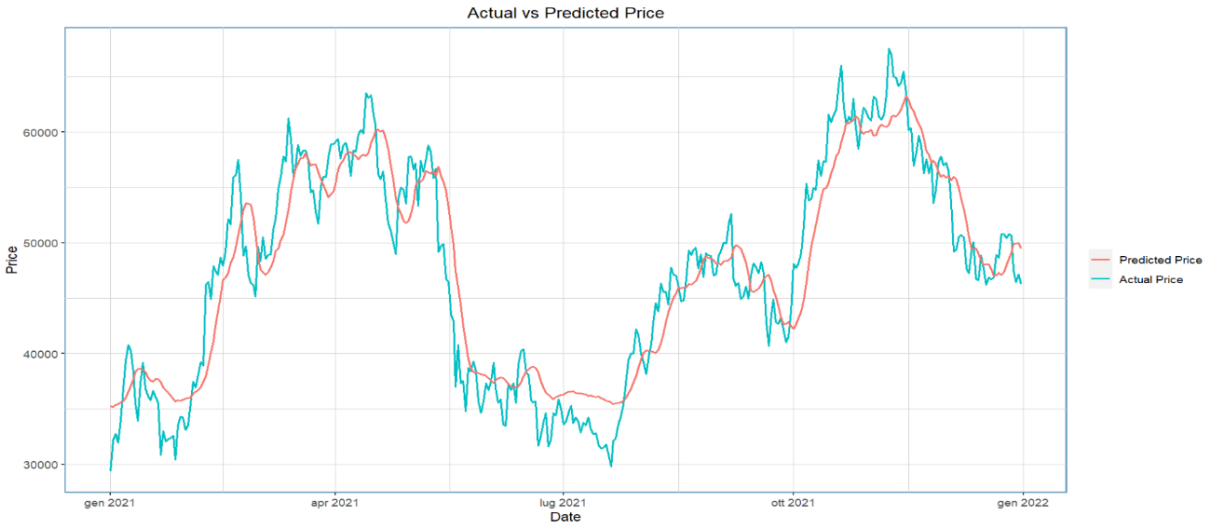


Figure 4.11: 1st model testing results

The first thing to remember is that in this first model forecasts for the next day are made by looking at the previous 7 days. The results seem to follow the general trend fairly well over time, except in some periods, evident especially between January - February and July - August. In addition, the values corresponding to the positive and negative peaks are never reached, so the forecast curve fails to capture the frequent fluctuations. It can also be noted that the forecast curve seems to be delayed, so it does not seem to be appropriate for forecasting, although it fairly correctly predicts the price movements.

Figure 4.12: 2nd model testing results

The second model, on the other hand, makes forecasts for the next day by considering only some of the previous 10 days, more precisely P_t, P_{t-5}, P_{t-6}, P_{t-7}, P_{t-8} and P_{t-9} as mentioned in the previous section. The results seem worse than in the previous model. In general, the trend is quite replicated, but often it is possible to notice that the predicted movement is opposite to the real one. This last aspect is evident in the period between February and June. The only positive aspect, compared to the previous model, is that the amplitude of fluctuations is captured better. Some considerations made for

the first model are confirmed, such as the delay of the forecast curve that in this case is larger. Moreover, the distance between predicted and real values is greater than in the previous model. Consequently, this model also does not seem suitable for forecasting.

