T.C.
MANİSA CELAL BAYAR UNIVERSITY
GRADUATE SCHOOL

MASTER THESIS
DEPARTMENT OF COMPUTER ENGINEERING

TREND ANALYSES AND PRICE PREDICTIONS OF CRYPTOCURRENCIES BY USING ARTIFICIAL INTELLIGENCE

Coşkun DENİZ

Supervisor
Assoc. Prof. Dr. Bora CANBULA

MANİSA–2024

COŞKUN DENİZ

TREND ANALYSES AND PRICE PREDICTIONS ON CRYPTOCURRENCIES BY USING ARTIFICIAL INTELLIGENCE

2024

COMMITMENT

I declare that this thesis was written in Manisa Celal Bayar University, Graduate School, Department of Computer Engineering in accordance with academic and ethical rules, and all the literature information used is included in the thesis with reference.

Coşkun DENİZ

CONTENTS

LIST of ABBREVIATIONS

AI	Artificial intelligence
ANN	Artificial neural network
BTC	Bitcoin (as a cryptocurrency)
CNN	Convolutional neural network
ESA	Evrişimli Sinir Ağları (Turkish), CNN (English)
ETH	Ethereum (as a cryptocurrency)
FFN	Feed-forward network
DL	Deep learning
DÖ	Derin Öğrenme (Turkish), DL (English)
DNN	Deep neural network
ML	Machine learning
LR, LReg	Linear regression
LSTM	Long short-term memory network
LTC	Litecoin (as a cryptocurrency)
MAE	Mean Absolute error
MLR	Multiple linear regression
MSE	Mean squared error)
MÖ	Makine öğrenmesi
NB	Naive bayes
NN	Neural network
NSA	National Security Agency (of the USA)
RMSE	Root mean squared error
RNN	Recurrent neural network
SLR	Simple linear regression
SHA-256	Secure hashing algorithm-256 (bits)
SVM	Support vector machine
UKSB	Uzun Kısa Süreli Bellek (Turkish), LSTM (English)
XMR	Monero (as a cryptocurrency)
XRP	Ripple (as a cryptocurrency)
TSE	Total squared error
YSA	Yapay Sinir Ağları (Turkish), AI (English)
YZ	Yapay zeka (Turkish), AI (English)

LIST of FIGURES

LIST of TABLES

ACKNOWLEDGEMENT

I would like to express my sincere indebtedness and gratitude to my thesis advisor Assoc. Prof. Dr. Bora CANBULA who made this work possible with his guidance, patience, and effort. His vast wisdom and wealth of experience, beside his spectacular lectures have inspired and encouraged me throughout my studies. Core ideas based on the candle-stick approach in this work have been inspired mostly from his lectures and homework in his precious Data Science classes which I was enrolled.

I would like to express my gratitude thanks also to my dear lecturers in the department for providing me with the studentship that allowed me to learn the necessary knowledge through their valuable classes I was enrolled.

Most importantly, I would like to thank to my dear wife Onur MERDOL DENİZ, without whom this work would not be possible, and our parents & families with all our beloved ones for being heavenly source of love, care, dedication, support, and guidance which enable all.

Coşkun DENİZ
Manisa, 2024

ABSTRACT

M. Sc. Thesis

Coşkun DENİZ

Manisa Celal Bayar University
Graduate School of Natural and Applied Science Institute
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Bora CANBULA

Since the introduction of Bitcoin (BTC) for the first time in 2008 by Satoshi Nakamoto through his white paper, concept of cryptocurrency or block-chain currency where encrypted data string is used in an architecture of decentralized network and transparent ledger has become the basis of today's digital currencies. Today there are more than 25,000 cryptocurrencies like BTC, Ethereum (ETH), Litecoin (LTC), Ripple (XRP), Monero (XMR), etc. and prediction of their future values is a major task. In this work, *trend analyses and price predictions of cryptocurrencies by using artificial intelligence (AI)* is being studied.

AI is a wide-ranging interdisciplinary branch of computer science concerned with building smart machines capable of performing tasks that normally require human intelligence. It involves the ability of machines to replicate or enhance human intelligence, such as reasoning and learning from experience. AI has a main subclass of *Machine Learning (ML)* which has another subclass called *Deep Learning (DL)*. From literature it seems that, beside statistical methods, some certain methods like *Artificial Neural Network (ANN)* and *Linear Regression (LR)* in the ML and *Convolutional Neural Network (CNN)* and *Long Short-Term Memory (LSTM)* in the DL are used to predict future price values of crypto-currencies.

In this work, we study two fundamental AI regression techniques, namely, LR and ANN in two different prediction models to predict the future value of the most popular crypto-currency, BTC, by using different candle-stick datasets obtained from Yahoo-finance websites via Python codes. Next ten upcoming BTC values outside the five different datasets studied here are predicted successfully.

Keywords: Cryptocurrency, Cryptocurrency price prediction, Bitcoin, Bitcoin price prediction, Artificial Intelligence, Machine learning, Deep learning

2024, 74 pages

ÖZET

Yüksek Lisans Tezi

Coşkun DENİZ

Manisa Celal Bayar Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Bölümü

Danışman: Doç. Dr. Bora CANBULA

Bitcoin’in (BTC) ilk kez 2008 yılında Satoshi Nakamoto tarafından beyaz sayfasasında ileri sürülmesinden itibaren, şifrelenmiş veri dizgisinin merkezi olmayan şebeke mimarisi ve şeffaf defter olarak kullanıldığı kripto-para ya da blok-zincir para kavramı günümüz dijital parasının temelini oluşturmuştur. Günümüzde, BTC, Ethereum (ETH), Litecoin (LTC), Ripple (XRP), Monero (XMR), gibi 25,000’in üzerinde kripto-para bulunmaktadır ve bunların gelecek değerlerinin tahmin edilmesi önemli bir araştırma konusudur. Bu çalışmada, *Kripto paraların yapay zeka (YZ, İng.: AI) ile yönelim analizleri ve fiyat tahminleri* araştırılmıştır.

YZ, bilgisayar biliminin normalde insan zekası gerektiren görevleri yerine getirecek kapasitede makinelerin tasarlanması ile ilgilenen geniş yelpazede bir disiplinler arası daldır. YZ, muhakeme etme ve tecrübe ederek öğrenme gibi makinelerin insan zekasını taklit etme ve geliştirme yeteneğini içerir. YZ’nın, *Makine Öğrenmesi (MÖ, İng. ML)* şeklinde alt sınıfı ve onun da *Derin Öğrenme (DÖ, İng. DL)* olarak adlandırılan diğer bir alt sınıfı vardır. Kripto paraların gelecek değerlerinin tahmininde, istatistiksel yöntemlerin yanı sıra, MÖ’nin *Yapay Sinir Ağları (YSA, İng. ANN)* ve *Linear Regresyon (LR)* yöntemleri ile, DÖ’nin *Evrişimli Sinir Ağları (ESA, İng. CNN)* ve *Uzun Kısa-Süreli Bellek (UKSB, İng.: LSTM)* şeklinde bellli bazı yöntemlerin kullanıldığı literatürde görmektedir.

Bu çalışmada, Yahoo-finance web sitelerinden elde edilen mum-çubuk veri setleri kullanılark, en popular kripto para birimi olan BTC’in gelecek fiyat tahmini yapmak üzere, LR ve ANN şeklinde iki temel YZ regresyon yöntemi, iki farklı tahmin modeli üzerinde Python kodlarıyla çalışılmıştır. Çalışılan beş farklı veri setinin dışında kalan on tane gelecek BTC değeri doğrulukla tahmin edilmiştir.

Anahtar Kelimeler: Kriptopara, Kriptopara fiyat tahmini, Bitcoin, Bitcoin fiyat tahmini, Yapay zeka, Makine öğrenmesi, Derin öğrenme

2024, 74 Sayfa

1. INTRODUCTION

Cryptocurrency or block-chain currency is a new form of digital currency where encrypted data strings are used in a decentralized network and a transparent ledger architecture [1-17]. Bitcoin (BTC) is the first cryptocurrency which was originally introduced by someone or some group of people with name (or pseudonym): *Satoshi Nakamoto* through his white paper entitled by *"Bitcoin: peer-to-peer Electronic Cash System"* in October 2008 [3]. Although the writer of this paper or group of people behind this name is still a mystery and the controversy on this hot topic seem to continue much further [1], it is obvious that it has a revolutionary effect in the world's digital financial system. The BTC became the most popular crypto-currency around 2012 among the others (aka altcoin or alternative virtual currencies) whose principles are similar or modified forms of the BTC and its leadership continues since then [1,2,10,11]. As of June 2023, there exist more than 25,000 cryptocurrencies like Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), Ripple (XRP), Monero (XMR), etc., where around 40 of them had a marked capitalization more than $1 billion with the BTC being the pioneer and the most popular among them [1,2, 7-11].

Cryptocurrency, as a digital currency, enables peer-to-peer money transfers between the peers "*in decentralized peer-to-peer network transactions via a transparent ledger architecture"* [1,2]. Although decentralization concept brings about some concerns such as impacting the national economies of countries through breaking national taxing system or other potential misuses like laundering illicit money (or other crime related uses as such) leading cryptocurrencies to be banned in some certain countries [9-13], its decentralized architecture via a shared transparent ledger within all the peers (through smart contracts based on Block-chain technology [13-23]) makes the cryptocurrency today's most trustable and most popular modern currency globally [1,10-14]. However, its price volatility presents a major challenge in predicting its future value for the investors or financiers. From literature, we see that prediction of future value (or price) of cryptocurrencies involves statistical methods, e.g., [24-27] and/or artificial intelligence (AI), e.g., [25-44]. We also see that some of these AI-based methods are applied to the financial data directly to treat them as statistical data, e.g., [25-38], whereas some involves sentiment analyses based on cryptocurrency-based social media data (including Twitter and news) globally by the hypothesis that such data could be used as a measure of tendency of global interest to affect the price of the cryptocurrency [39-43]. In this work, we study the general properties of cryptocurrencies and their future price predictions by the conventional AI methods *via the statistical-financial data* only.

Artificial intelligence (AI) is a multidisciplinary field primarily involving computer sciences to equip machines with the ability to replicate or enhance human intelligence, such as reasoning and learning from experience. Its two main subclasses are machine learning (ML) and deep learning (DL) [44-51]. ML involves three types of learning, namely, *supervised learning*, *unsupervised learning*, and *reinforcement learning* with various techniques like linear regression (LR), artificial neural networks (ANN), support vector machines (SVM), Naive bayes (NB), etc. [44-51]. We also see from literature that *semi-supervised learning* can also be added to the classification of the ML as the fourth one such as in [48-51]. Here we follow the supervised learning in the prediction of cryptocurrency prices where labelled datas are used. ANN mimicks the human brain's neural network (NN) where information data or inputs (referred to as stimulus) are processed in dense layers. As the volume of data and the number of dense layers increase, ANN becomes more complicated and becomes deep neural network (DNN) which constitutes DL. In this work, rather than classification, we study the fundamental regression models within AI (such as LR, ANN, CNN, LSTM, etc.) to predict the future price of cryptocurrencies theoretically.

For the application part of this work, we selected the most famous cryptocurrency, BTC, and studied its future price prediction by the LR and ANN regression models within AI. We use time series analyses with the candle-stick model of the related financial BTC data downloaded from one of the financial web sites as in [29]. Namely, we downloaded data from Yahoo-finance web pages by using the related Python libraries. Our codes are also designed in Python programming language with the use of the related conventional Python libraries like, Numpy, Pandas, Tensorflow, Keras, Scikit learn, etc.

In Chapter 2, we study cryptocurrency concept and the general properties of cryptocurrencies. In Chapter 3, we provide a brief study of the AI and its subfields by focusing on the fundamental regression methods used for price predictons of cryptocurrencies theoretically. In Chapter 4, we study two different future prediction candlestick models to make prediction of the future price of BTC by using five different candlestick datasets (in various richness) obtained from Yahoo-finance pages. Each candlestick involves financial data like “Open (O)”, “High(H)”, “Low(L)”, “Close(C)” values of the related asset (it is BTC here). Our first prediction model involves a relationship (via regression) inside each candlestick data whereas our second prediction model involves a relationship between four adjacent candlestick data. In our first prediction model, we try to establish a relationship between the “C” value and the “O-H-L” values within the same candle. In our second prediction model, we try to establish a relationship between the “C” value of a candle and all the “O-H-L-C” values of the three preceding candles to make a future prediction. In Chapter 5 and Chapter 6, we present the results of our LR and ANN studies, respectively. Our LR and ANN results involve the two studied prediction models (model#1&model&2) in these five diverse datasests (dataset #1, dataset#2, …dataset#5). In Chapter 7 we discuss and conclude these results.

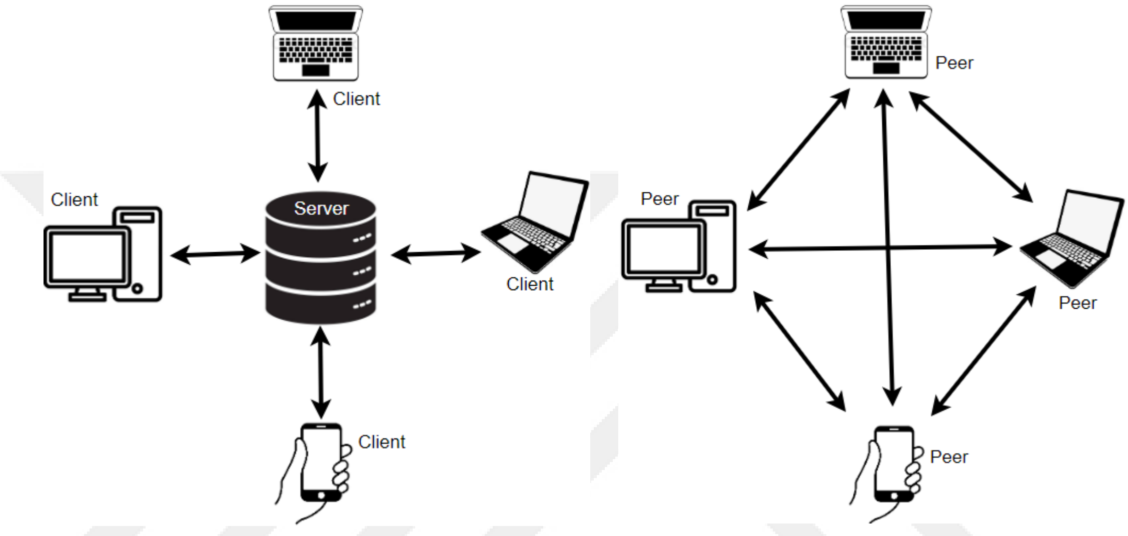
2. CRYPTOCURRENCY AND BLOCKCHAIN TECHNOLOGY

2.1. Cryptocurrency Concept

Cryptocurrency concept was originally introduced with Bitcoin (BTC), which is the first crypto-currency using block-chain technology, through a paper with a name of *Satoshi Nakamoto* in October 2008 [1-3]. Actually, blockchain technology involves numerous current and future applications where *"audit of production or possession where transparent, indelible, and honest record keeping among the peers in a network"* is essential as in applications in healthcare, mass-production, real estate, voting, etc. and the cryptocurrency is one of the typical (and the most fundamental) applications of it in the finance area. The technical mechanisms of blockchain within the context of cryptocurrency will be explored in the next subsection. Regarding cryptocurrency applications of blockchain technology, there exist now more than 25,000 cryptocurrencies using encrypted data with smart contracts being just the same (i.e., in the BTC, Ethereum, etc.) as or similar (i.e., in IOTA, Nano, Byteball, etc.) to the block-chain technology all over the world [10,11]. Consequently, terms *"cryptocurrency"* and *"blockchain currency"* are frequently used interchangeably.

Cryptocurrencies other than BTC are known as *altcoins* or *alternative virtual currencies* where Ethereum (ETH), Litecoin (LTC), Ripple (XRP), Monero (XMR), etc. can be counted among the most popular cryptocurrencies [1,2,10,11,13-18]. Crypto-currency or block-chain currency is an encrypted data string used as a new form of currency, i.e., encrypted digital currency, in contrast to traditional physical currency.

Schematic sketches of the traditional and block-chain client-server architectures are given in Figure 2.1 for comparison. While data is registered and processed in *a centralized-distribution* in the traditional client-server architecture as shown in Figure 2.1.a, in contrast, in the block-chain architecture it is registered and processed in *a decentralized* and *peer-to-peer distribution* as shown in Figure 2.1.b. Since all the records are stored in a centre in the traditional architecture in Figure 2.1a, whereas a non-central (peer-to-peer) interaction is used in the blockhain architecture in Figure 2.1.b, everything in the blockchain is transparent and always open to inspection [16-21]. Consequently, processes on the blockchain are independent of central authorities like banks, individuals, or organizations. It is obvious that traditional client-server architecture is open to manipulation, intervention, or control of the center authority like a bank, a person, a government, an organization, etc. However, in the block-chain system, data is registered in a transparent ledger distributed to all peers instantly with trust ensured through the use of algorithms and smart contracts [1,10,11,14,21-23]. Moreover, there is technically no apparent transaction fee and the transaction cost becomes very small (almost zero) in the block-chain system since there is no center to keep the records to serve the clients in this respect.



a) Traditional centralized client-server architecture

b) Decentralized Blockchain architecture (in peer-to-peer ledger distribution)

Figure 2.1: Schematic sketches of the traditional and blockchain client-server architecture.

When it is studied focusing on Bitcoin, the reasons for the increasing importance and popularity of cryptocurrencies, along with the fundamental outcomes of their block-chain-based structure, can be summarized as follows:

- Transfer of the Bitcoin from one peer to another is easy, cheap, and secure, thus establishing it as a means of payment, or simply, as money.
- Bitcoin is anonymous all around the globe enabling a 7-day/24-hour transfer opportunity within the world banking system without any religious or national holiday.
- Bitcoin is valuable due to its limited availability and the potential for unrestricted price increases.

- Bitcoin can be exchanged for other currencies on demand, which makes it also an investment tool beside being a payment tool.
- Increasing trust through the block-chain technology beside increasing curiosity in Bitcoin with the hope of being rich through the probable instant increase in price (also its risk to decrease) makes it very popular worth considering or studying by the people over the globe.
- Bitcoin is recognized by the governments (but some of the countries like North Korea and China may have restrictions due to political reasons or by considering its influence to their national economies, i.e., [9-13]).
- Popularity in shopping by Bitcoin in big trading sites increases with attractive promotions day by day.
- There is a freedom in paying by Bitcoin with hidden identities of the Bitcoin users feeling secure by the encryption program with no intermediary firm, which makes it a confident and favorite means of payment.
- Being almost not probable to be attacked or hacked due to its encryption (with SHA-256) in its decentralized and distributed structure makes it also a confident means of payment.
- Since the quantity in demand is defined, minting money by the governments to gain financial opportunities or loading inflation to the public is not probable.
- Bitcoin has a currency quantity technically independent of macroeconomic factors.

2.2. Blockchain Technology and BTC

Currently, the use of blockchain technology is predominant in the finance and banking sector but it has also countless novel applications in various fields where all members in a network must reliably agree (against the probable attackers or hackers to prevent forgeries) on an achievement, production, or a decision such as in electronic voting, electronic copyrighting, voting, ownership issues, production tracking systems, etc. where the data (i.e., decisions, achievements, etc.) are registered in a decentralized, transparent, and indelible manner [16-23]. Although the blockchain technology is complex and beyond the scope of this work, it could be useful to overview its basic principles to see how the blockchain system works and transactions occur in its cryptocurrency applications. We will consequently see the power of its encryption via

SHA-256 hashing beside emphasizing the huge potential power of the blockchain technology in various areas other than cryptocurrencies.

Working principle of the blockchain architecture enabling trust and transparency in a *decentralized* and *peer-to-peer distributed ledger* shown in Fig.2.1b is given in Figure 2.2 for cryptocurrency applications from which we can see that transaction data -detailing who is sending (how much money to whom)- is distributed to every client (peer) in the network in an encrypted block. Once a new block is approved, it is added to the end of the chain of previous blocks, thereby extending the blockchain, as the name *"blockchain"* suggests. As the transactions occur, blockchain grows with each added block resulting in an immutable chain.

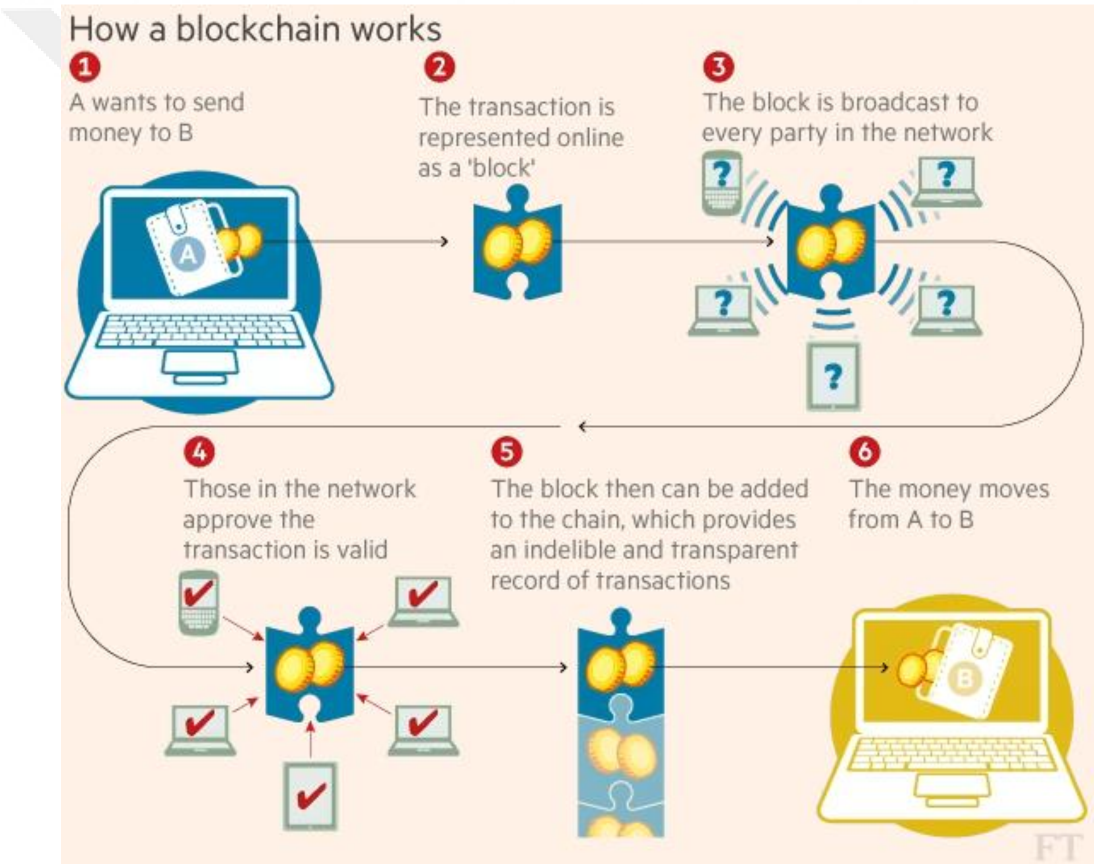


Figure 2.2: Working principle of blockchain [18,19].

A copy of the blockchain, which includes the data, i.e., the ledger here in this cryptocurrency application, is created in a decentralized manner (with the agreement of all the peers) and it is distributed to all of the peers instantly. As of December 1, 2023, the size of the Bitcoin blockchain has reached 530,305.403MB [52] since the first transaction of BTC, which was the distribution of the 50 BTC reward for mining the block to a certain address (which can not be discovered since it was not recorded in the same way as the other blocks), by Satoshi Nakamoto, the pseudonymous creator of the Bitcoin system, on January 3, 2009 [1,3]. Market price (in USD) and Blockchain size graphs of BTC for the last 3 years are given in Fig.1.3 (Graphs taken as a screen-shot of the official web-page of the BTC in [52]). We should also note here the high volatility of the market price of the BTC which makes it a difficult task to predict its future value. Similarly, graphs of market price value, total numbers of BTCs in circulation, exchange trade volume, and total block-chain size of BTC for all times since the first introduction of BTC in 2009, obtained from the official website of the BTC in [52] are given in Figure 2.4.

Figure 2.3: Market price and Blockchain size graphs of BTC for the last 3 years (Graph has been created from the official web-page of the BTC in [52]. *Blue: Blockchain size in MB, Black: Market Price in USD*).

Figure 2.4a: Graph of market price value of BTC as a function of time for all times.

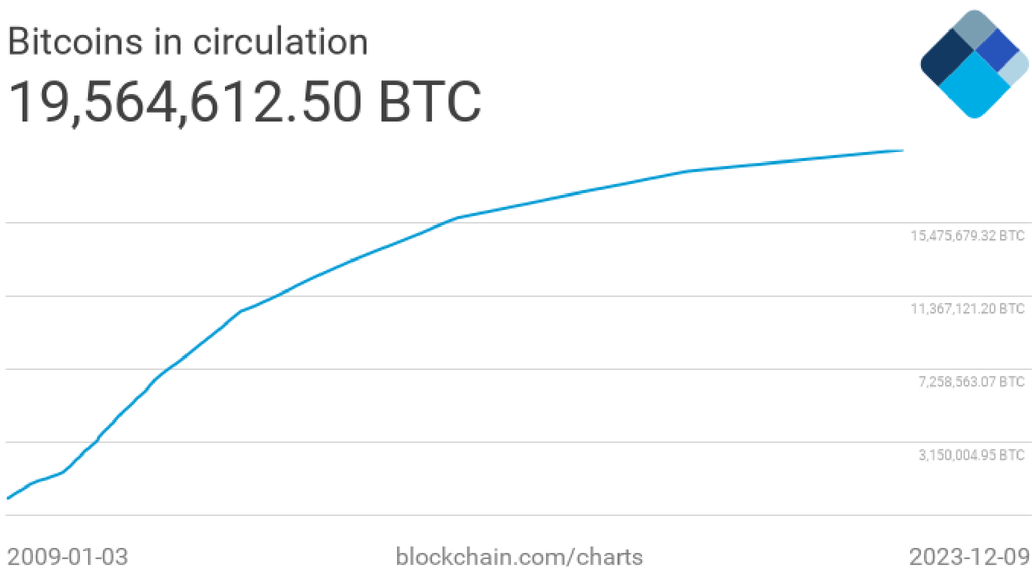


Figure 2.4.b: Graph of total numbers of BTC in circulation as a function of time for all times.

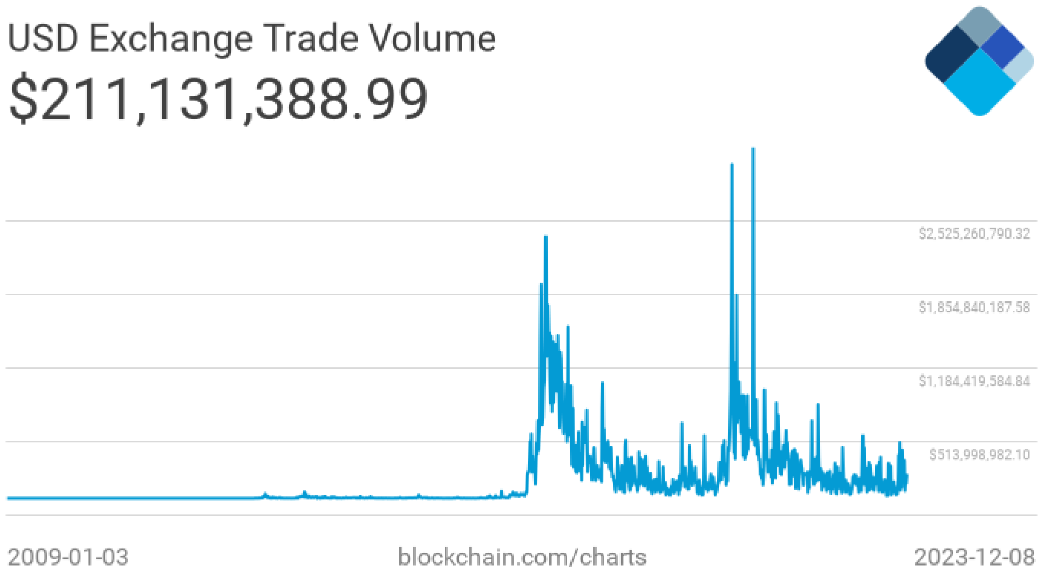


Figure 2.4c: Graph of **e**xchange trade volume of BTC in USD as a function of time for all times.

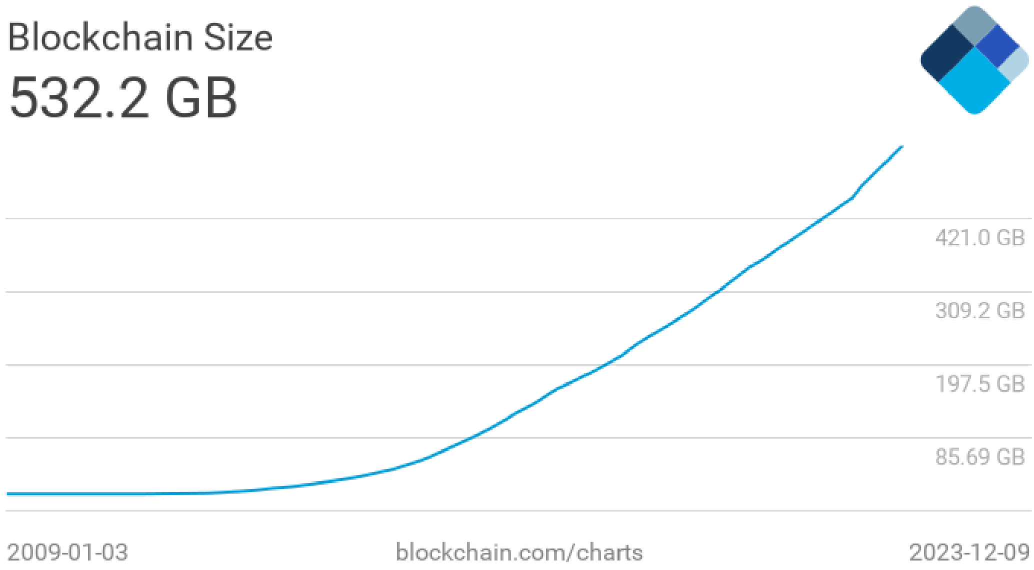


Figure 2.4d: Graph of Blockchain size of BTC as a function of time for all times.

Figure 2.4: Some of the BTC statistical data for all times since the first introduction of it in 2009 (Graphs have been created from the official web-page of the BTC in [52]).

In order to keep secure, each block in the blockchain should involve the encrypted data (the transaction data in our case here for cryptocurrency applications) to work trustworthy against probable attackers or hackers for avoiding forgeries. In deed, each block involves the following information: *i) Data, ii) Hash, iii) Hash of the* *previous block*. A simplified schematic sketch to show the principle of operation is given in Figure 2.5. Each block involves two essential hashes: *i) own hash code of the block, ii) hash code of the previous block* to form a chain as the name suggests, *the blockchain*. Bitcoin uses SHA-256 (Secure hash algorithm in 256 bit) and *nonce* value is the necessary code to produce the hash code in a specific type, such as starting with four zeros as shown in Figure 2.5. A Nonce (acronym of *"number only used once"*) is actually a special number randomly generated to be added to a specific block of the blockchain to produce a unique hash value which satisfies a predetermined level of difficulty. Miners gather transaction data and create a new block header for a block to be added to the chain and they are paid for their work. Once a new block is added to the blockchain, it is distributed to all peers automatically with all the hash values and it becomes indelible and immutable. One may think that a fraud or hijaker can change all the hashes (as a result of changing some data in a specific block) after that block with a great effort to keep the same form (each hash is chained to the next block) but this would produce a faulty blockchain, being obvious to all other peers since the true blockchain is distributed to all peers and democracy applies here to select the majority. *"The 51% attack problem*" saying that *"BTC could be hijacked if miners (hijackers) controlled more than 50% of the network's mining hash rate, which is actually almost zero probability"* is a typical phenomenon in BTC [14,17,22]. Public keys and private keys of the owners play an important role in the secure transaction process as shown in Figure 2.6, which was taken from original paper of Nakamoto in 2009 [3].

The very first block upon which the new blocks in a blockchain are added as time evolves is called *"The Genesis Block"*, also known as *"Block 0"* (rarely, *"Block 1"* is also used in some very early sources). The genesis block is effectively the ancestor of all the blocks in a specific blockchain and each new block is added to the end of the last block of the blockchain via an encryption process as explained above. We can see from the official web site of the BTC that block number of BTC has been reached to an instantaneous value of 820,557 by the date 10th December 2023 whose details taken from the official website of the BTC are given in Figure 2.7.

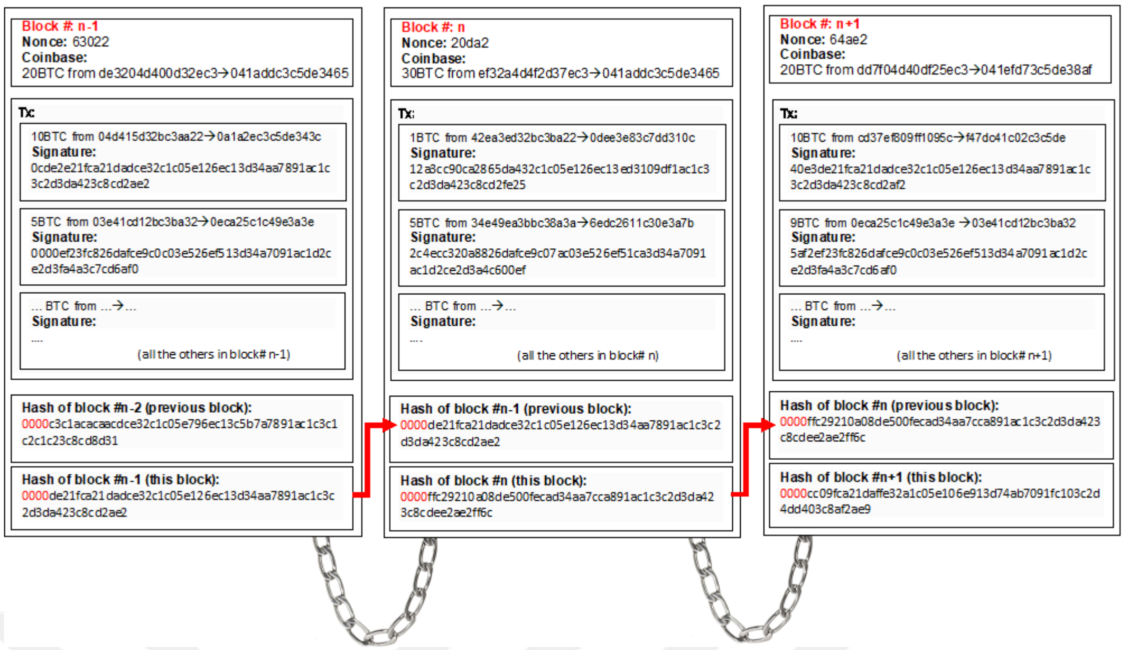


Figure 2.5 A schematic sketch to show the operation principle of BTC via blockchain *(Given values including address and hash values are not real but for illustration).*

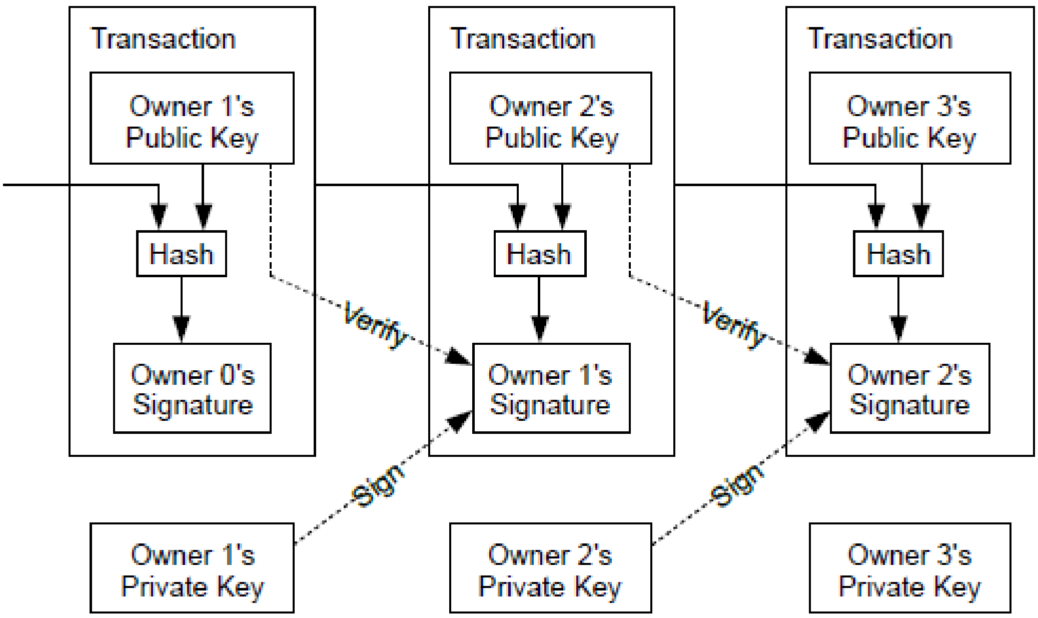


Figure 2.6 Schematic sketch of the BTC system given by Nakamoto in [3].

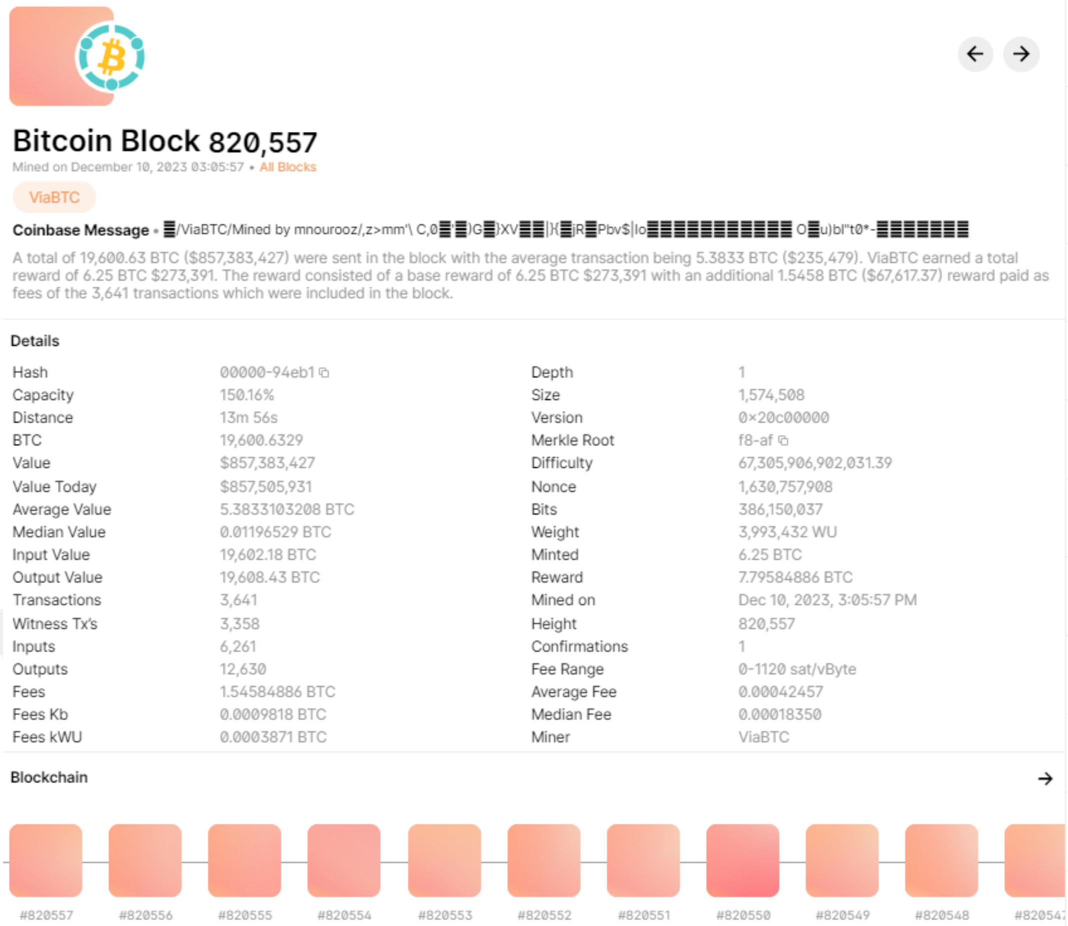


Figure 2.7 Instantaneous data of Block #820,557 added in 10th December 2023 (Figure has been created from the official web page of the BTC in [52]).

When a block is broadcasted to the blockchain, it references the previous block with the related digital signatures and hashes. Then this new block is verified through the smart algorithms and then it is signed (or validated) after cryptographic validation by the miners as explained above. In our Bitcoin blockchain example, miners are rewarded with a certain amount of Bitcoin for their work in adding new blocks to the blockchain.

2.2.1. Hash Code and Encryption by SHA-256 in BTC

For any input string, regardless of length, a unique, finite-length code -akin to a fingerprint- is generated as the hashcode (or *"digest"*, or simply *"hash"*). In the block-chain system of the BTC, SHA-256 (an abbreviation for *"Secure hashing algorithm-256")*, developed by the NSA (National Security Agency) of the USA is used. It generates a 256-bit code that is irreversible, in other words, the original input can not be derived from the SHA-256 code. Probability to break this code is actually $1/2^{256}$, which is almost zero (since the denominator is astronomically a huge number), hence it is almost impossible. Examples of Python codes to generate two words, namely, *"Hello"* and *"hello"* by using the *"hashlib.sha256"* library are given in Figure 2.8. We can see that a small change in the input string (where only the first letter is capitalized in comparison to the other one where it is in lowercase, namely, "H" and "h") changes the generated hash code entirely. Each time we run the codes we will get the same generated ciphertext (SHA-256 codes) for the same input strings but we can not generate the input strings from their SHA-256 codes in the reverse direction. We also note that SHA-256 has 256 bits in binary codes but we generate its hexadecimal form in Figure 2.8 (as in the previous figures) to save space.

```
#Hashing 2 strings ("Hello" and "hello") with hashlib.sha256
import hashlib
my_string1 = 'Hello'
my_string2 = 'hello'
hashed_string1 = hashlib.sha256(my_string1.encode('utf-8')).hexdigest()
hashed_string2 = hashlib.sha256(my_string2.encode('utf-8')).hexdigest()
print('Hello:SHA256 hash-->',hashed_string1,'\n','hello:SHA256 hash-->',hashed_string2)
#Returns:
Hello:SHA256 hash--> 185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969
hello:SHA256 hash--> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
```

Figure 2.8. Python codes to generate hexadecimal SHA-256 hash codes for two strings, namely, for *"Hello"* and *"hello"* strings by using the "hashlib.sha256" library.

2.3. The Most Popular Cryptocurrencies

When it is considered in the scope of global trading volume, market value, popularity, and daily transaction numbers, we can see that the most famous and important currencies among the blockchain currencies (cryptocurrencies) can be counted as Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), Ripple (XRP), and Monero (XMR) [10,11]. It is not much appropriate to line up the top cryptocurrencies in an ascending or descending rigid order since the process is dynamic in a multi-parameter environment. Selected properties of these cryptocurrencies, sourced from Wikipedia pages, are listed in Table 2.1. Cryptocurrencies in this list are selected from the top 50 of the cryptocurrencies (among today's around 25,000 of them) by considering the long-term leadership in popularity via the global trading volume, market value, popularity, and daily transaction numbers. Some other parameters can be preferred as essential indicators leading a change in the order. A common feature among these cryptocurrencies is their encrypted and decentralized architecture, as previously explained. Graphs regarding daily transaction numbers for the three of them, namely, Bitcoin, Ethereum, and Litecoin, between January 2011 and January 2021 are given in Figure 2.9 [53]. The cryptocurrencies in Table 2.1 also show the diversity in the methods used around the common properties, such as in various hash algorithms used to achieve the same goal, namely, *an encrypted and decentralized architecture to maintain a transparent but secure ledger*. To illustrate, SHA-256 is used in the BTC whose fundamental properties have been studied above whereas all the others use other strong hash algorithms. From Figure 2.9 we can see that BTC trading has an increasing *"daily transaction number"* with a great long way in comparison to all other two top cryptocurrencies by year 2021. Some up-to-date financial data of these selected top cryptocurrencies taken from the Yahoo-finance web site by setting the selected items in its Watch-list is given in Figure 2.10 [54] from which we also see that BTC is again leading by a long way also today as also being stated in [10,11].

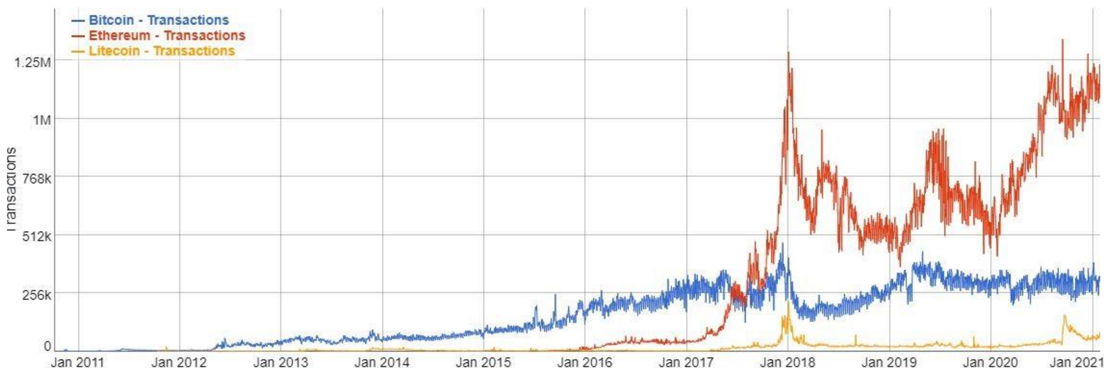


Figure 2.9: Transactions of Bitcoin, Ethereum, and Litecoin per day between January 2011 and January 2021 [53].

Table 2.1: Some properties of the selected top cryptocurrencies (values arranged from the related Wikipedia pages, 2023: https://www.wikipedia.org/)

Property	Bitcoin	Ethereum	Litecoin	Monero	Ripple
Logo					
Symbol	₿ (Unicode: U+20BF ₿ BITCOIN SIGN)	Ether (ETH or Ξ)	Ł	_M_	XRP
Code	BTC, XBT	ETH 2.0 (for Ethereum 2.0 releases)	LTC	XMR	XRP
Original Authors	Satoshi Nakamoto Paper: "Bitcoin: A Peer-to-Peer Cash System", 2008.	Vitalik Buterin Gavin Wood	Charlie Lee	Nicolas van Saberhagen	Arthur Britto, David Schwartz, Ryan Fugger
Developers/ Founders	MIT License	Ethereum Foundation, Hyperledger, Nethermind, OpenEthereum, EthereumJS	Litecoin Core Development Team	Monero Core Team	Ripple Labs Inc.
Initial release	0.1.0 / 9 January 2009	30 July 2015	0.1.0 / 7 October 2011	18 April 2014	June 2012
Stable release	22.0 (13 September 2021)	1.12.2 / 13 August 2023	-	-	1.0.0/15 May 2018
Last release	25.1 / 19 October 2023	1.12.2 / 13 August 2023	0.21.2.1/ 7 June 2022	0.18.3.1 / 7 October 2023	-
Written in	C++	Go, Rust, C#, C++, Java, Python	C++	C++	C++
Lisence	MIT License, Open-source licenses	Open-source licenses	MIT license	MIT license	ISC license
Website	https://bitcoin.org/	http://ethereum.org/	https://bitcoin.org/	http://getmonero.org/	https://ripple.com/
Hash code	SHA-256	Ethash	Scrypt	RandomX	ECDSA
Concen-sus mecha-nism	PoW (Proof of work)	PoW, PoS (Proof of stake)	PoW	PoW	Concensus

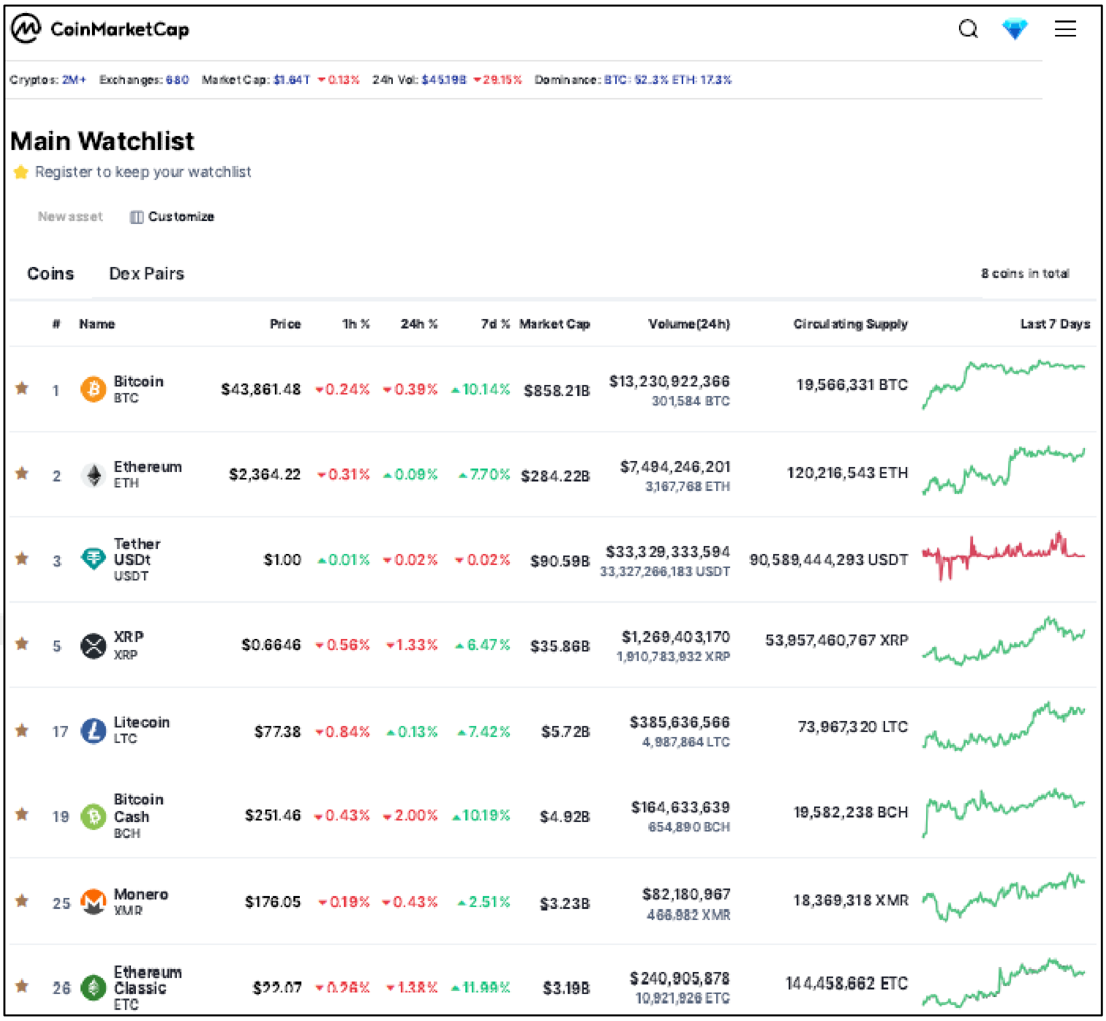


Figure 2.10. Some financial data of the selected top cryptocurrencies taken from the Yahoo-finance web site (selected items in its Watch-list) [54].