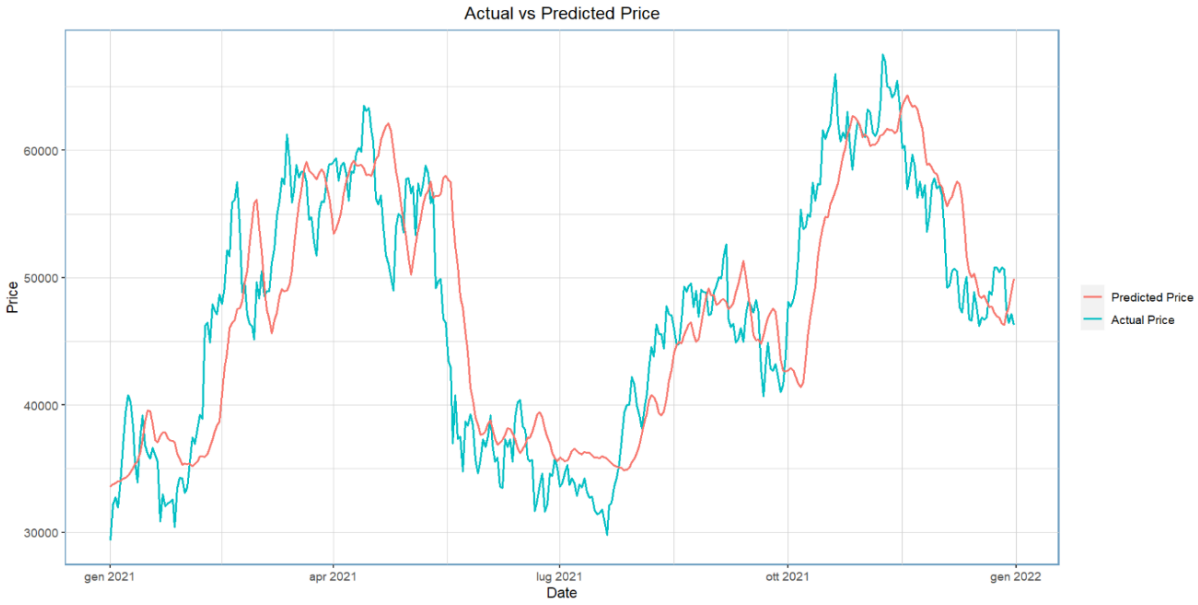


Figure 4.13: 3rd model testing results

The third and final model remains to be analysed. It is the same as the second model except that today's price is removed (P_t), so the forecast will be six days in the future. Results and considerations are the same of the previous model, except that in some periods it manages to capture positive or negative peaks better. Here, too, a consideration made for previous models, namely the delay of the forecast curve, is confirmed.

Now that the results of the three models have been viewed and commented on graphically, empirical evidence is needed to determine which of the three models is the best based on unseen data. The evaluation of the model is done using the Mean Squared Error (MSE) as a reference metric. There follows a table with the values of the MSE obtained for the various models, where a smaller error indicates greater efficiency.

Model	MSE
1st	0.1390288
2nd	0.234281
3rd	0.0732968

Figure 4.14: MSE of the models

According to the results obtained after the calculation of the MSE, we can finally say that the third model is the best. The MSE of the latter is almost half than that of the first model and mose than three times less than that of the second one.

As much as this model may seem to be the best of the three, as I already mentioned, it is easy to see that the results appear to be the same as the original data but delayed. This problem is quite common when approaching the prediction of financial historical series because the algorithm, failing to learn the patterns, tries in every way to minimize the error. Failing to predict what will happen tomorrow, the model shows what the most likely future values will be based on past behaviour. This is the so called random walk hypothesis, a theory that stock market prices are a random walk and cannot be predicted. A random walk is one in which future steps or directions cannot be predicted on the basis of past history.

In conclusion, even this model is not suitable for making predictions. The results, based on the minimization of the error, will lead us to make a choice that is based on the results of previous days. In financial application areas such as this, this can be very problematic. Here is an illustrative example: if I decided to invest $1,000 in Bitcoin today based on my model, which predicted a 5% increase in the closing price today compared to the previous day, my fate would be purely coincidental. That 5% increase, as we said earlier, is based on error minimization, so that price increase is what the asset experienced the day before and not today. Therefore, what will happen today will not be predicted by the model. It could go well or not and financially speaking, such manoeuvres carry a high risk.

CONCLUSIONS

In this paper, a model was proposed which, based on historical data related to the closing price and through the implementation of a deep neural network, can provide a forecast of Bitcoin's closing price. The techniques related to deep learning, have provided important support for the definition of the model, especially in terms of computational time, thanks to a set of tools that have allowed to execute an accurate setting of the hyperparameters that make up the network. This last theme was at the same time the greatest difficulty in defining the model because of the multitude of configurations that can be obtained by modifying the hyperparameters. The current state of affairs has therefore made it necessary to proceed with several test sessions to arrive at a suboptimal solution for all the tested configurations. The models presented in the previous chapter are only a few of the many models which have been studied to arrive at the final result.

The use of deep learning allowed managing a considerable amount of input, capturing for each variable within the dataset non-linear components that were discriminating for the definition of the model. It was found that the exclusion of variables with a high correlation with other input variables leads to a better performance in terms of processing time, as it does less overload the training phase of the neural network.

To evaluate the performance of the three models, they were tested with data never seen by the network in the training phase. As a result, the best model is the third one, where data are used to predict the closing price six days in the future. However, it is evident the problem announced also in the chapter related to the analysis of the result, where the predicted price seems to be very similar to the price of the previous day but delayed to the following day. This result is a fairly common problem when trying to make financial forecasts as it is very complicated to predict the future, especially when dealing with random walk time series and markets as volatile as cryptocurrencies.

Future elaborations may consider the use of other variables or an application other than simple price forecastings, such as the implementation of a trading system or portfolio analysis. This issue, however, must not distract the attention from the fact that neural

networks, and artificial intelligence more broadly, are a powerful, useful tool with a wide range of applications, particularly in the financial field.