2.4. Financial Data, Candlestick Graphs and Prediction of Cryptocurrencies

Due to the increasing reliability and popularity of major cryptocurrencies, many people choose to invest in them, often motivated by the prospect of wealth from potential price increases. People may also be drawn to cryptocurrencies as a means to preserve the purchasing power of their money or assets against inflation or other economic concerns in their country of residence. Other factors, like the economic conditions of a person's country, global crisis, and social media news can also influence their inclination towards cryptocurrencies. Thus, we can say that demand-supply balances in cryptocurrencies are mostly determined by the people's attitudes in buying, selling, or transferring them globally since they are not controlled by any central authority. Moreover, since there are more than 125,000 cryptocurrencies today, increasing trends in the price of some specific cryptocurrencies are obvious to attract people, especially those willing to be rich, by using such specific cryptocurrencies. Investors and financiers must estimate the future value of a specific cryptocurrency to decide whether to buy or sell now, which necessitates the use financial data. However, their nature in being high volatility makes it to be a difficult task. Line and candlestick graphs to show the financial parameters of BTC in USDs for the last one year is given in Figure 2.11 [55].

Figure 2.11: Line and candlestick graphs of price of BTC in USD for the last one year [55] (https://finance.yahoo.com/chart/).

From these graphs we see that fluctuations should be well analysed for this purpose. The same is also true for the other blockchain currencies. Explanation of the candlestick diagrams of cryptocurrencies is given in Figure 2.12. The origin and detailed information on candlestick diagrams with detailed analyses is available in

[56,57]. A green candlestick indicates that its “Close” value is higher than its “Open” value (price increases at the end of the period of the candlestick) whereas a red candlestick indicates that its “Close” value is lower than its “Open” value (price decreases at the end of the period of the candlestick). Since the color tells which level to be “Open” and which level to be “Close” value of the related candle (hence indicating whether price increases or decreases), all the candlestick diagrams in this work are meaningful in color figures as they are given *(i.e., any black-white printed candlestick figure of this work would have strongly missing information).* “High” and “Low” values are the highest and lowest values within the candle period.

We see in the literature that predicting future value of a cryptocurrency by using such past known financial data via statistical and artificial intelligence (AI) methods comes into prominence by increasing work in the field, i.e., [26,32]. Fundamental AI methods used in the prediction of cryptocurrency are studied in the next section.

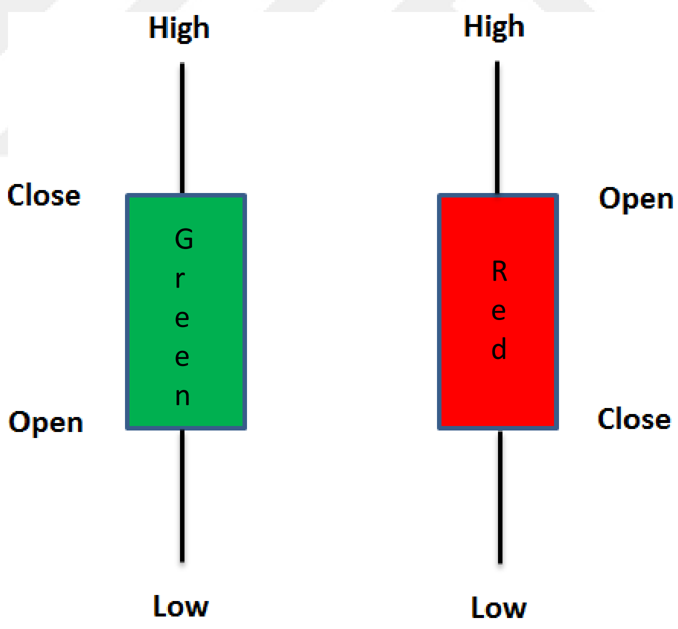


Figure 2.12. Candlestick diagrams used in the cryptocurrency graphs *(Note that colors in figures are important since being informative as explained above).*

3. ARTIFICIAL INTELLIGENCE

As mentioned above, in the prediction of the unknown (future) values and trends of cryptocurrencies from their known financial (past) data, there are statistical methods, i.e., [24-27] and/or artificial intelligence (AI) methods, i.e., [25-38]. Moreover, AI based methods may also involve the addition of other data such as social media data in a sentiment analysis as in [39-43]. In this work, we study the trend analyses and price predictions of the cryptocurrencies in the scope of the BTC by using only financial data via the fundamental AI methods. Origins and fundamental properties of cryptocurrencies are studied in Chapter 2 above. In this chapter we study some essential properties of the fundamental AI techniques used in this work for the price prediction of the top cryptocurrencies.

3.1. Artificial Intelligence (AI) and Fundamental Concepts

Artificial intelligence (AI) is a computer-based multidisciplinary study of mostly computer sciences and others such as mathematics, statistics and data science, philosophy, neurosciences, etc. The purpose in the AI is to make smart machines like human intelligence. It is the general discipline which involves giving ability to the machines to make smart machines by replicating or enhancing human intelligence, such as reasoning and learning from experience. A conventional schematic sketch of contents of artificial intelligence is given in Figure 3.1 from which we see that it has two main subclasses in the following order: machine learning (ML) and deep learning (DL) [44-51] whose techniques are studied in the subsequent sections.

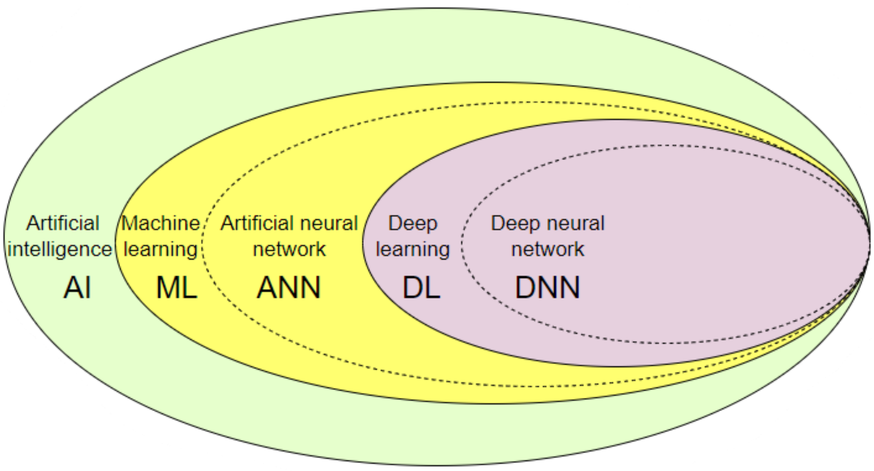


Figure 3.1. A schematic sketch of contents of artificial intelligence (AI) [58].

Machine learning is an up-to-date application of the artificial intelligence based on the idea of necessity on letting machines access the data in real time by allowing them to learn themselves. It is made up of the outermost cluster of the AI as shown in Figure 3.1. Machine learning involves a kind of continuous learning by inferring from the real time inputs obtained from the outside. The artificial neural network (ANN) is a ML technique inspired by the human nervous system. It is a modelling technique analogous to the human nerve system which involves inferring and learning from the real time input data from the outside. Deep neural networks (DNNs) in the innermost cluster of the AI as shown in Figure 3.1 infers using multilayer artificial neural networks which involves advanced techniques within the concept of deep learning (DL).

3.2. Machine Learning (ML) models:

Arthur Samuel first defined ML in 1959 as being *"field of study that gives computers the ability to learn without being explicitly programmed"* [59]. He has the pioneering contribution to the establishment of ML with the idea that programming computers in a way to learn from experience, which implies the ML, could eventually eliminate the need for much of the detailed programming effort. According to Tom M. Mitchell's definition of ML, as being a relatively more recent and formal definition, *"It is the learning of task T, from experience E, with performance measurement P, for a computer program. If performance measured by P enhances on duty T with experience E, then this process can be achieved by the machine learning."* [60]. Although there are various definitions around these two, by combining them we can define the machine learning as follows: *"To program a computer in a way to enable it to do a self learning to perform a task with the experience depending on performance measurement rather than writing codes to perform the task directly."*.

Consequently, a well-defined machine learning involves the following three components: <P, T, E> (where P: performance, T: task, and E: experience). Since the computer learns the necessary experience from a data set in the machine learning approach, it can also be called as a data-based approach. Since data types and their processing play an important role in the ML, ML algorithms are classified according to the learning type from the data set as supervised, unsupervised, and reinforcement types [44,45,47-49]. A conventional schematic description of this conventional classification along with their purpose and application fields, such as regression, classification, etc. are given in Figure 3.2. In our cryptocurrency applications, our purpose is to make financial trend analyses and future price predictions indicating mostly the *supervised learning* in Figure 3.2. We also note that another class referred to as *semi-supervised type*, as a mixture of the supervised and unsupervised, is added to this classification in some of the literature, i.e., [50,51].

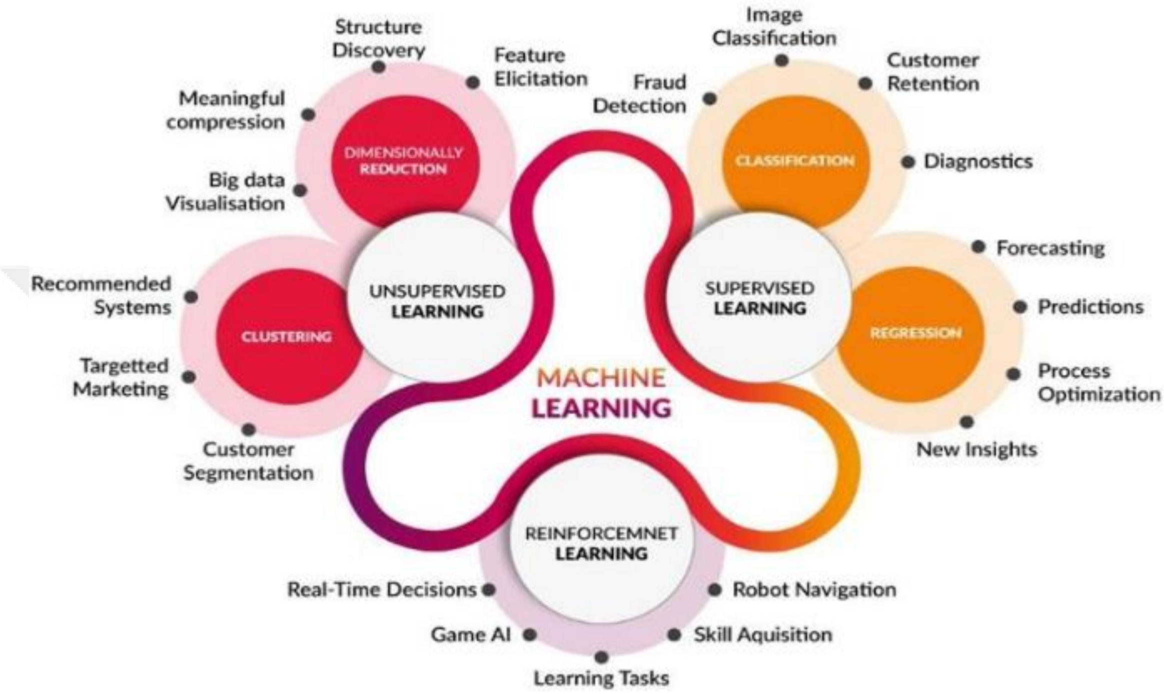


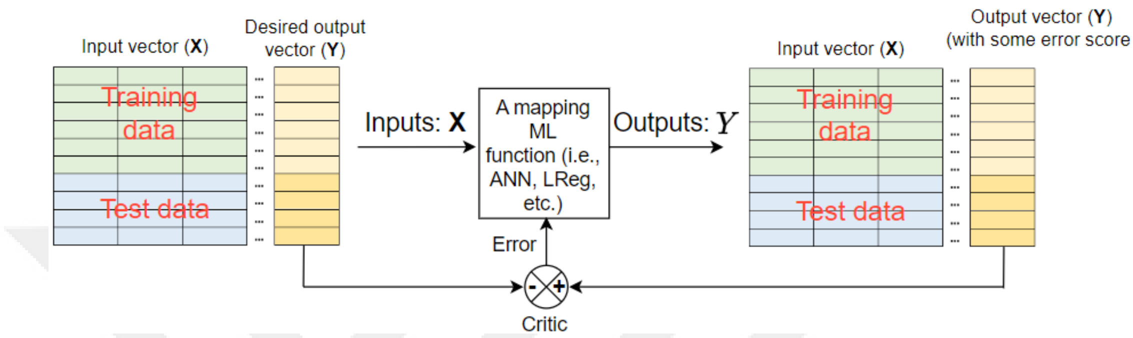
Figure 3.2. A schematic description of machine learning types [60].

Schematic sketches of operation principles of these three basic ML models are given in Figure 3.3. Fundamental properties of these three conventional learning types including the added fourth-one (semisupervised) are listed in Table 3.1. Their general properties can be summarized as follows [44,45,47-51]:

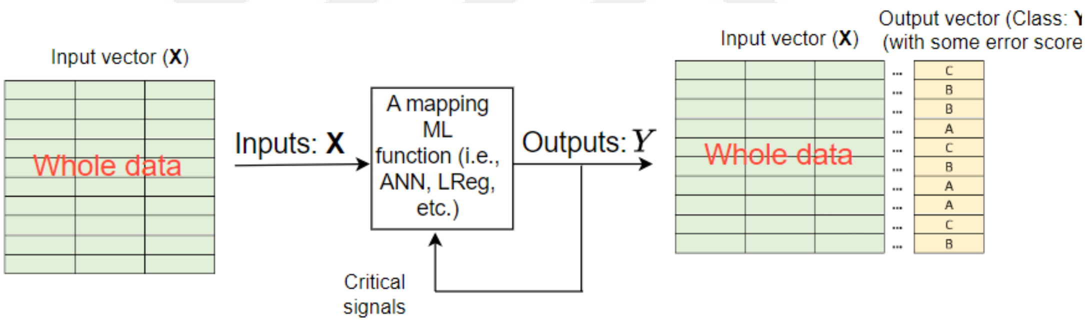
<u>i) Supervised learning:</u> It involves labeled input data to be used in the learning process as shown in Figure 3.3a and it achieves the desired process such as classification or regression. K-nearest neighbour, Naive Bayes, Decision trees, Linear regression, Support Vector Machines: (SVM), ANN, Classification and regression trees, Gradient Boosted Regression Trees, Perceptron Back-Propagation, and Random Forest can be given as the typical examples of the supervised learning.

<u>ii) Unsupervised learning:</u> It involves unlabeled data in the learning process as shown in Figure 3.3b. Due to the unlabeled data, there is no predefined features assigned to each data. Therefore, it can not learn directly from the data but it achieves

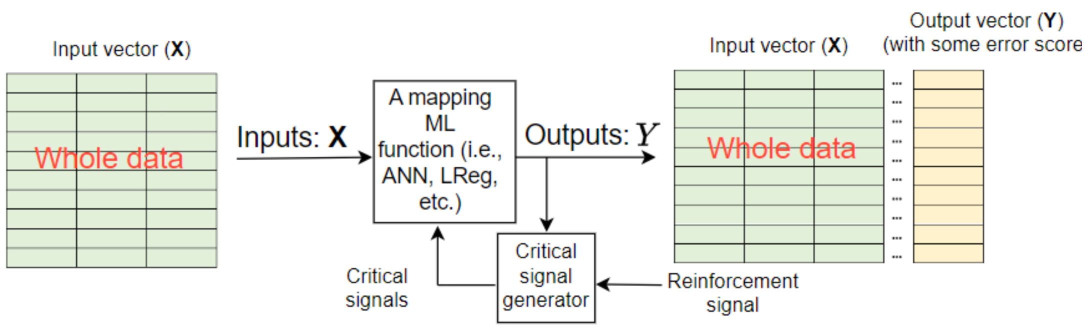
the learning process by studying the data in detailed analyses in this relatively more difficult task by some complicated methods such as searching similarities (or distances) or using size reduction techniques to infer reasonable meanings to extract information to classify them (in other words clustering) from the initially unlabeled data set. K-means clustering and association rules can be given as the typical samples of the fundamental algorithms in the unsupervised learning.



(a) Supervised learning.



(b) Unsupervised learning



(c) Reinforcement learning.

Figure 3.3. The ML models.

iii) Reinforcement Learning: It relies on a principle to do the necessary step to maximize the reward in a special condition. In the reinforcement learning, there is no supervisor but only real numbers and reward (or reinforcement) signal as shown in Figure 3.3c. Pavlov's dog is a classical example given for the reinforcement learning. Since it involves sequential decision, time plays an important role. Feedback is always delayed not sudden. Behavior of the agent determines the data taken in the next step. Markov Decision Process: MDP and Q-learning can be given as the typical samples of the reinforcement learning.

iv) Semisupervised learning: It involves both unlabeled and labeled input data condition where they are used in learning process. It is generally applied to the systems involving little labeled and lots of unlabeled data. Labeled ones are processed with supervised education to complete the inferences and education, losses are calculated and network weights are updated according to the gradient declining. Unlabeled data is labeled as pseudo-label and used to obtain more efficient and correct results. More success is obtained in semisupervised learning rather than unsupervised learning. Linear regression and logistic for the added unlabeled data can be given as the typical samples of the fundamental algorithms in semisupervised learning.

Table 3.1. Types of machine learning

	Overview	**Process**	**Subtypes**	**Methods**	**Examples**
Supervised Learning	-Majority of algorithms -Machine is trained using well-labeled data; inputs and outputs are matched	Mapping function takes inputs and matches to outputs, creating a target function.	-Classification -Regression	-Linear /Polynomial regression -Random forest -For classify-cation: KNN, Trees, Logistic Regression, Naive Bayes, SVM	-Medical imaging -Identity fraud detection -Customer retention -Image classification -Diagnostics -Blockchain currency
Unsupervised Learning	-Unlabeled data (inputs only) is analyzed. -Learning happens without supervision.	Inputs are used to create a model of the data	-Clustering -Association -Dimensi-onality reduction	-PCA -k-Means, -Hierarchical clustring	-Market basket analyses -Big data visualization -Meaningful compression -Structure discovery
Semi supervised Learning	Some data is labeled, some not.	Combination of above process (for	All the above	-Self training -Mixture models,	-Lane-finding on GPS data

	-Goal: better results than labeled data alone. -Good for real world data	supervised and unsupervised learning)		-Semi-supervised SVM.	-Speech recognition -Web content classification Text document classification - Protein Sequence Classification
Reinforcement Learning	-No predifined data -Learn through interaction with environment -Learn optimal strategy from experience Adaptable to changes through exploration	Behavioral learning: An agent interacts with its environment by performing actions & learningf from errors or rewards (reward based). It also learns from mistakes. It follows trial-and-error method	-Model-based methods -Model-free methods: i) value-based methods (Deep networks to represent value, Q-learning), ii) Policy-based methods (Neural network policies, more complex policy search)	-MDP (Markov Decision Process) -Deep Mind	-Optimized marketing -Driverless cars -Robot navigation -Skills acquisition -Game AI

3.3. Linear regression model in ML

A regression model of experimental data, i.e., a scatter-plot data, is a mathematical equation to establish a relationship between one (or more) independent variable(s), say, x (or x_i with index $i > 1$) and one dependent variable, say, y, as shown in Figure 3.4. If there is only one variable (x), it is called a simple regression and if there are more than one independent variables (x_i with $i > 1$), it is called multivariable regression. Moreover, if the relationship to be established is linear, it is called a linear regression and if it is non-linear, it is called a non-linear regression. Consequently, for a simple linear regression (SLR) model, we have the followings [29,62-65]:

Real relationship (between x and y): $$y = \beta_0 + \beta_1 x + \epsilon \quad (3.1a)$$

Regression line (or regression model): $$E(y) = \beta_0 + \beta_1 x \quad (3.1b)$$

Estimated relationship (via our regression model): $$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \epsilon \quad (3.1c)$$

Estimated regression line (in our regression model): $$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \quad (3.1d)$$

where x is the independent variable, y is the dependent variable, u is the error term (stochastic variable) which is not included to our linear regression model (since it cannot be determined), and β_0&β_1 are the regression coefficients where β_0 is the y-intercept (at $x = 0$) and β_1 is the slope of the regression line, respectively. Moreover, $\hat{\beta}_0$&$\hat{\beta}_1$are the estimated (or predicted) values of β_0&β_1. Similarly, for a multiple linear regression (MLR) model, we have the followings:

Real relationship (between x and y):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon = \left(\sum_{i=0}^{n} \beta_i x_i\right) + \epsilon \qquad (3.2a)$$

Regression line (or regr. model):

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n = \sum_{i=0}^{n} \beta_i x_i \qquad (3.2b)$$

Estimated relationship (via our regr. model):

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_n x_n + \epsilon = \left(\sum_{i=0}^{n} \hat{\beta}_i x_i\right) + \epsilon \qquad (3.2c)$$

Estimated regression line (or regr. model):

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_n x_n = \sum_{i=0}^{n} \hat{\beta}_i x_i \qquad (3.2d)$$

from which we see that, formulas in equations (3.2a)-(3.2d) for MLR involves the equations (3.1a)-(3.1d) for the SLR for $n = 1$.

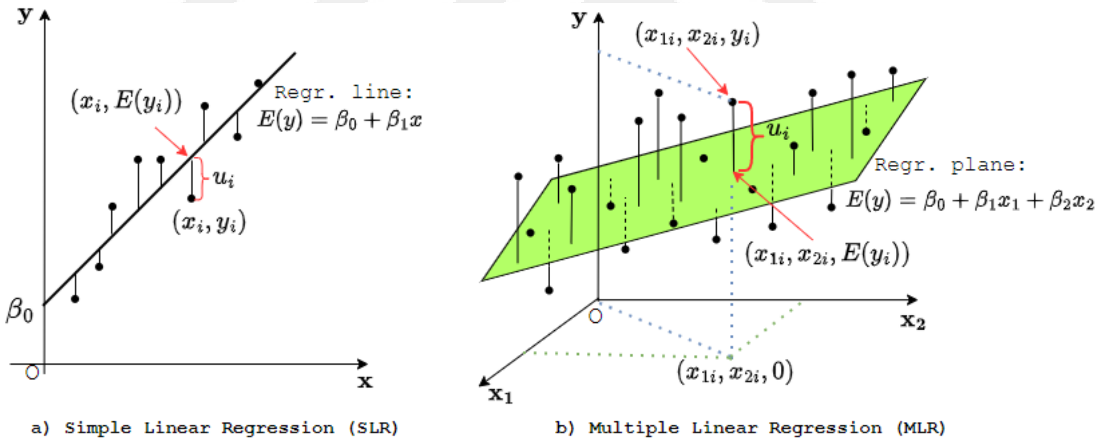


Figure 3.4. Geometrical interpretation of Simple Linear regression (SLR) and Multiple Linear Regression (MLR) models.

Consequently, the regression coefficients can be obtained by minimizing the total-square error (TSE) as follows [47,62]:

$$\hat{y} \approx y \Leftrightarrow (TSE)_{min} = \left(\sum_{i=0}^{n} e^2\right)_{min} = \left[\sum_{i=0}^{n} (y_i - \hat{y}_j)^2\right]_{min} =$$
$$\left[\sum_{j=0}^{m} \left(y_j - \sum_{i=0}^{n} \hat{\beta}_i x_{ij}\right)^2\right]_{min} \qquad (3.3)$$

which implies that

$$\frac{\partial}{\partial\widehat{\beta}_i}\sum_{i=0}^{n} e^2 = 0 \tag{3.4}$$

where x is the independent variable and y is the dependent variable, β_0 & β_1 are regression coefficients, and ϵ is the random error term in our simple linear regression model. Here the input data is the independent variable which does not depend on any variable as the name says and the output data are the dependent variables which depend on the input variables (or independent variables). *A simple regression model* involves only two variables: one independent (x) and one dependent variable (y) whereas *a multivariable regression model* with n numbers of inputs involves n numbers of independent variables ($\boldsymbol{X} = (x_1 \quad x_2 \quad \cdots \quad x_n)^{\boldsymbol{T}}$) and one dependent variable ($Y = (y_1 \quad y_2 \quad \cdots \quad y_n)^{\boldsymbol{T}}$)). Consequently, our SLR and MLR models can be expressed in the matrix notation as follows [47,62]:

$$SLR: \begin{bmatrix} y_1 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix} \Leftrightarrow \boldsymbol{Y}_{n\times 1} = \boldsymbol{X}_{n\times 2}\beta_{2\times 1} + \epsilon_{n\times 1} \tag{3.5a}$$

$$MLR: \begin{bmatrix} y_1 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,p-1} \\ 1 & x_{1,2} & x_{2,2} & \dots & x_{2,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1,n} & x_{2,n} & \cdots & x_{n,p-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$$\Leftrightarrow \boldsymbol{Y}_{n\times p} = \boldsymbol{X}_{n\times p}\beta_{p\times 1} + \epsilon_{n\times 1} \tag{3.5a}$$

Where p is the number of parameter estimates. Here, $\boldsymbol{X}$ is called the design matrix, β is the vector parameters, ϵ is the error vector, and $\mathbf{Y}$ is the response vector. For the estimated (or predicted) values we use a hat on the related symbols, namely,

$$\boldsymbol{Y} \to \widehat{\boldsymbol{Y}} \text{ and } \beta \to \hat{\beta} \tag{3.6}$$

as used in the equations above to give the residuals and the error metrics MAE (Mean Absolute error), MSE (Mean squared error), and the RMSE (Root mean square error) and R^2 formulas become as follows [29, 63]:

$$\text{Residuals: } \boldsymbol{e} = \boldsymbol{Y} - \widehat{\boldsymbol{Y}} \tag{3.7a}$$

$$MAE = \frac{1}{n}\sum_{i=0}^{n}|y_i - \hat{y}_i| \tag{3.7b}$$

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2 \tag{3.7c}$$

$$RMSE = \sqrt{ME} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2} \tag{3.7d}$$

$$R^2 = 1 - \frac{\sum_{i=0}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n}(y_i - \bar{y})^2} \tag{3.7e}$$

where $\bar{y}$ is the average value of y.

3.4. L1 and L2 Regularization

In any multiple regression model of the ML, labeled data is split into randomly selected train and test datasets. Typical test-train percentage values are (20-80) %, (30-70) %, etc. Regression model tries the train data set to determine the regression coefficients ($\hat{\beta}_{i\neq 0}$) and intercept value ($\hat{\beta}_0$) to estimate a regression line. Although the regression line determined from the train dataset during training process is perfect, this regression line can not be a best (or optimum) choice for the whole dataset when we use the test data to make prediction by the determined regression line as shown in Figure 3.5. This process is called as *overfitting*. To illustrate with a typical example, a 90% or above accuracy for the training testing data but an accuracy value around or less than 65% strongly indicates the overfitting problem.

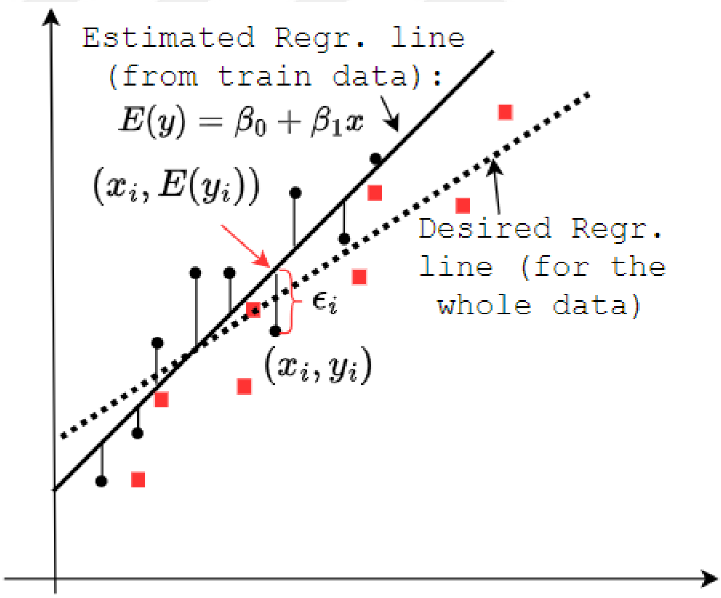


Figure 3.5. A schematic sketch of the overfitting problem in a ML regression process *(Black dots: train data, red squares: test data).*

In order to prevent overfitting, regularization techniques are used. Here, L1 regularization technique is called *Lasso regression (Least Absolute Shrinkage and Selection Operator)* and L2 regularization technique is called *Ridge regression*. The key difference between the two is the penalty term. Cost functions to be minimized in these two types are as follows [51]:

$$L1\ (Lasso\ \ Reg.): cost = RSS = \sum_{i=0}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\left|\beta_j\right| \qquad (3.8a)$$

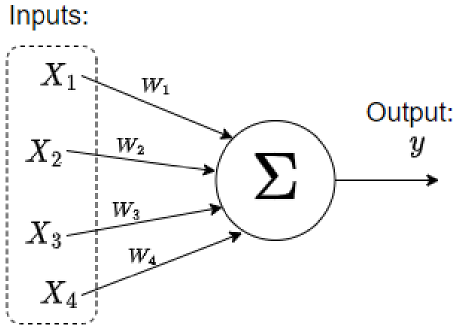
$$L2\ (Ridge\ \ Reg.): cost = RSS = \sum_{i=0}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\beta_j^2 \qquad (3.8b)$$

where RSS is the residual sum of squares, λ is the *Lagrange multiplier*, and the second terms with lambdas on the right-hand side of the equations are penalty terms added to the normal cost function to compensate the overfitting problem. We see from these formulas that the Lasso Regression adds the absolute value of magnitudes of the coefficients as the penalty term to the loss function whereas the Ridge regression adds the square of the coefficients. Since these penalties are associated with the norm concepts where *norm1* (or *L1-norm*) involves the length of a vector (absolute value) and *norm2* (or *L2-norm*) involves the square of a vector, these regularization techniques are also called as *L1 and L2 regularization*.

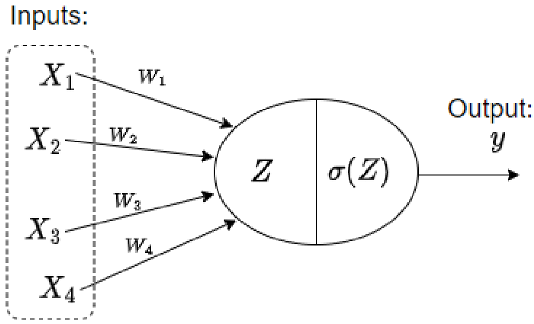
3.5. Artificial Neural Networks (ANNs) and Deep Neural Networks (DNNs)

From Figure 3.1 we see that ANN is a technique in the ML and below it there exists Deep Neural Networks (DNNs) in the field of Deep Learning (DL). ANNs is a special and novel regression technique in the ML inspired from the nerve system of the human brain. ANNs have high learning capability to learn the nonlinear relationship among the variables to recognise the relationships in high degree. ANNs can be applied with both supervised and unsupervised learning [66]. The first neural network model involving only a single neuron is the *Perceptron* which was developed by Frank Rosenblatt in 1950s for the first time [67]. The most fundamental neuron model in today's studies is called the *Sigmoid neuron model* which is again in the form of a single layered structure like Perceptron but in a developed version. Schematically illustrated figures for a perceptron and a sigmoid neuron with three inputs are given comparatively in Figure 3.6.

In perceptron, there exist little numbers of inputs and these inputs are multiplied by the associated weight values (w) to be summed up to find its z value by finally adding a bias (b) which can have zero values. Then a binary output is produced in such a way that it is logic one if that calculated z-value is above a specific level or logic zero if it is below that level. As to the Sigmoid neuron, inputs are not binary but real numbers between 0 and 1 and the z-value is produced similar to the perceptron by multiplying them by the weight functions and summing them up and finally by adding bias (b). But the produced z-value, which is a real number, is transferred by the sigmoid function, namely, in the form σ(z), to produce output y.



(a) Perceptron



(b) Sigmoid neuron

Figure 3.6. Sigmoid neuron and perceptron models.

Formulas of the output functions for the Perceptron and the Sigmoid neuron are as follows:

$$z(X_i) = \sum_i^n w_i X_i + b, X_i = 0 \; or \; 1 \Longrightarrow y_{SN}(X_i) = \begin{cases} 0, if \; z(X_i) < c \\ 0, if \; z(X_i) \geq c \end{cases} \quad (3.9a)$$

$$z(X_i) = \sum_i^n w_i X_i + b, \;\; 0 \leq X_i \leq 1 \Longrightarrow y_P(X_i) = \; \sigma(z) = \frac{1}{1+e^{-z}} \quad (3.9b)$$

Here there are n numbers of inputs X_i where w_i being the weight value for input X_i (with index *i*), number *b* being the bias value between 0 and 1, and σ(z) being the *Sigmoid function*. Here, the reason for using a nonlinear Sigmoid function is to prevent the limited learning capacity occurring normally in the linear function application case for the solutions of the nonlinear problems. ANNs are designed similar to the Sigmoid neuron but in a multilayered form and beside the sigmoid function, some other nonlinear transferring functions like hyperbolic tangent function, ReLU (Rectified Linear Units) function, Leaky RELU function, Softmax function, etc. are frequently used. Some of them (i.e., Hyperbolic tangent, ReLU, etc.) are in a normal function type but some can be in a *logit* form such as *Softmax* as as shown in Table 3.2 below. *Logits* are not like normal functions which we can describe in the form of outputs corresponding to independent input variables to be described by $y = f(z)$. They are defined as the raw score values produced by the last layer of the neural network before applying any activation function on it, consequently they can be expressed in the form of $y_i = g_i(\vec{z})$ as shown in Table 3.2. Fundamental nonlinear function-type activation functions used in the ANN architecture are given in Figure 3.7. and the structure of the layers in the ANN architecture is given in Figure 3.8.

Table 3.2. Some of the activation functions used in the ANN architecture

Name	**Mathematical formula**	**Derivative**	**Range**
Identity	$g(z) = z$	$g'(z) = 1$	$(-\infty, \infty)$
Binary step	$g(z) = \begin{cases} 0, if\ z < 0 \\ 1, if\ z \geq 0 \end{cases}$	$g'(z) = 0$	$\{0,1\}$
Hyperbolic tangent	$g(z) = Tanh(z) = \dfrac{e^z - e^{-z}}{e^z + e^{-z}}$	$g'(z) = 1 - [g(z)]^2$	$(-1, +1)$
Sigmoid*	$g(z) \coloneqq \sigma(z) = \dfrac{1}{1 + e^{-z}}$	$g^{\prime(z)} \coloneqq \sigma^{\prime(z)} = \sigma(z)[1 - \sigma(z)]$	$(0,1)$
ReLU (Rectified linear unit)	$g(z) := (z)^+ = \begin{cases} 0, if\ z \leq 0 \\ z, if\ z > 0 \end{cases} = max(0, z) = z\mathbf{1}_{z>0}$	$g'(z) = \begin{cases} 0, if\ z < 0 \\ 1, if\ z > 0 \end{cases}$	$[0, \infty)$
Softplus	$g(z) = ln(1 + e^z)$	$g^{\prime(z)} = \dfrac{1}{1 + e^{-z}} =: \sigma(z)$	$(0, \infty)$
LReLU (Leaky ReLU)	$g(z) = \begin{cases} 0.01z, if\ z \leq 0 \\ z, if\ z > 0 \end{cases}$	$g'(z) = \begin{cases} 0.01, if\ z < 0 \\ 1, if\ z > 0 \end{cases}$	$(-\infty, \infty)$
PReLU (Parametric ReLU)	$\begin{cases} \alpha z, if\ z < 0 \\ z, if\ z \geq 0 \end{cases}$	$g'(z) = \begin{cases} \alpha, if\ z < 0 \\ 1, if\ z \geq 0 \end{cases}$	$(-\infty, \infty)$
SiLU**(Sigmoid linear unit)	$g(z) = \dfrac{z}{1 + e^{-z}}$	$g'(z) = \dfrac{1 + e^{-z} + ze^{-z}}{(1 + e^{-z})^2}$	$[-e, \infty)$
ELU (Exponenial linear unit)	$g(z) = \begin{cases} \alpha(e^z - 1), if\ z \leq 0 \\ z, if\ z > 0 \end{cases}$	$g'(z) = \begin{cases} \alpha e^z, if\ z < 0 \\ 1, if\ z > 0 \end{cases}$	$(-\alpha, \infty)$
SELU (Scaled exponential linear unit)	$g(z) = \lambda \begin{cases} \alpha(e^z - 1), if\ z \leq 0 \\ z, if\ z > 0 \end{cases}$	$g'(z) = \lambda \begin{cases} \alpha e^z, if\ z < 0 \\ 1, if\ z > 0 \end{cases}$	$(-\lambda\alpha, \infty)$
Gaussian	e^{-z^2}	$g'(z) = -2ze^{-z^2}$	$(0,1]$
Softmax	$g_i(\vec{z}) = \dfrac{e^{z_i}}{\sum_{j=1}^{J} e^{z_j}}, for\ i = 1,2, \ldots, J$	$g'(z) := \dfrac{\partial g_i(\vec{z})}{\partial z_j} = g_i(\vec{z})[\delta_{ij} - g_j(\vec{z})]$	$(0,1)$
Maxout	$g_i(\vec{z}) = \max\limits_i z_i$	$g'(z) := \dfrac{\partial g_i(\vec{z})}{\partial z_j} = \begin{cases} 1, if\ j = arg \max\limits_i z_i \\ 0, if\ j \neq arg \max\limits_i z_i \end{cases}$	$(-\infty, \infty)$

**Sigmoid* is also referred to as *logistic* or *soft step*.

***Sigmoid Linear unit* is also referred to as *Sigmoid shrinkage* or *Swish-1*. Its abbreviation can also be used in SiL rather than SiLU [68].

The ANN architecture can involve more than one neuron in a layered form as shown in Figure 3.8. Outermost layers are called *input layer* (for the layer on the left) and *output layer* (for the layer on the right) and the layers between them is called the *hidden layers*. To illustrate, in Figure 3.8 there are three hidden layers. Inputs X_i entering via the input layers are processed from left to the right through the hidden layers to be transferred to the output layers where the calculated output value z_j is calculated. Numbers of the hidden layers are determined from the properties of the problem and the design structure.

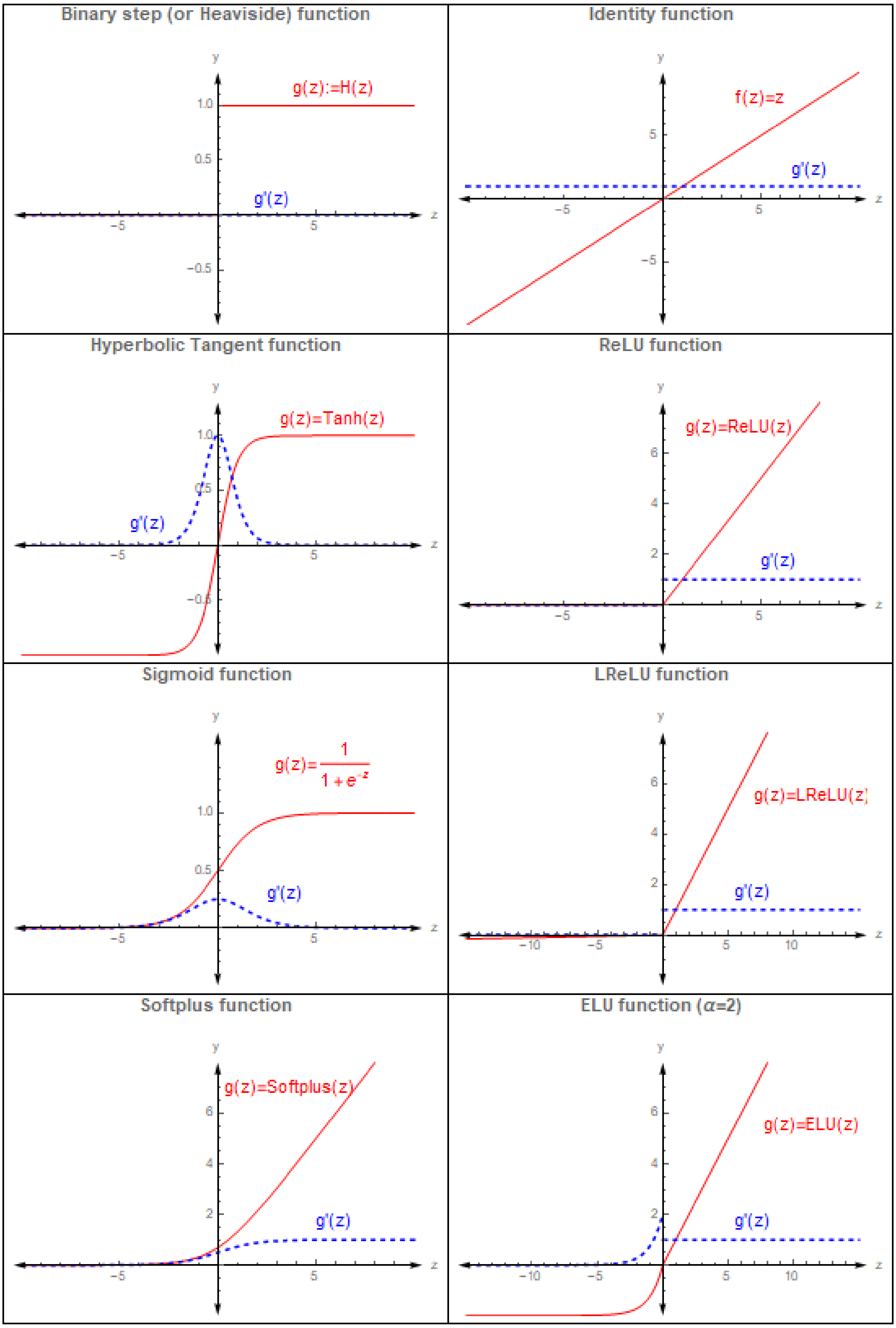


Figure 3.7. Graphs of the fundamental activation functions used in the ANN.

In literature, the input layer is not added to the whole layer number while it is mentioned and according to this rule, the ANN shown in Figure 3.8 is called a four-layered ANN. In Figure 3.9, there is a single output ($j = 1$) but there could be more than one output in an ANN. Although it is made up of Sigmoid neurons, from time to time it can be called as *MLP (Multi-Layer Perceptron*) in the literature.

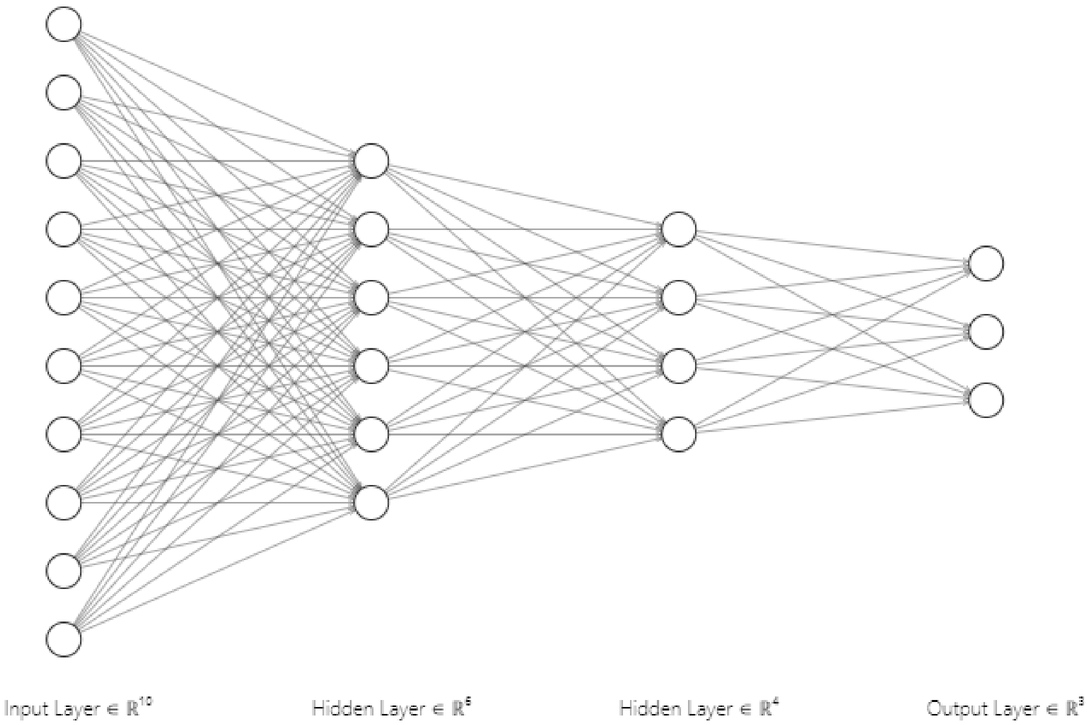


Figure 3.8. Layers of an ANN

By calculating the output values (z_j) from the related weight (w_i) and bias (b) values in the forward propagation (from left to right), cost function c_j is determined and then the weight functions and the bias values are updated in such a way that the difference (error) becomes minimum in the backward propagation process (from right to left). Such ANNs are also in the form of a Feed-forward network (FFN) since the information flows in one direction (from left to right) in comparison to the bi-directional networks such as recurrent neural networks (RNN). By using these updated values, the same procedure repeats and from left to the right propagation, the output values become closer to the cost value but the same reverse propagation is applied to update the values once again to make it closer and these forward and backward calculation processes repeat one after another. When the output values reach the desired closeness to the desired cost values, the education becomes completed. Here the cost function plays an important role to give a measure of pleasure to enable updating the parameters. Consequently, the ANN education can be summarised as finding the web parameters minimizing the cost values. In this aspect, the back propagation algorithms used in the ANN education are the algorithms to minimize the cost function. Web parameters to minimize the cost function are determined by using the *gradient descent* or advanced optimisation algorithms (or a mixture in the coexistence of them) to update the web parameters for the next propagation. Since the derivative of the sigmoid function given in Table 3.2 becomes:

$$\frac{d\sigma(z)}{dz} = \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right) = \frac{e^{-z}}{(1+e^{-z})^2} = \sigma(z)[1-\sigma(z)] \quad (3.10)$$

which also involves again the sigmoid function itself, it presents important easiness in the calculations. ML models including regression, hence the ANNs, according to the learning types (supervised, unsupervised, etc.) discussed above and schematic sketches are given in Figure 3.3 above where we see that ANN (being in regression modelling) is of the type Supervised Learning.

Deep learning (DL) residing in the innermost cluster of the artificial intelligence diagram in Figure 3.1 is among today's pimary research and it involves the following advanced neural networks: *Deep Neural Network – DNN*, *Convolutional Neural Network – CNN*, and *Recurrent Neural Network – RNN* algorithms. Moreover, very important results are obtained by using the *Long-Short-Term-Memory-LTSM* algorithms where four interaction layers are used as a developed model. A schematic diagram of the multi-loops in the RNN is given in Figure 3.9, the web diagram of a single layered RNN is given in Figure 3.10, and an LSTM neural web with four layers in a single module is given in Figure 3.11. Such developed methods regarding the deep learning methods seem giving important results in the value prediction of the Bitcoin cryptocurrency as well as other daily hot topics in literature, i.e., in [26,29,32].

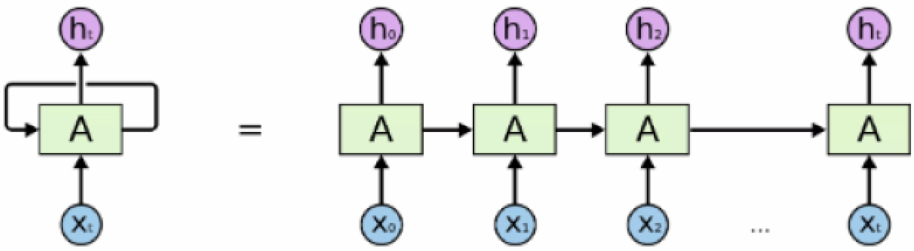


Figure 3.9. A schematic diagram of the multiloops in the RNN [69,70].

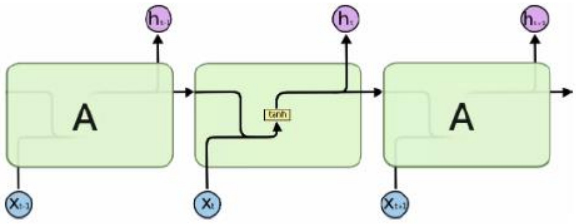


Figure 3.10. The web diagram of a single layered RNN [69,70].