BIBLIOGRAPHY

Aydar Mehmet, Cetin Salih Cemil, Ayvaz Serkan, Aygun Betul (2019), *Private key encryption and recovery in blockchain*, Kent State University

Bacao Pedro, Portugal Duarte Antonio, Sebastiao Helder and Redzepagic Srdjan (2018), *Information Transmission Between Cryptocurrencies: Does Bitcoin Rule the Cryptocurrency World?*, Scientific Annals of Economics and Business, vol. 65, no. 2, pp. 97-117

Berentsen Aleksander, Schar Fabian (2018), *A Short Introduction to the World of Cryptocurrencies*, Federal Reserve Bank of St. Louis Review, vol. 100, no. 1, pp. 1-16

Chai Tianfeng, Draxler Roland R. (2014), *Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature*, Geoscentific Model Development, https://doi.org/10.5194/gmd-7-1247-2014

Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall (2018), *Activation Functions: Comparison of Trends in Practice and Research for Deep Learning*, https://arxiv.org/pdf/1811.03378.pdf

Chiu Jonathan, Koeppl Thorsten V. (2017), *The Economics of Cryptocurrencies - Bitcoin and Beyond*, Queen's Economics Department Working Paper no. 1389, pp- 1-55

Chuen David Lee Kuo (2015*), Handbook of digital currency: Bitcoin, innovation, financial instruments, and big data*, Amsterdam: Academic Press - Elsevier

Cong Lin William, He Zhiguo (2018), *Blockchain disruption and smart contracts*, National Bureau of Economic Research, NBER Working Paper no. 24399

Corbet, Shaen, Lucey Brian, Urquhart Andrew and Yarovaya Larisa (2018), *Cryptocurrencies as a financial asset: A systematic analysis*, International Review of Financial Analysis, article in press, pp. 1-18.

De Best Raynor (2021), *Bitcoin average energy consumption per transaction compared to that of VISA as of September 24, 2021,* Statista.com

De Rose Chris (2015), *Behind the ingenious security feature that powers the blockchain*, American Banker, https://www.americanbanker.com/opinion/behind-the-ingenious-security-feature-that-powers-the-blockchain

Dos Santos Renato P., Swan Melanie (2018), *PoW, PoS, & Hybrid protocols: A Matter of Complexity?*, Computing Research Repository (CoRR), May 2018

Nakamoto Satoshi (2008), *Bitcoin: A peer-to-peer electronic cash system*, https://bitcoin.org/bitcoin.pdf

Rohith Gandhi (2018). *A Look at Gradient Descent and RMSprop Optimizers*, https://towardsdatascience.com/a-look-at-gradient-descent-and-rmsprop-optimizers-f77d483ef08b

Staudemeyer Ralf C., Morris Eric Rothstein (2019), *Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks*, Schmalkalden University of Applied Sciences, Germany

Wehle Hans-Dieter (2017), *Machine Learning, Deep Learning, and AI: What's the Difference?*, IBM, July 2017

Xue Ying (2019), *An Overview of Overfitting and its Solutions*, Journal of Physics Conference Series 1168022022, https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022022/pdf

Yaga Dylan, Mell Peter, Roby Nik, Scarfone Karen (2018), *Blockchain Technology Overview*, United States National Institute of Standards and Technology (NIST), https://doi.org/10.6028/NIST.IR.8202

SITOGRAPHY

https://en.wikipedia.org/wiki/Legality_of_bitcoin_by_country_or_territory

https://www.coingecko.com/it/monete/bitcoin/historical_data/usd#panel

https://www.criptoinvestire.com/come-funziona-la-crittografia-nelle-blockchain.html#:~:text=La%20blockchain%20utilizza%20la%20crittografia,e%20di%20una%20chiave%20privata

https://www.datasciencecentral.com/profiles/blogs/artificial-intelligence-vs-machine-learning-vs-deep-learning

https://digiconomist.net/bitcoin-energy-consumption/

https://www.ilsole24ore.com/art/bitcoin-ecco-perche-non-e-moneta-vero-valore-blockchain-AEYilviD

https://www.investopedia.com/terms/p/ptop.asp

https://www.investopedia.com/terms/t/target-hash.asp

https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/

https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

https://towardsdatascience.com/choosing-the-right-hyperparameters-for-a-simple-lstm-using-keras-f8e9ed76f046

https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4

https://www.treccani.it/enciclopedia/crittografia/#:~:text=crittografia%20Tecnica%20di%20rappresentazione%20di,a%20trasformazioni%20che%20lo%20rendano