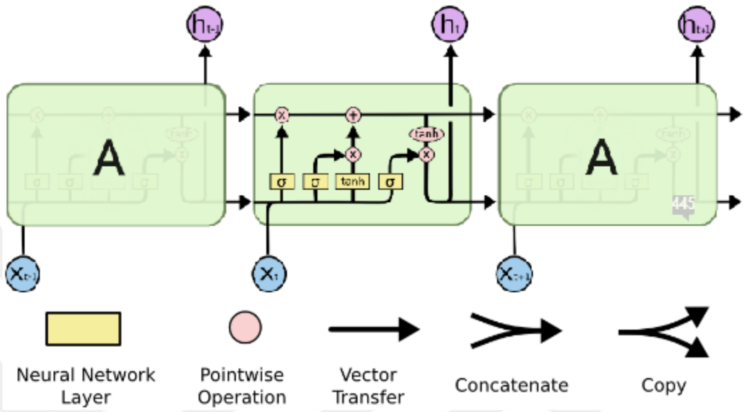


Figure 3.11. An LSTM neural web with four layers in a single module [69,70].

4. BTC PREDICTON MODELS OF THIS STUDY AND PREPARATION OF THE DATA SET

In this work, we study two BTC prediction models based on the candle-stick model of the financial BTC data similar to [64,65]. A typical BTC candle stick diagram of BTC is given in Figure 2.11 and explanation of the candle-stick charts is given in Figure 2.12. In this section, we study our prediction models and preparation of dataset parsed from the Yahoo-finance web pages for our study-models by using the Python programming language.

First of all, n numbers of parsed and trimmed candlestick data set we use in our models have the form as follows:

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D_1} \\ \boldsymbol{D_2} \\ \boldsymbol{D_3} \\ \vdots \\ \boldsymbol{D_i} \\ \vdots \\ \boldsymbol{D_n} \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ \vdots & \vdots & \vdots & \vdots \\ d_{i1} & d_{i2} & d_{i3} & d_{14} \\ \vdots & \vdots & \vdots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & d_{n4} \end{bmatrix} \tag{4.1a}$$

where the i-th row involves the Open (O), High (H), Low (L), and Closed (C) values of the related candle stick in our time-series data frame whose index is *Datetime* or *Date* (depending on the time interval between any two subsequent candle sticks), namely,

$$\boldsymbol{D_i} = [d_{i1} = O \ \ d_{i2} = H \ \ d_{i3} = L \ \ d_{i4} = C] \tag{4.1b}$$

where i is the candle index (row number in our data frame) and the second index in d_{ij} where $1 \leq j \leq 4$ is the column index representing O-H-L-C values in the given order as the definition in (4.1b) tells. Other values such as “Adjusted Volume” in our parsed dataframe are trimmed and we start with the dataframe described by the equations 4.1a-b above.

Our interest here is to establish a relationship in our dataframe and study a prediction model to provide help to the financiers (or investors) wondering the *Close price value* (or simply C-value, or mathematically, d_{i4} value) of BTC in the future by using the fundamental AI (or more specifically, ML) techniques. Consequently, we work on labelled data and our study basically involves the supervised learning with a preferred regression model of ML. We study MLR and ANN regression models in this work.

4.1. Our First Prediction Model

In our first model, the dependent variable ("C"="Close" value) in a row of our dataset is assumed to be depending only on the other three data of the same candle, in other words, in the same row (with the same candle index, i) so that our independent variables under this assumption can be expressed as follows:

$$\vec{X}_{\boldsymbol{j}} = \left[x_{1j} \leftarrow d_{1j} \quad x_{2j} \leftarrow d_{2j} \ldots x_{ij} \leftarrow d_{ij} \ldots x_{nj} \leftarrow d_{nj}\right]^T,$$

$$i = 1,2, \ldots, n; j = 1,2,3 \qquad (4.2a)$$

where

$$\boldsymbol{X} = \left[\vec{X}_1 \quad \vec{X}_2 \quad \vec{X}_3\right] = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \\ \vdots & \vdots & \vdots \\ d_{i1} & d_{i2} & d_{i3} \\ \vdots & \vdots & \vdots \\ d_{n1} & d_{n2} & d_{n3} \end{bmatrix} \qquad (4.2b)$$

Then, our dependent variable under the assumption becomes as follows:

$$C \leftarrow d_{i4} \Longrightarrow \vec{Y} = [y_1\ y_2\ y_3 \ldots y_i \ldots y_n]^T \qquad (4.2c)$$

Consequently, under this assumption our first prediction model is a *3-dimensional multiple regression:*

$$\widehat{\boldsymbol{Y}} =: \vec{Y} \approx \beta_0 + \sum_{j=1}^{3} \beta_j \vec{X}_j + \epsilon \qquad (4.2.d)$$

4.2. Our Second Prediction Model

In our second prediction model, the dependent variable ("C"="Close" values) for the i-th candle is assumed to be depending on the values of the three preceding candles (with the candle indices, $i-3, i-2, i-1$) so that our independent variables under this assumption can be expressed as follows:

$$\vec{X}_{\boldsymbol{j}} = \left[x_{1j} \leftarrow \{d_{1j,} d_{2j}, d_{3j}\} \quad x_{2j} \leftarrow \{d_{4j}, d_{5j}, d_{6j}\} \ldots x_{i-4,j} \leftarrow \{d_{n-2,j}, d_{n-1,j}, d_{n,j}\}\right]^T,$$

$$i = 1,2, \ldots, n; j = 1,2,3 \qquad (4.3a)$$

$$\boldsymbol{X} = [\vec{X}_1 \quad \vec{X}_2 \quad \dots \quad \vec{X}_{12}] = \begin{bmatrix} \boldsymbol{D_1} & \boldsymbol{D_2} & \boldsymbol{D_3} \\ \boldsymbol{D_4} & \boldsymbol{D_5} & \boldsymbol{D_6} \\ \boldsymbol{D_7} & \boldsymbol{D_8} & \boldsymbol{D_9} \\ \vdots & \vdots & \vdots \\ \boldsymbol{D_i} & \boldsymbol{D_{i+1}} & \boldsymbol{D_{i+2}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{D_{n-3}} & \boldsymbol{D_{n-2}} & \boldsymbol{D_{n-1}} \end{bmatrix} \tag{4.3b}$$

Then, our dependent variable under the assumption becomes

$$y_i \leftarrow d_{i+3,4} \Longrightarrow \vec{Y} = [y_1 \; y_2 \; y_3 \dots y_i \dots y_{n-3}]^T$$

$$= [d_{44} \; d_{54} \; d_{64} \dots d_{i+3,4} \dots d_{n4}]^T \tag{4.3c}$$

Consequently, under this assumption we have *12-dimensional multiple regression:*

$$\widehat{\boldsymbol{Y}} =: \vec{Y} \approx \beta_0 + \sum_{j=1}^{12} \beta_j \vec{X}_j + \epsilon \tag{4.3d}$$

Here, since each $\boldsymbol{D_i}$ has a dimension of (1×4), $\boldsymbol{X}$ has a dimension of $(n-3) \times 12$.

4.3. Preparation of the Dataset

In order to measure the success of our MLR or ANN-based regression models, we need to use short, medium, and large-scale financial BTC data to make a comparison. In order to acquire such data, we parsed Yahoo-finance datasets to study five different data set by using *Pandas library* in *Python programming language* as follows:

i) Dataset#1 (1-day BTC data):

Between dates: 2023-12-01 and 2023-12-02, in 15-minute intervals.

ii) Dataset#2 (1-week BTC data):

Between dates: 2023-12-04 and 2023-12-11, in 15-minute intervals.

iii) Dataset#3 (1-month BTC data):

Between dates: 2023-12-01 and 2024-01-01, in 15-minute intervals.

iv) Dataset#4 (1-year BTC data):

Between dates: 2023-01-01 and 2024-01-01, in 1-hour intervals.

v) Dataset#5 (Whole BTC data available in Yahoo-finance):

Between dates: 2014-09-17 and 2024-01-01, in -day intervals.

We here note that we enter the Python codes to retrieve the data starting from 2009-01-01 which involves the date of the first BTC transaction (in 2009-09-17) but the data retrieved starts from date 2014-09-17. Python code we used to parse, save, and visualize Dataset#1 is given in Figure 4.1 for illustration. Parsing others are similar. Candlestick diagrams of our dataframes are given in Figure 4.2. The last candlestick diagram in Figure 4.2 also shows that substantial volatility starts around the end of year 2017.

```
import pandas as pd
import yfinance as yf
def BTC_download(start, end, period, interval):
    return yf.download(tickers='BTC-USD', start=start,end=end, period=period,interval=interval)
data1=BTC_download('2023-12-1', '2023-12-2','1d','15m') #Parsing
data1.reset_index(level=0, inplace=True)#Seting Datetime as a column
data1.to_csv('data1.csv',index=False)#Save the dataframe in csv
import os;os.getcwd()#Get current directory (optional)
dataread = pd.read_csv('data1.csv') #Read the whole file here
import plotly.io as pio
pio.renderers.default = 'browser'
pio.renderers
import plotly.offline as pyo
import plotly.graph_objects as go
from datetime import datetime
pyo.offline.init_notebook_mode()
fig1 = go.Figure(data=[go.Candlestick(x=data1['Datetime'],
                open=data1['Open'],
                high=data1['High'],
                low=data1['Low'],
                close=data1['Adj Close'])])
fig1.update_layout(title_text='Dataset#1: 1-day BTC data (12-01-2023) in 15m intervals', title_x=0.5)
pyo.plot(fig1)#Visualise data
```

Figure 4.1. Python codes we used to parse, save, and visualize for Dataset#1

Figure 4.2. Candlestick graphs of our datasets *(continuing)*

Figure 4.2. Candlestick graphs of our datasets *(continuing)*

Figure 4.2. Candlestick graphs of our datasets

5. RESULTS OF THE PREDICTION MODELS BY THE MLR

The five datasets prepared for our two-prediction models are first split randomly into 80-to-20 train-test data (in percentage) by using the *Sklearn library*. Results of coefficients, intercepts, and test scores (for both train and test data) in our MLR application for model#1 and model#2 are given in Table 5.1 and Table 5.2, respectively.

Table 5.1. Results of MLR for model#1

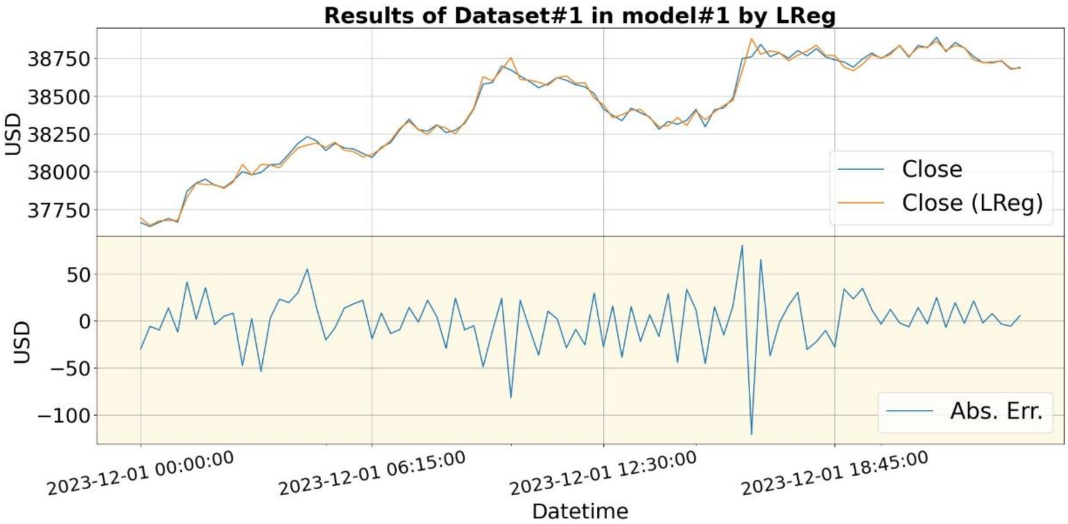
	Train and test scores	**MLR Coefficients** (β_i $(1 \leq i \leq 3)$)	**MLR intercept** (β_0)
Data#1	0.994232 0.980316	-0.61627 0.757475 0.851538	281.490264
Data#2	0.998882 0.997566	-0.712889, 0.90393 0.806328	111.688285
Data#3	0.999351 0.999452	-0.705252 0.885687 0.819123	16.406285
Data#4	0.999927 0.999926	-0.550544 0.777678 0.772873	0.802857
Data#5	0.999590 0.999649	-0.511248 0.853502 0.653808	7.894775

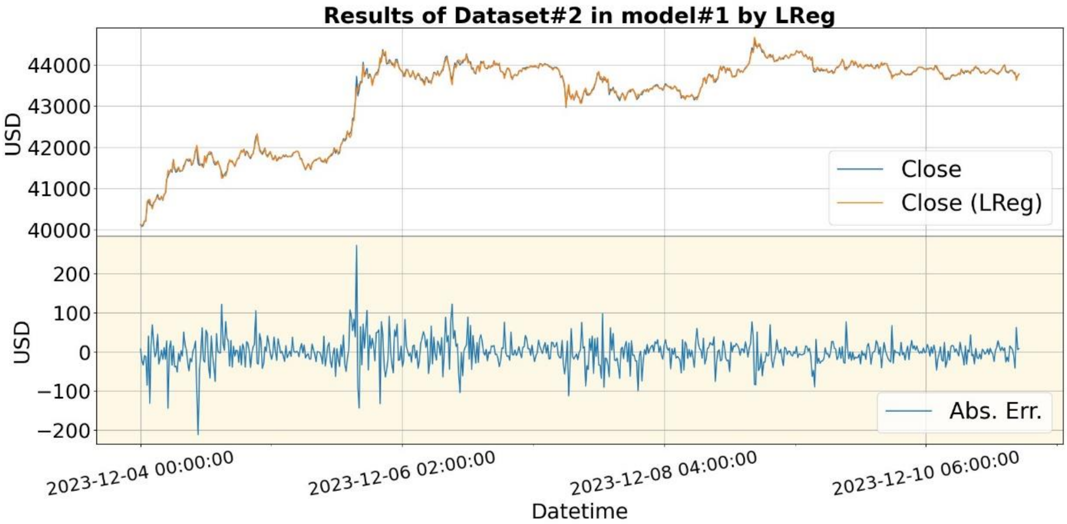
Table 5.2. Results of MLR for model#2

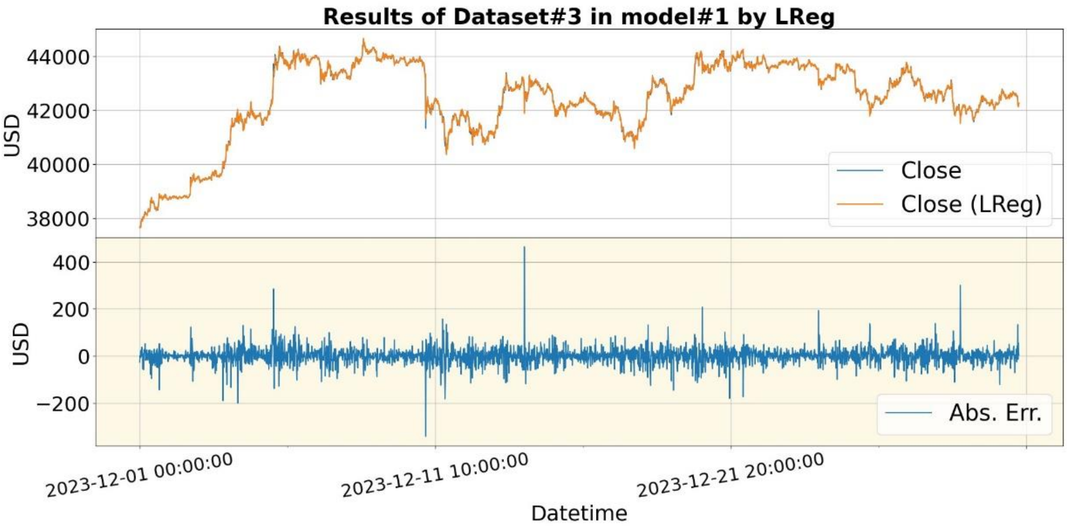
	Train and test scores	**MLR Coefficients** (β_i $(1 \leq i \leq 12)$)	**MLR intercept** (β_0)
Data#1	0.966409 0.943293	-0.214874 -0.188937 0.380744 0.250318 -0.549029 0.022449 0.079868 1.488428 -1.466346 0.072767 0.333304 0.745847	1788.519483
Data#2	0.990888	-0.218178	643.678380

	0.989331	0.176527 0.167576 0.107731 0.034794 -0.328562 -0.318441 0.64757 -0.132534 -0.12843 -0.13848 1.115655	
Data#3	0.995184 0.995857	-0.025989 -0.009715 -0.014484 -0.062381 0.206184 -0.111345 -0.119399 0.404523 -0.227782 -0.009755 -0.047833 1.013887	172.769168
Data#4	0.999527 0.999495	0.008322 -0.009652 -0.036199 -0.050362 0.058507 0.053952 -0.041032 -0.224249 0.311728 -0.033642 -0.039971 1.001637	22.431565
Data#5	0.997674 0.997665	-0.082452 0.042404 0.069167 -0.23954 0.167231 -0.029504 0.020707 0.373895 -0.182197 -0.076875 -0.184298 1.118138	35.093951

Graphs of our BTC-price prediction within the dataset for our prediction model#1 and model#2 are given in Figure 5.1 and Figure 5.2, respectively.







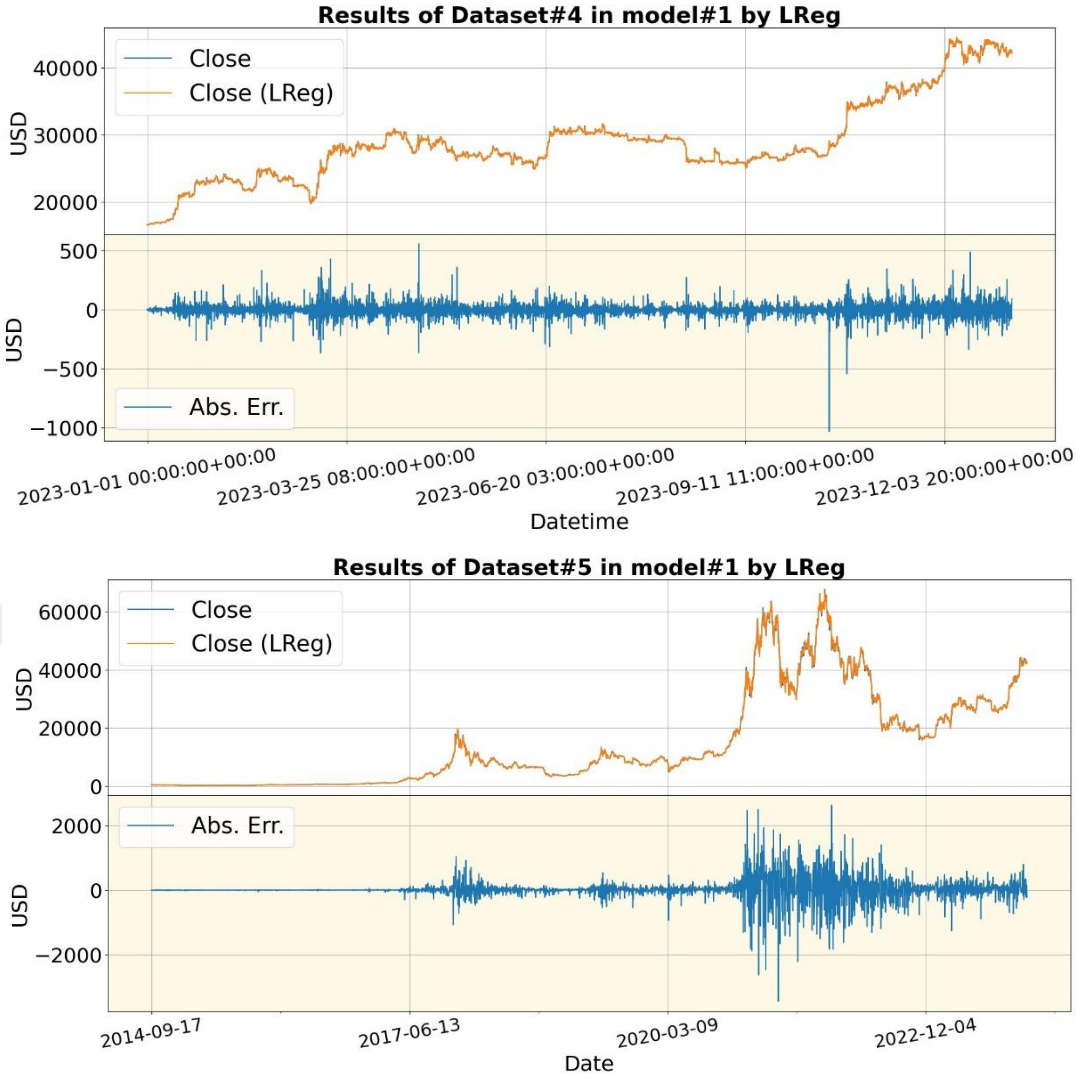
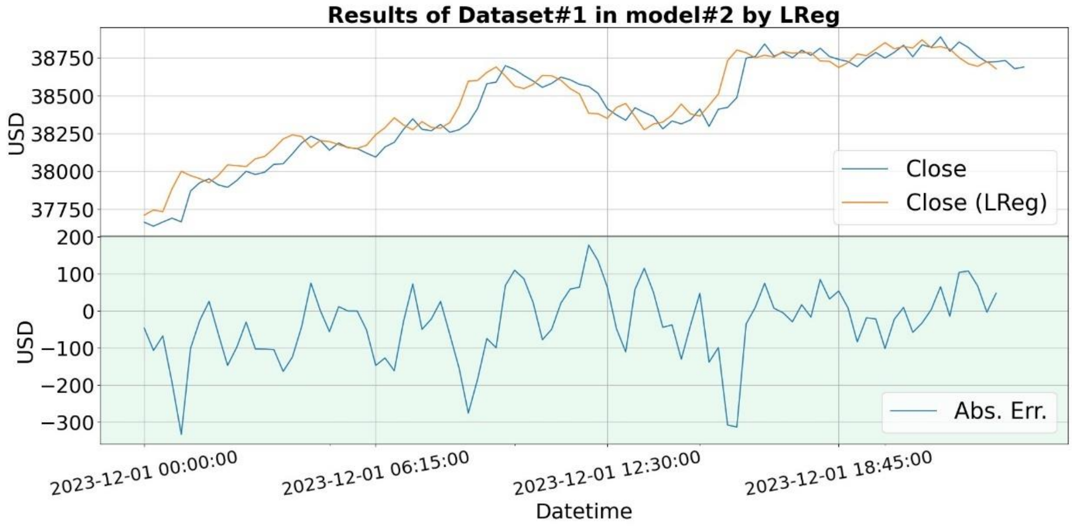
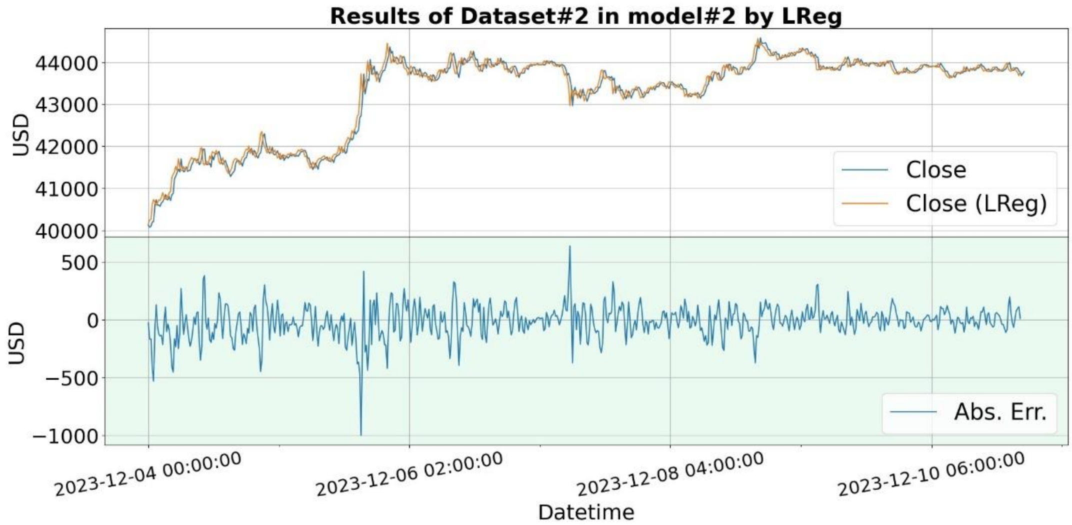
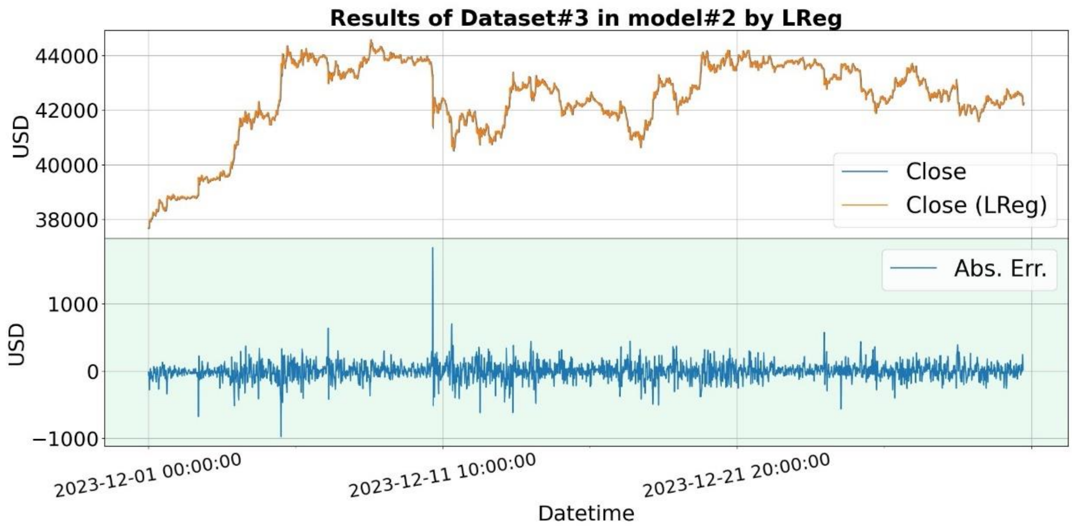


Figure 5.1. LR-Results of our <u>first prediction model</u> (for <u>model#1</u>) within our dataset.


Results of Dataset#1 in model#2 by LReg
USD
38750
38500
38250
38000
37750
Close
Close (LReg)
200
100
0
−100
−200
−300
Abs. Err.
2023-12-01 00:00:00
2023-12-01 06:15:00
2023-12-01 12:30:00
2023-12-01 18:45:00
Datetime


Results of Dataset#2 in model#2 by LReg
USD
44000
43000
42000
41000
40000
Close
Close (LReg)
500
0
−500
−1000
Abs. Err.
2023-12-04 00:00:00
2023-12-06 02:00:00
2023-12-08 04:00:00
2023-12-10 06:00:00
Datetime


Results of Dataset#3 in model#2 by LReg
USD
44000
42000
40000
38000
Close
Close (LReg)
Abs. Err.
1000
0
−1000
2023-12-01 00:00:00
2023-12-11 10:00:00
2023-12-21 20:00:00
Datetime

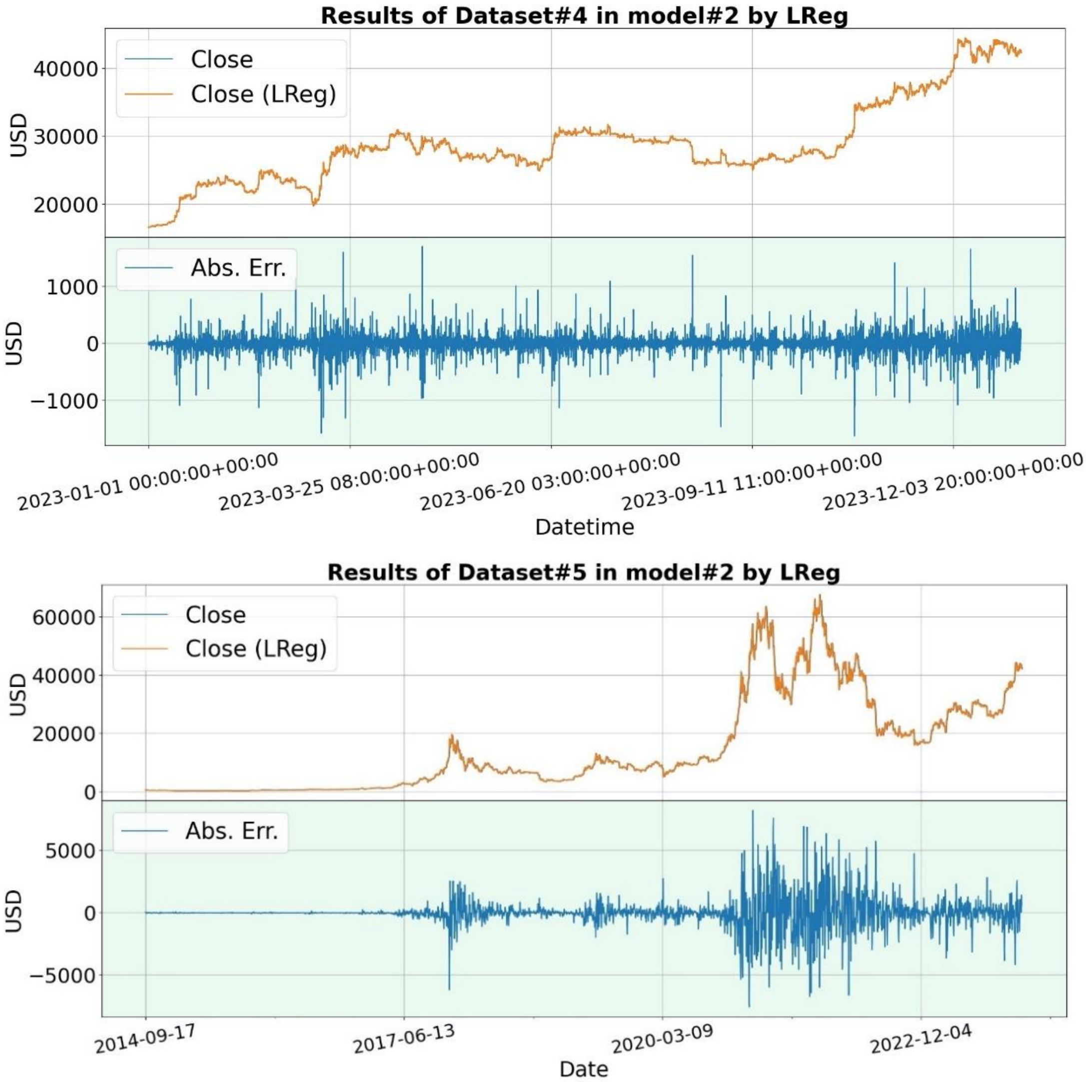
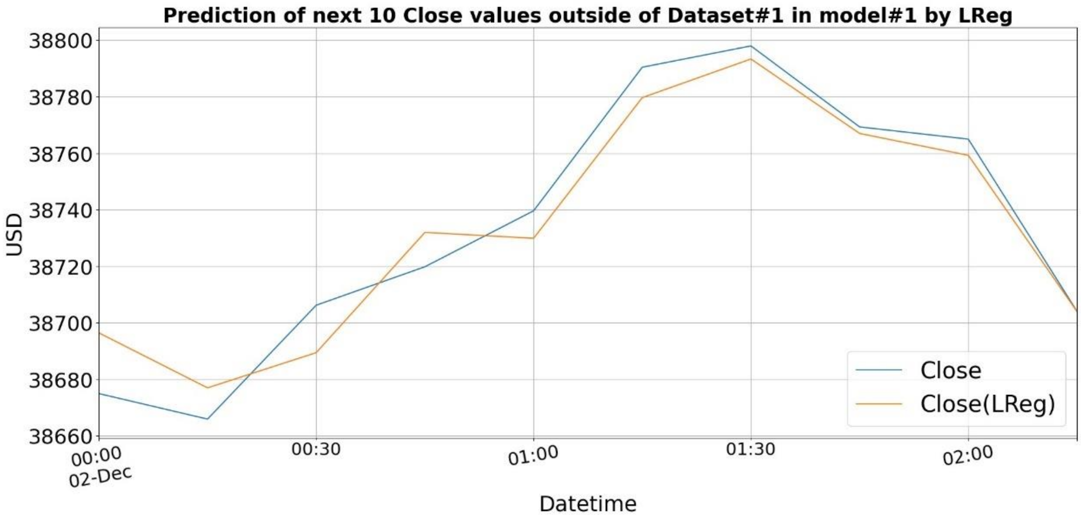
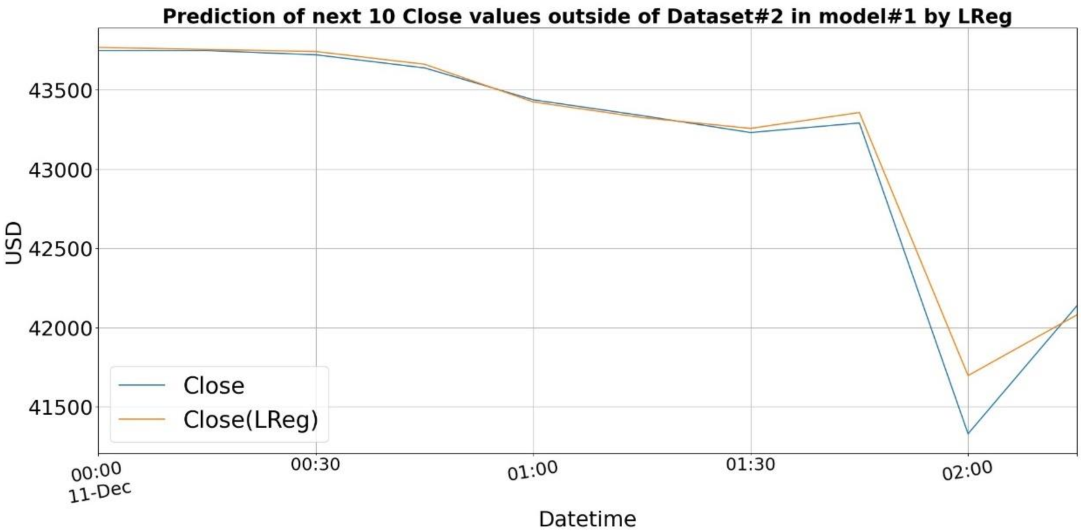
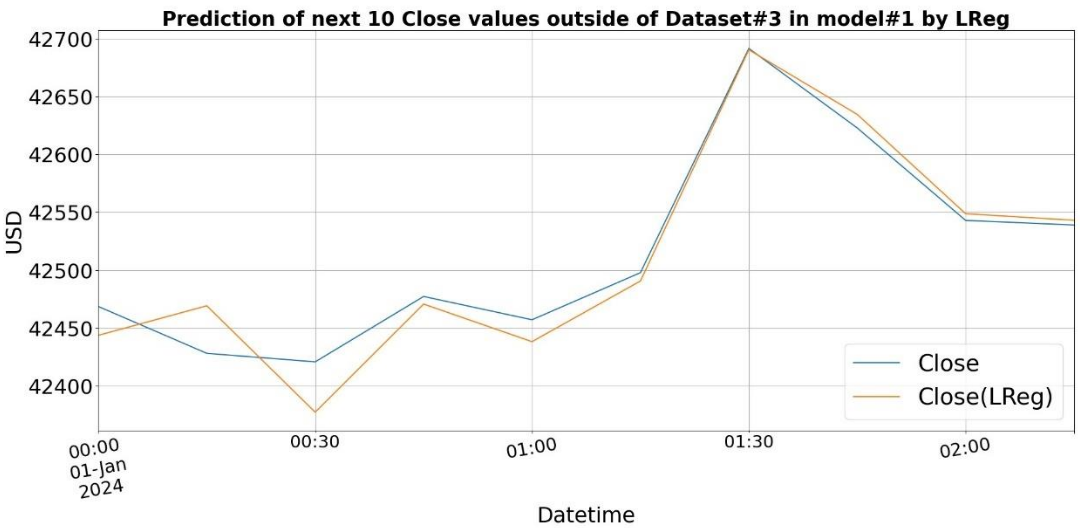


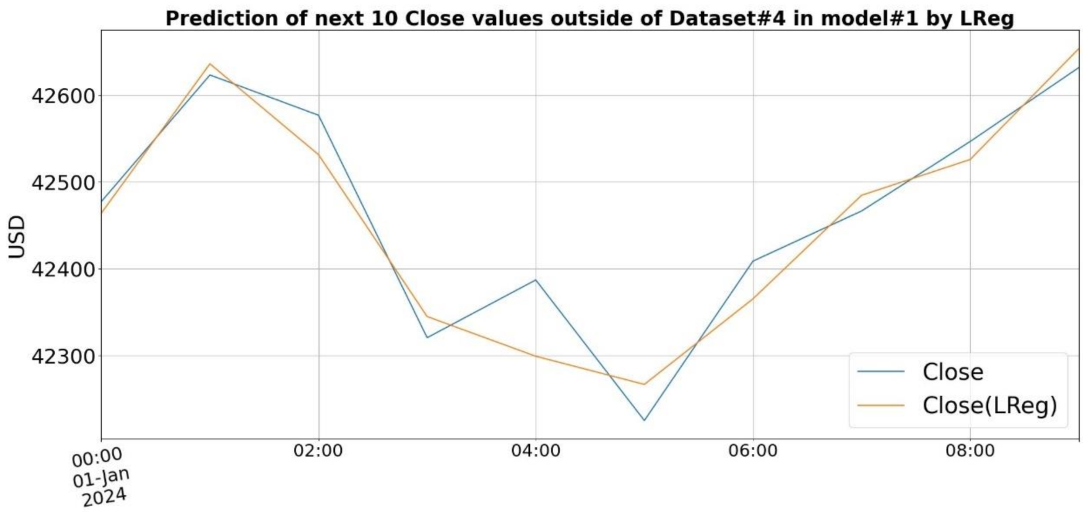
Figure 5.2. LR-Results of our second prediction model (for model#2) within our dataset.

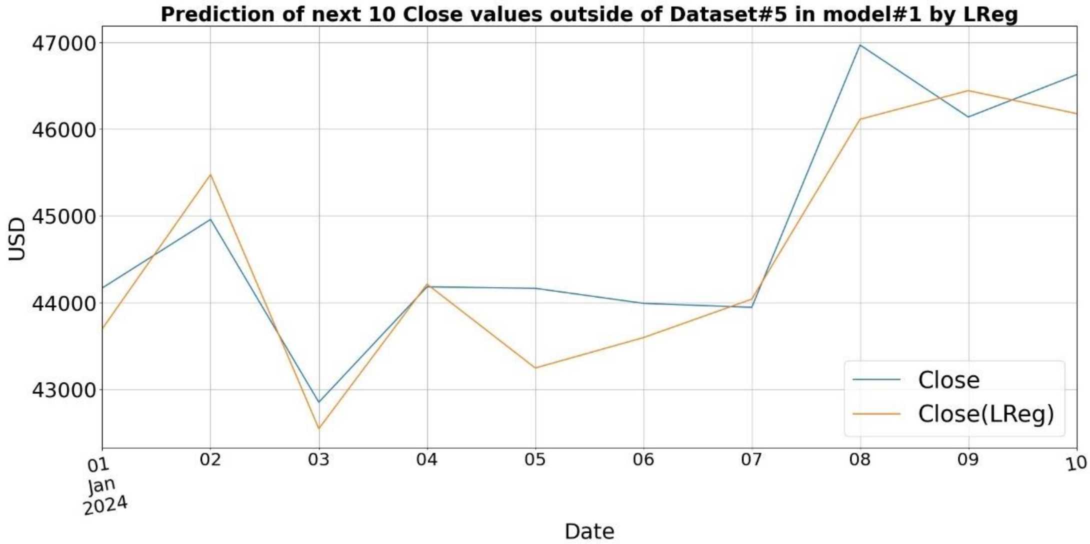
Graphs of our future BTC-price prediction for the next 10 data (beyond the last value of our dataset) for model#1 and model#2 are given in Figure 5.3 and Figure 5.4, respectively.


Prediction of next 10 Close values outside of Dataset#1 in model#1 by LReg
USD
38800
38780
38760
38740
38720
38700
38680
38660
00:00
02-Dec
00:30
01:00
01:30
02:00
Datetime
Close
Close(LReg)

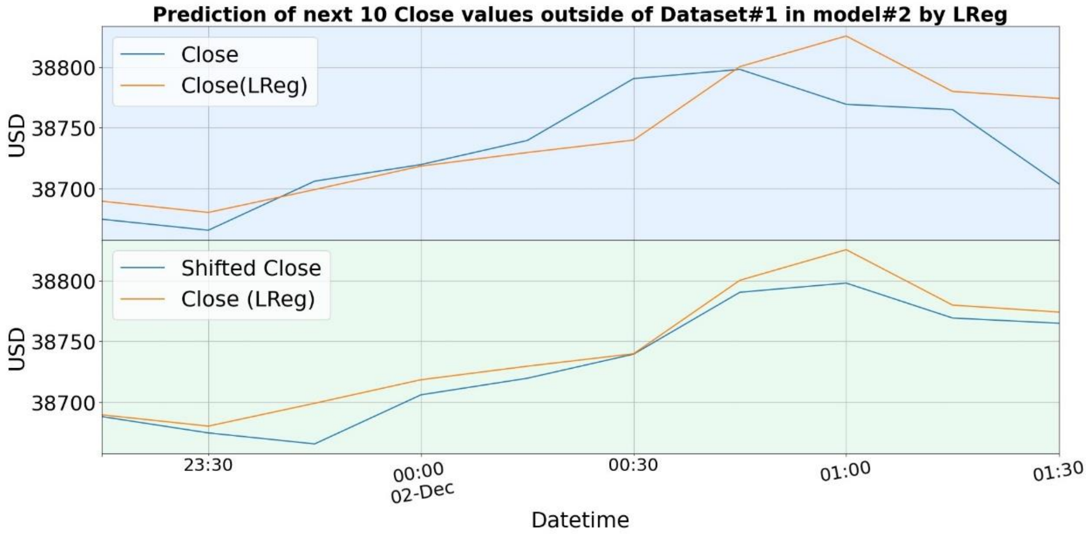

Prediction of next 10 Close values outside of Dataset#2 in model#1 by LReg
USD
43500
43000
42500
42000
41500
00:00
11-Dec
00:30
01:00
01:30
02:00
Datetime
Close
Close(LReg)

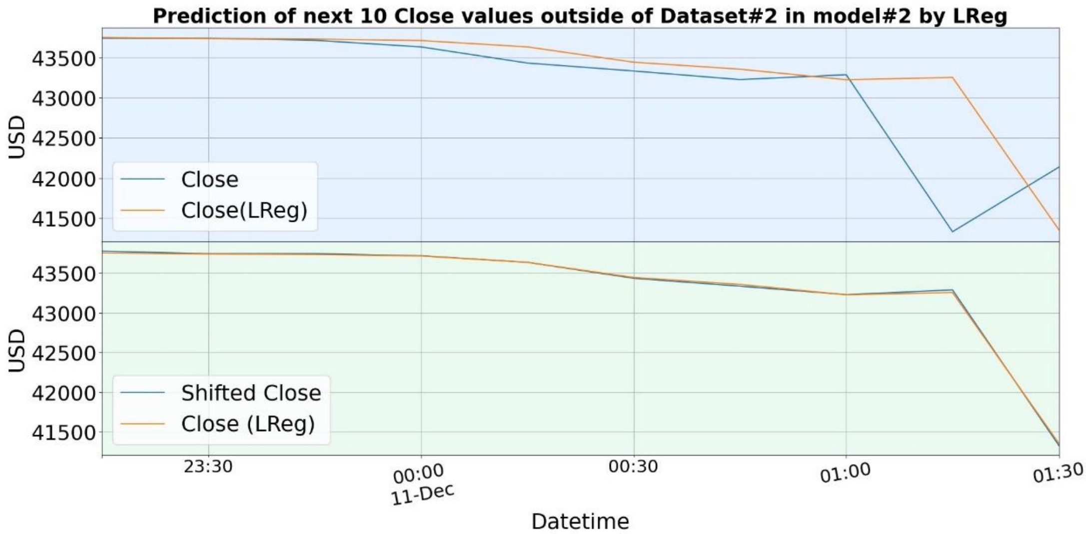

Prediction of next 10 Close values outside of Dataset#3 in model#1 by LReg
USD
42700
42650
42600
42550
42500
42450
42400
00:00
01-Jan
2024
00:30
01:00
01:30
02:00
Datetime
Close
Close(LReg)

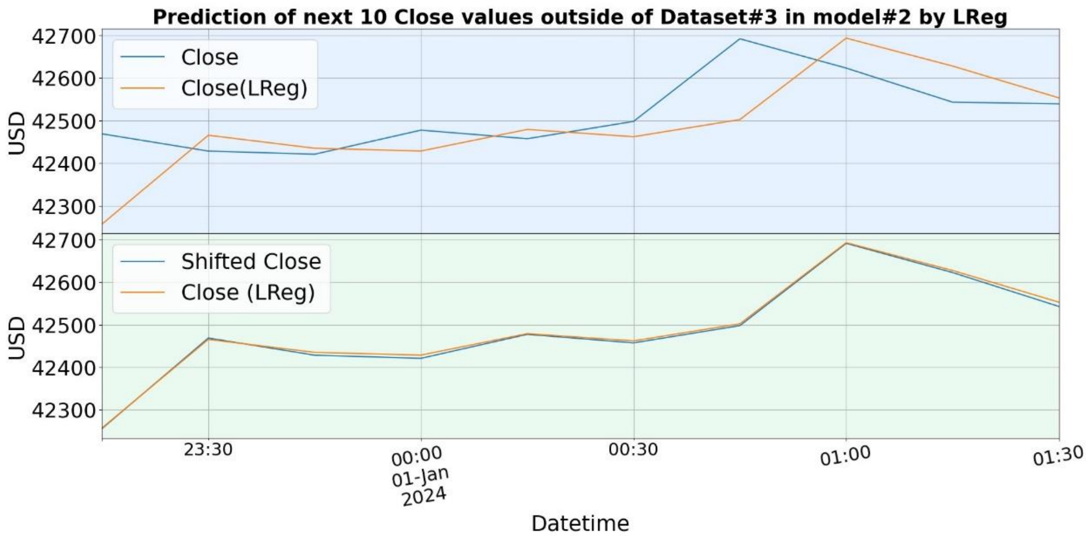


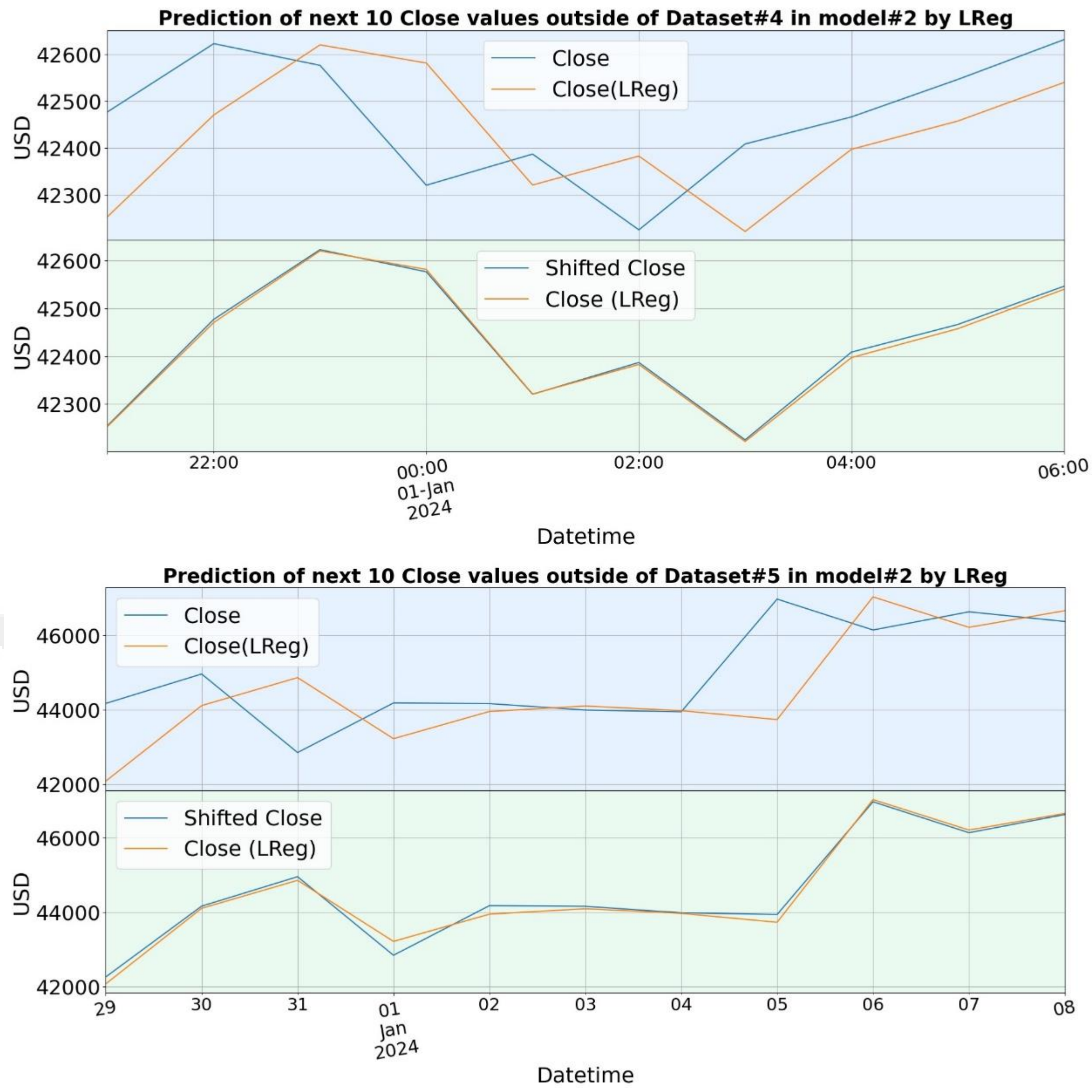


Figures 5.3. LR-Results of our first prediction model (for model#1) in predicting future values (for the prediction of the next 10 data beyond the last data in our dataset).


Prediction of next 10 Close values outside of Dataset#1 in model#2 by LReg
Close
Close(LReg)
Shifted Close
Close (LReg)
USD
38700
38750
38800
23:30
00:00
02-Dec
00:30
01:00
01:30
Datetime

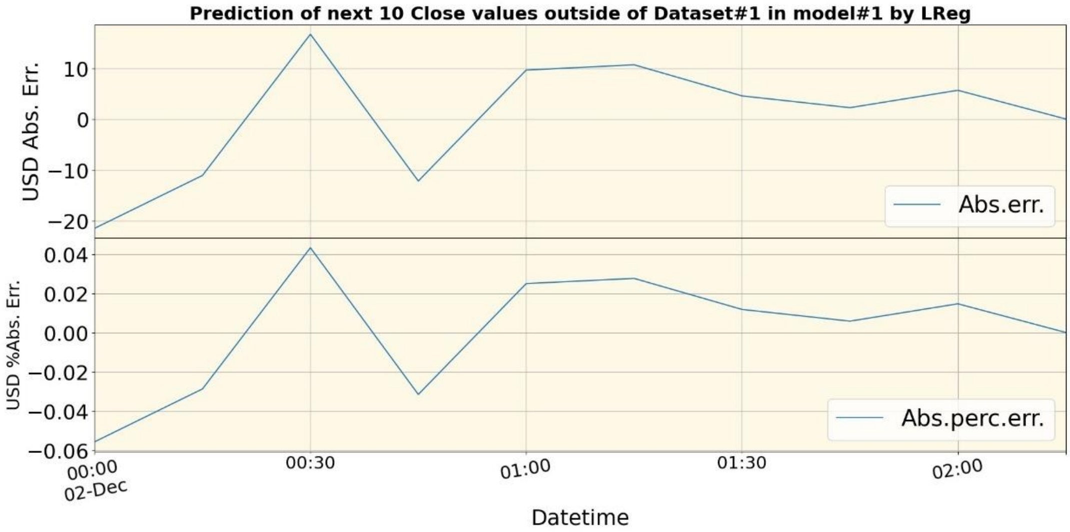

Prediction of next 10 Close values outside of Dataset#2 in model#2 by LReg
Close
Close(LReg)
Shifted Close
Close (LReg)
USD
41500
42000
42500
43000
43500
23:30
00:00
11-Dec
00:30
01:00
01:30
Datetime

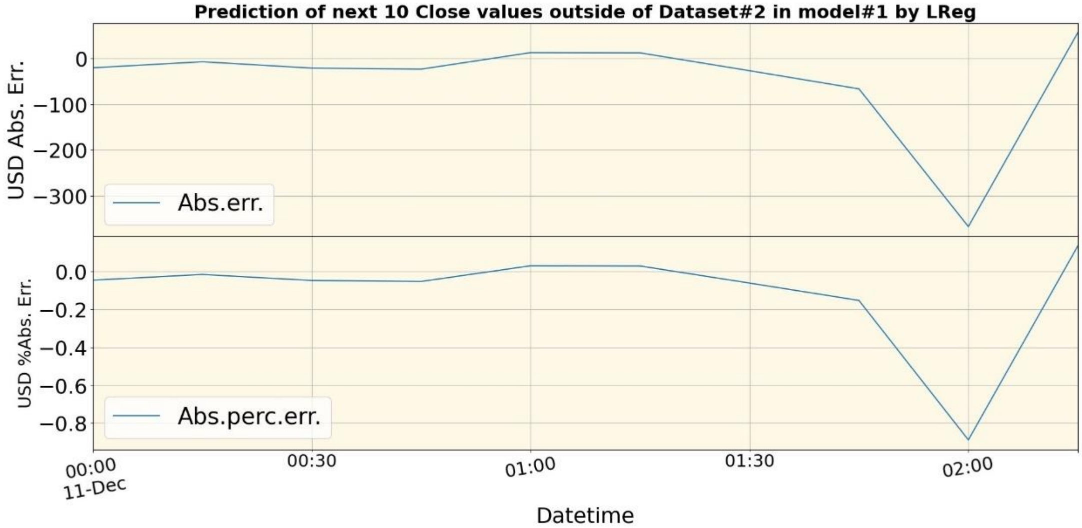

Prediction of next 10 Close values outside of Dataset#3 in model#2 by LReg
Close
Close(LReg)
Shifted Close
Close (LReg)
USD
42300
42400
42500
42600
42700
23:30
00:00
01-Jan
2024
00:30
01:00
01:30
Datetime

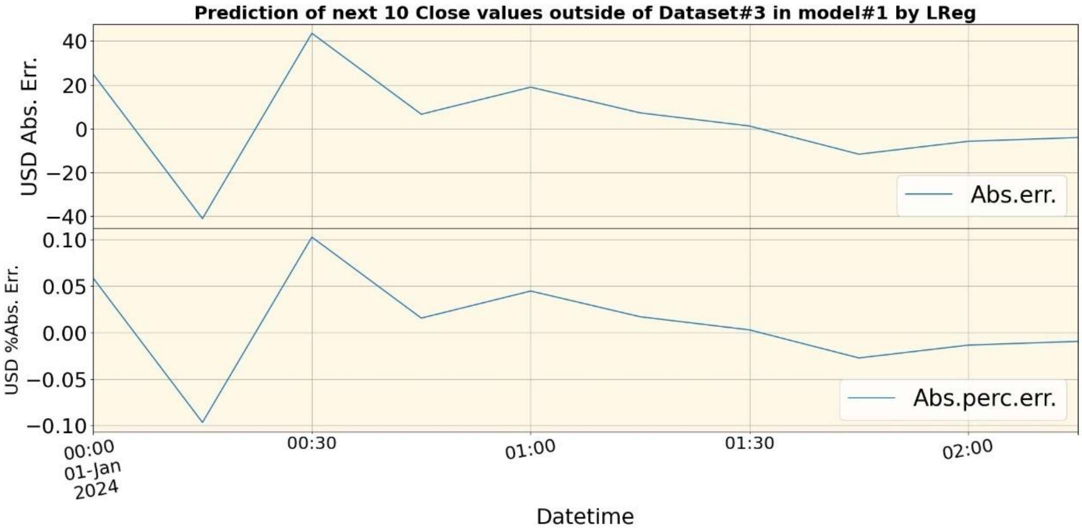


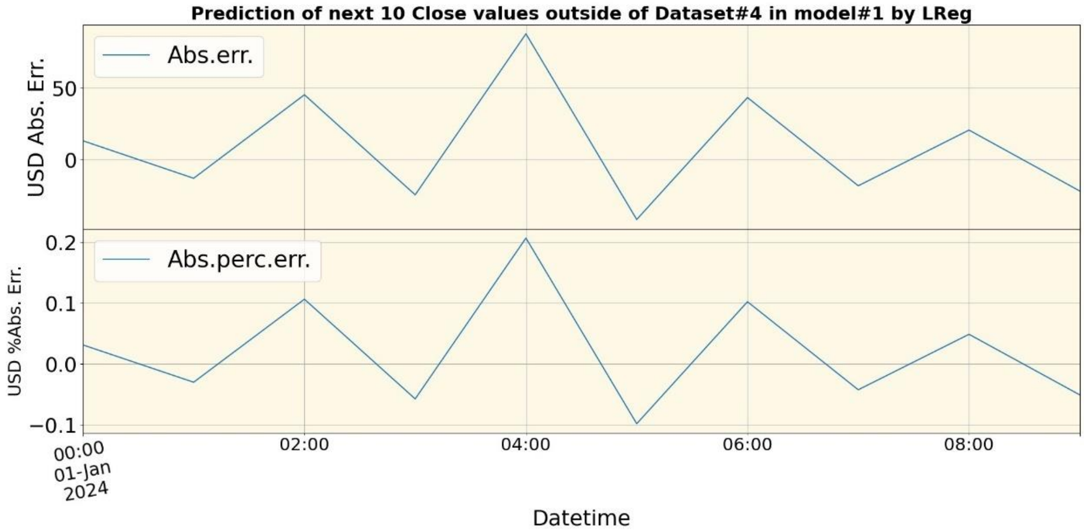
Figures 5.4. LR-Results of our <u>second prediction model</u> (for <u>model#2</u>) in predicting future values (for the prediction of the next 10 data beyond the last data in our dataset).

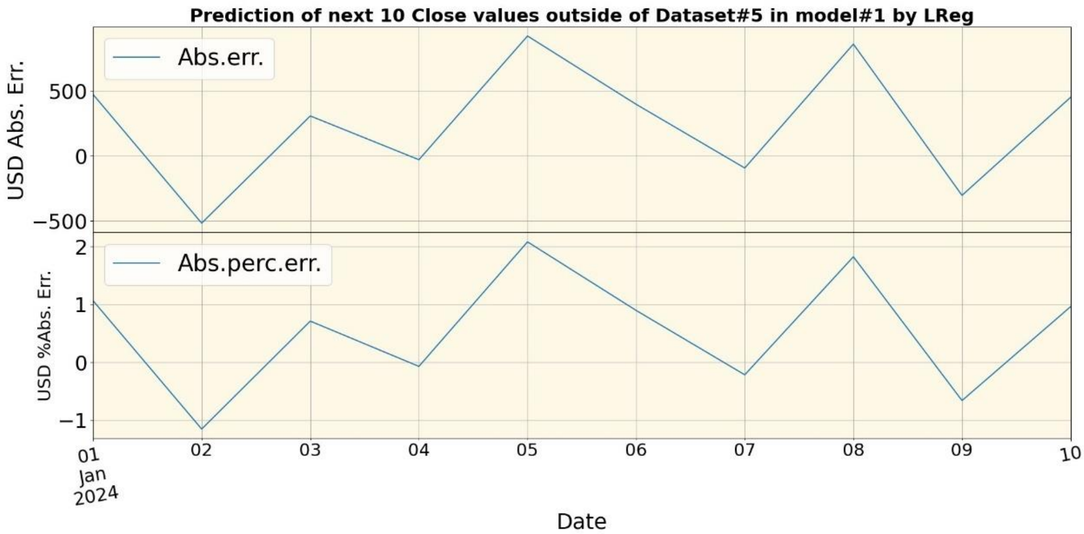
Graphs of absolute errors and absolute percentage errors in our BTC-price prediction for the next 10 data (beyond the last value of our dataset) for model#1 and model#2 are given in Figure 5.5 and Figure 5.6, respectively.


Prediction of next 10 Close values outside of Dataset#1 in model#1 by LReg
USD Abs. Err.
10
0
−10
−20
Abs.err.
USD %Abs. Err.
0.04
0.02
0.00
−0.02
−0.04
−0.06
Abs.perc.err.
00:00
02-Dec
00:30
01:00
01:30
02:00
Datetime

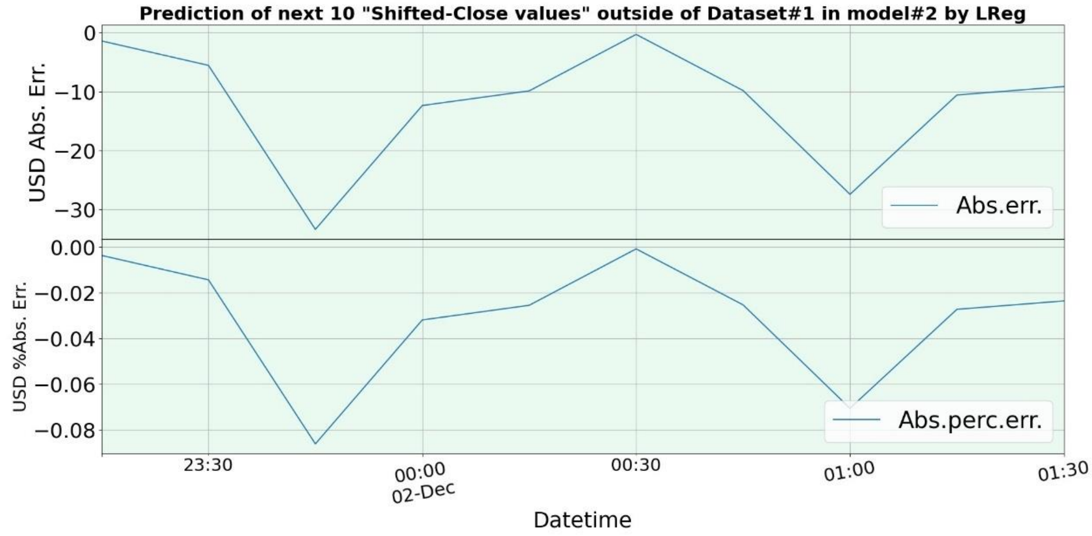

Prediction of next 10 Close values outside of Dataset#2 in model#1 by LReg
USD Abs. Err.
0
−100
−200
−300
Abs.err.
USD %Abs. Err.
0.0
−0.2
−0.4
−0.6
−0.8
Abs.perc.err.
00:00
11-Dec
00:30
01:00
01:30
02:00
Datetime

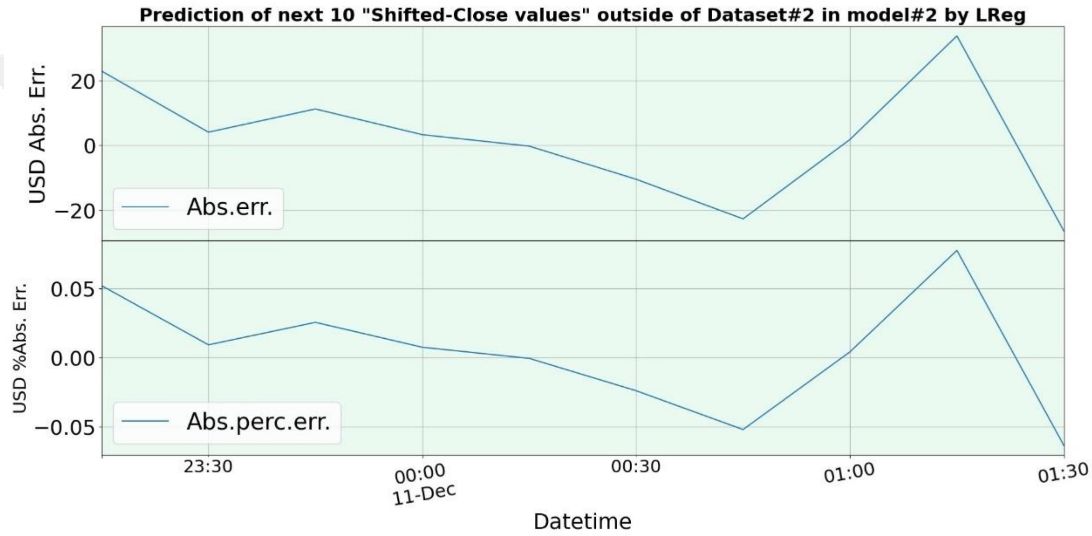

Prediction of next 10 Close values outside of Dataset#3 in model#1 by LReg
USD Abs. Err.
40
20
0
−20
−40
Abs.err.
USD %Abs. Err.
0.10
0.05
0.00
−0.05
−0.10
Abs.perc.err.
00:00
01-Jan
2024
00:30
01:00
01:30
02:00
Datetime

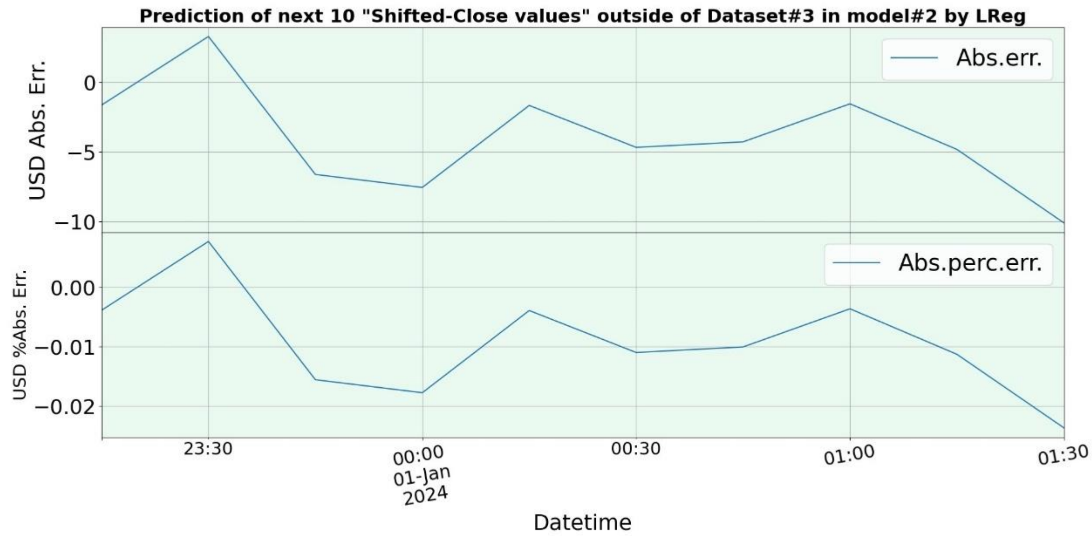


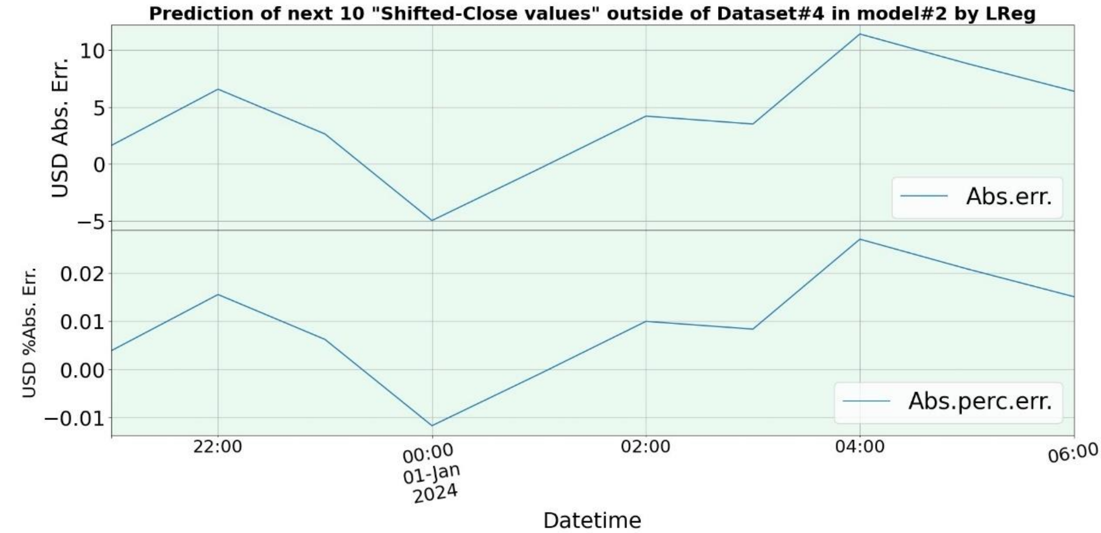


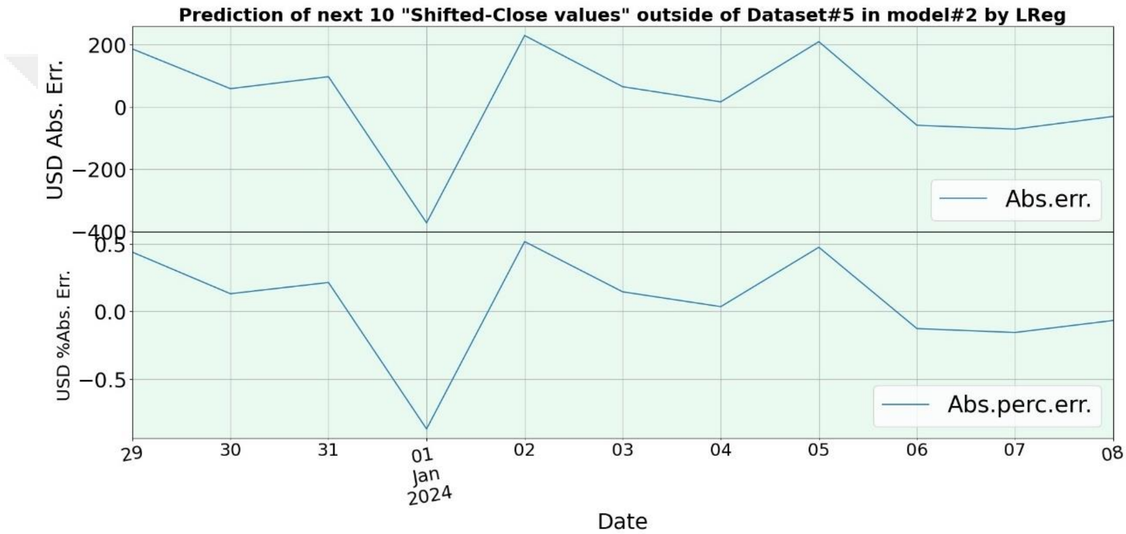
Figures 5.5. LR-Error graphs our <u>first prediction model</u> (for <u>model#1</u>) in predicting future values (for the prediction of the next 10 data beyond the last data in our dataset).


Prediction of next 10 "Shifted-Close values" outside of Dataset#1 in model#2 by LReg
USD Abs. Err.
USD %Abs. Err.
Abs.err.
Abs.perc.err.
Datetime


Prediction of next 10 "Shifted-Close values" outside of Dataset#2 in model#2 by LReg
USD Abs. Err.
USD %Abs. Err.
Abs.err.
Abs.perc.err.
Datetime


Prediction of next 10 "Shifted-Close values" outside of Dataset#3 in model#2 by LReg
USD Abs. Err.
USD %Abs. Err.
Abs.err.
Abs.perc.err.
Datetime





Figures 5.6. LR-Error graphs our second prediction model (for model#2) in predicting future values (for the prediction of the next 10 data beyond the last data in our dataset).

6. RESULTS OF THE PREDICTION MODELS BY THE ANN

Since model#2 enables a prediction of future value of BTC directly, we studied it in an ANN model. Python codes we used for constructing our ANN regression model is as follows:

```
#Prepare our ANN model
# Split dataset
from sklearn.model selection import train_test_split
X_train,X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2,random_state=5)
# create the model
from keras.layers import Dense
from keras.models import Sequential
model = Sequential()
model.add(Dense(4*3, activation="relu", input_dim=4*3))
model.add(Dense(128, activation="relu"))
model.add(Dense(64, activation="relu"))
model.add(Dense(32, activation="relu"))
model.add(Dense(16, activation="relu"))
model.add(Dense(1, activation="linear"))
model.summary()
```

Here, X is the dataframe for inputs and y is the dataframe for the output. We have twelve inputs (O-H-L-C values of three adjacent candles in dataframe X) and one output (C-value of the next candle in dataframe y for the known values and in $\hat{y}$ for the predicted values) in Model#2. Consequently, we defined twelve input values for the input layer. In our studied ANN model, the input layer has 12 outputs (=neurons) then the subsequent layers have 128, 64, 32, and 16 outputs (=neurons) all having the “relu” activation function. Then the output layer has one output (=neuron) to give the C-value of the next candle with the “linear” activation function for the prediction value of the BTC price. We again used random split of 80% train and 20% test in our datasets. Structure of the ANN model we studied here is given in Figure 6.1.

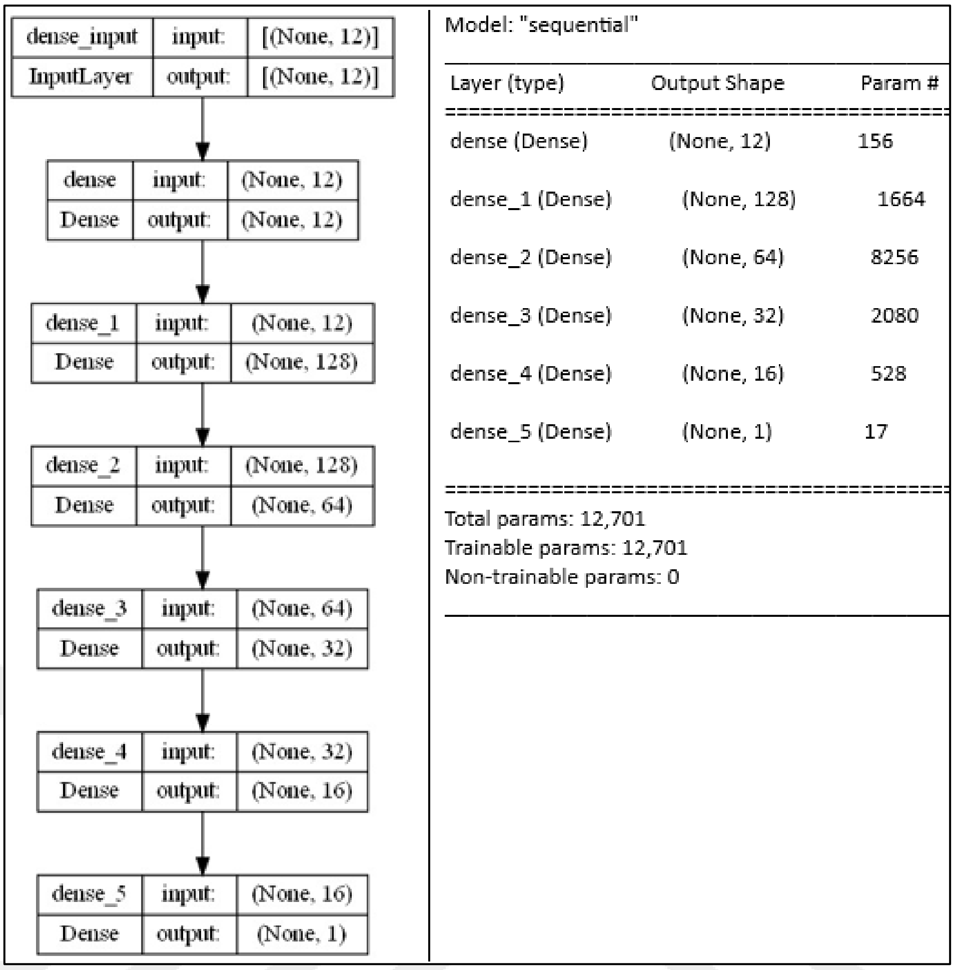


Figure 6.1. Structure of the ANN model used here (Figures are created and taken from our Python Codes)

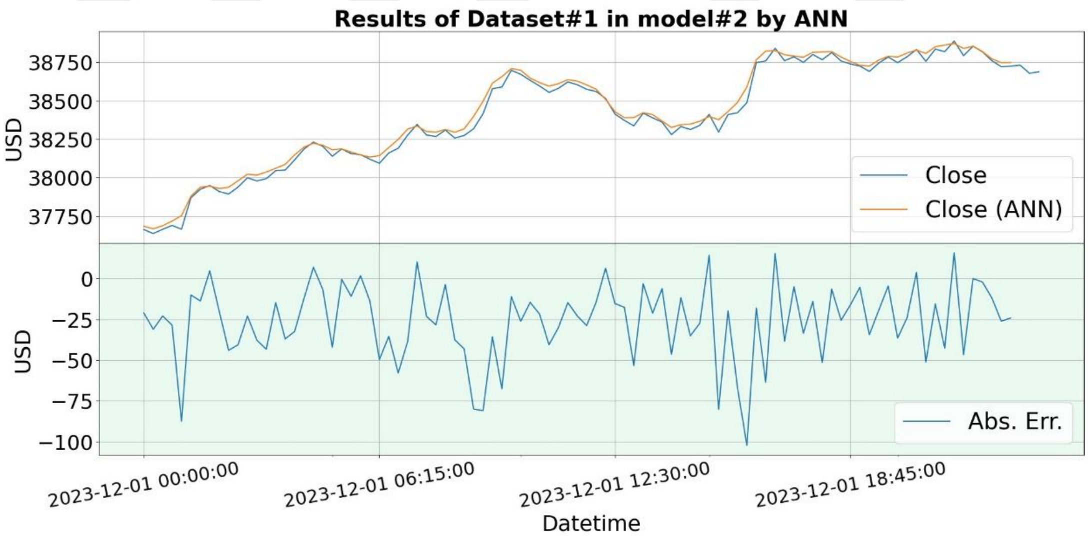
```
# compile the model
model.compile(optimizer="adam",loss="mean_squared_error"
, metrics=["accuracy"])
# train the model
model.fit(X_train,y_train,epochs=100, batch_size=32)
# evaluate the model
model.evaluate(X_test, y_test)
# predict the Close value for the test data
y_pred_test= model.predict(X_test)
y_pred_test=pd.DataFrame(y_pred_test)
# predict the Close value for the whole data
y_pred= model.predict(X)
y_pred=pd.DataFrame(y)
y_pred=pd.DataFrame(y_pred) #We use y_pred=y_hat here
```

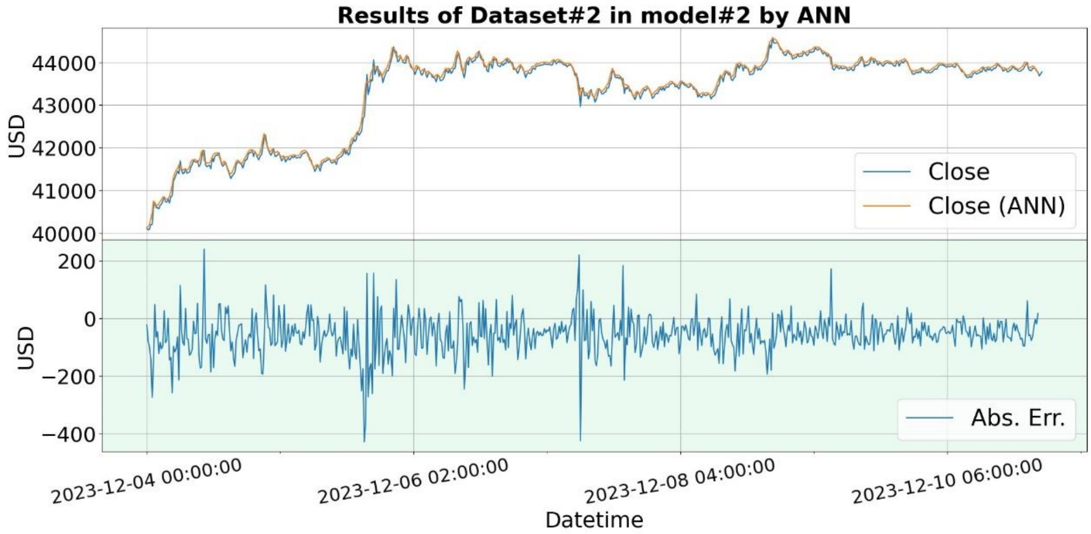
In the ANN model we used, we have epoch and batch-size values of 100 and 32, respectively.

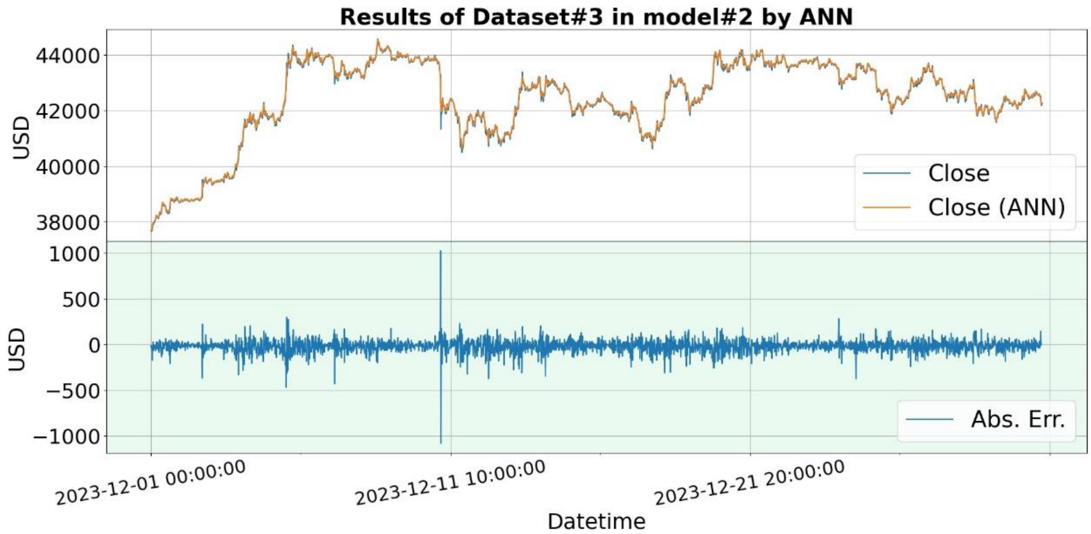
ANN Results of our second prediction model (for model#2) within our datasets and outside of our datasets (for predicting the next 10 BTC values) are given in Figure 6.2 and Figure 6.3, respectively. Absolute error and %Absolute error (Rel. percentage error) graphs of the ANN results of future predictions in model#2 are given in Figure 6.4. Train and test score values associated with these graphs are given in Table 6.1.

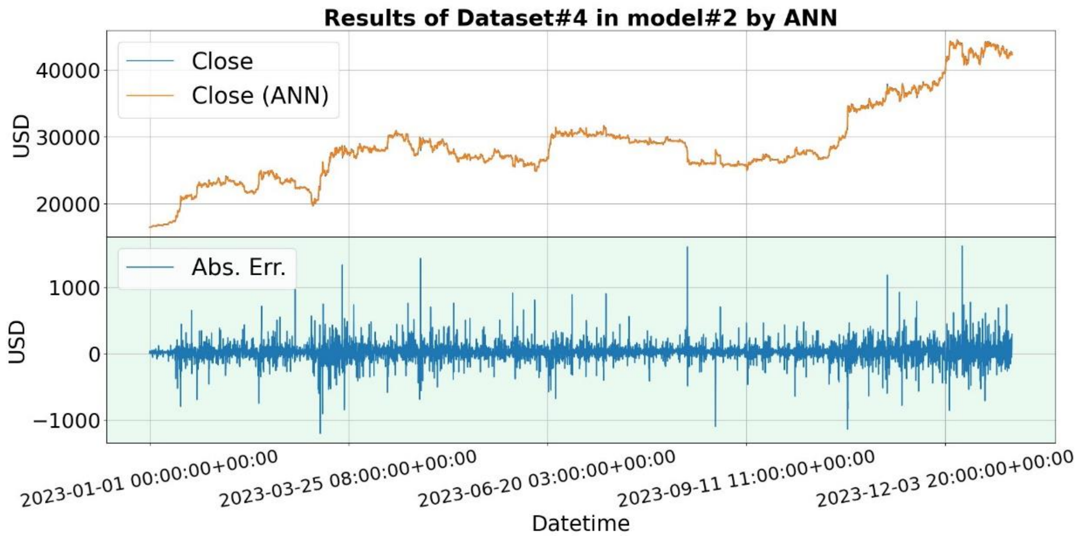
Table 6.1. Train and test scores of the datasets in our ANN application (for our prediction model#2).

	Dataset#1	**Datataset#2**	**Dataset#3**	**Dataset#4**	**Dataset#5**
Loss score (MSE) for train data	10380.2402	18120.8867	33973.2500	37744.3672	799102.7500
Loss score (MSE) for test data	12393.8584	14601.0264	13687.4004	19779.9512	798212.75




Results of Dataset#2 in model#2 by ANN
USD
44000
43000
42000
41000
40000
Close
Close (ANN)
200
0
−200
−400
Abs. Err.
2023-12-04 00:00:00
2023-12-06 02:00:00
2023-12-08 04:00:00
2023-12-10 06:00:00
Datetime


Results of Dataset#3 in model#2 by ANN
USD
44000
42000
40000
38000
Close
Close (ANN)
1000
500
0
−500
−1000
Abs. Err.
2023-12-01 00:00:00
2023-12-11 10:00:00
2023-12-21 20:00:00
Datetime


Results of Dataset#4 in model#2 by ANN
USD
40000
30000
20000
Close
Close (ANN)
Abs. Err.
1000
0
−1000
2023-01-01 00:00:00+00:00
2023-03-25 08:00:00+00:00
2023-06-20 03:00:00+00:00
2023-09-11 11:00:00+00:00
2023-12-03 20:00:00+00:00
Datetime

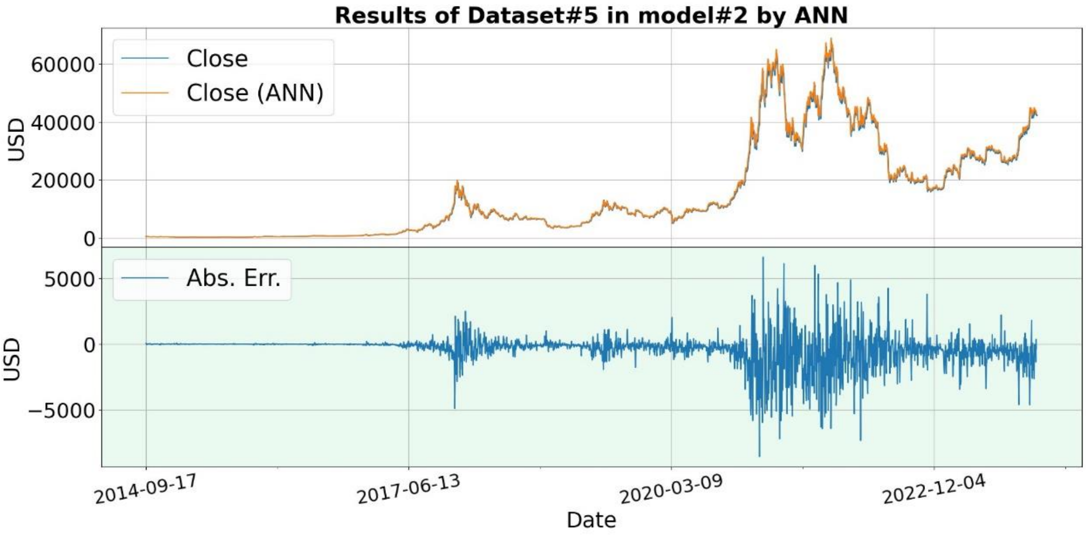
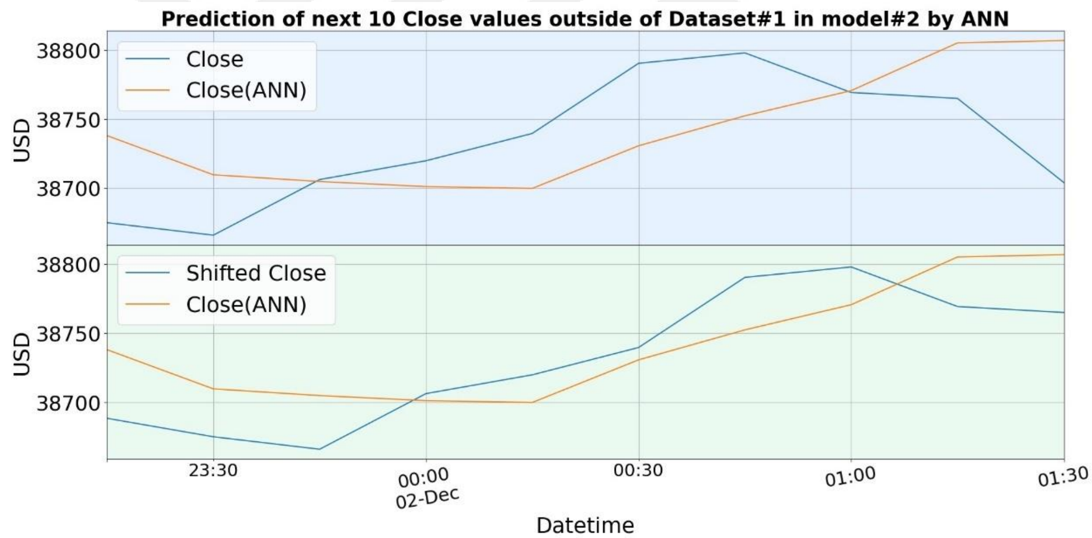
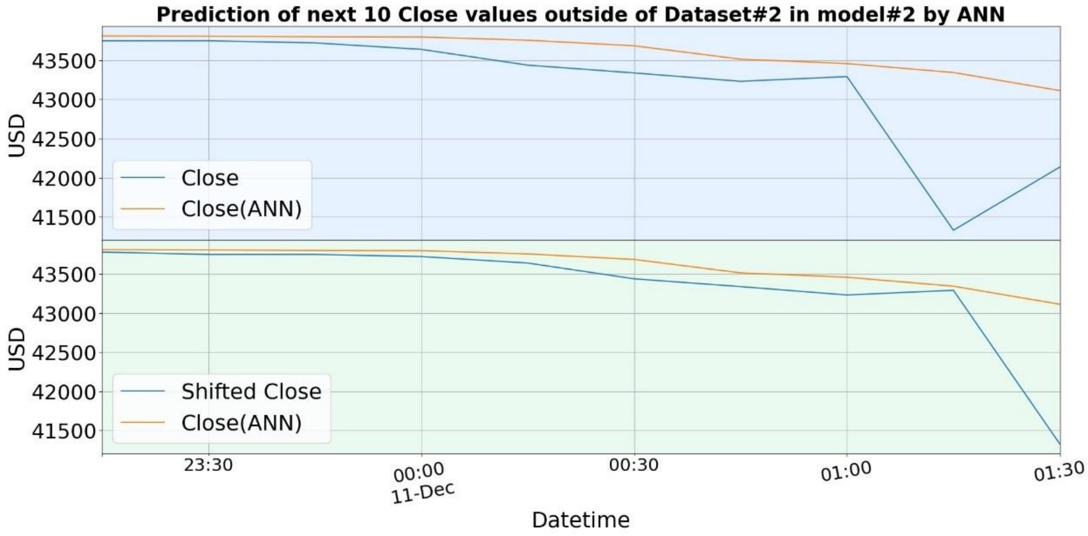
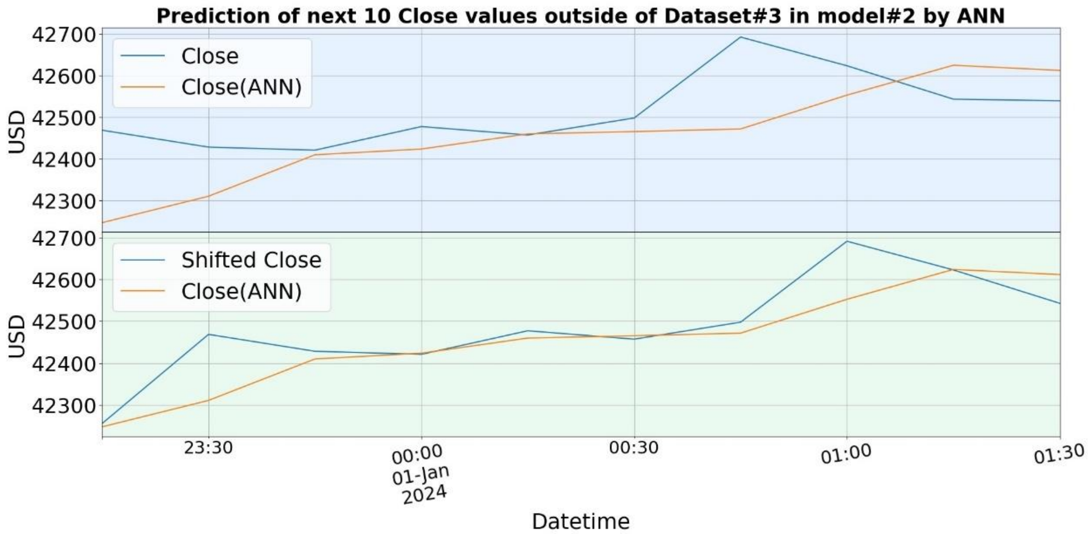
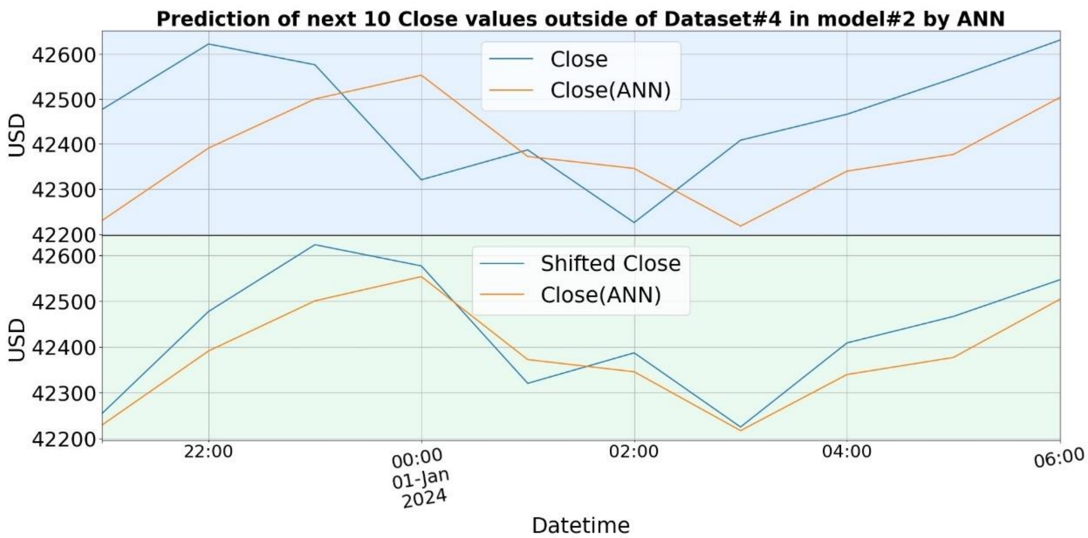


Figure 6.2. ANN-Results of our second prediction model (for model#2) within our datasets.




Prediction of next 10 Close values outside of Dataset#2 in model#2 by ANN
USD
Close
Close(ANN)
Shifted Close
Datetime


Prediction of next 10 Close values outside of Dataset#3 in model#2 by ANN
USD
Close
Close(ANN)
Shifted Close
Datetime


Prediction of next 10 Close values outside of Dataset#4 in model#2 by ANN
USD
Close
Close(ANN)
Shifted Close
Datetime

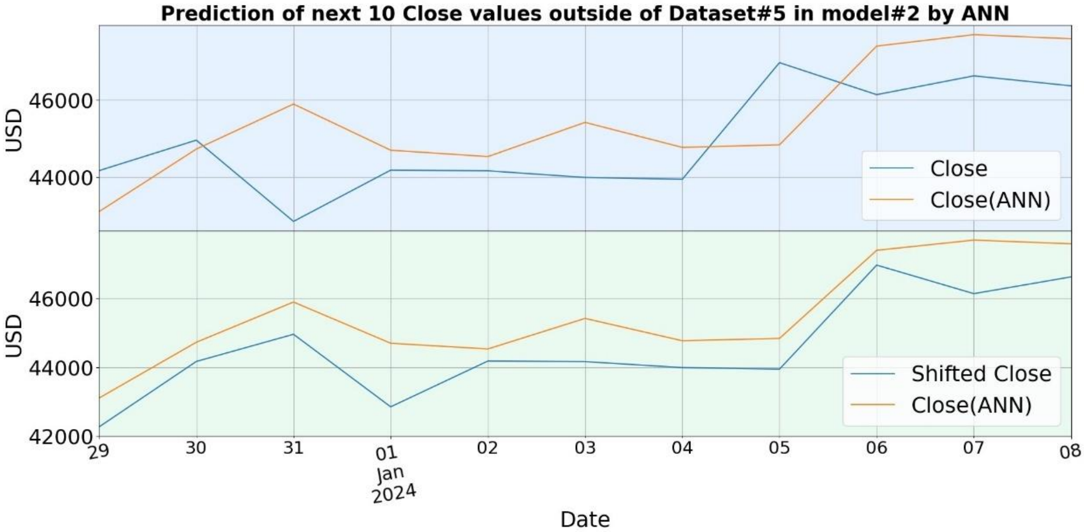
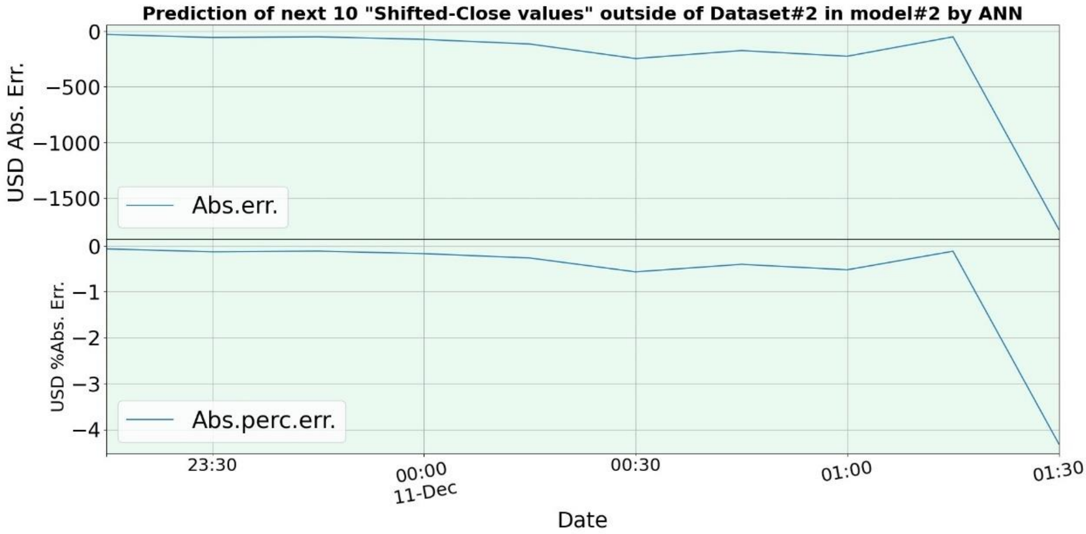
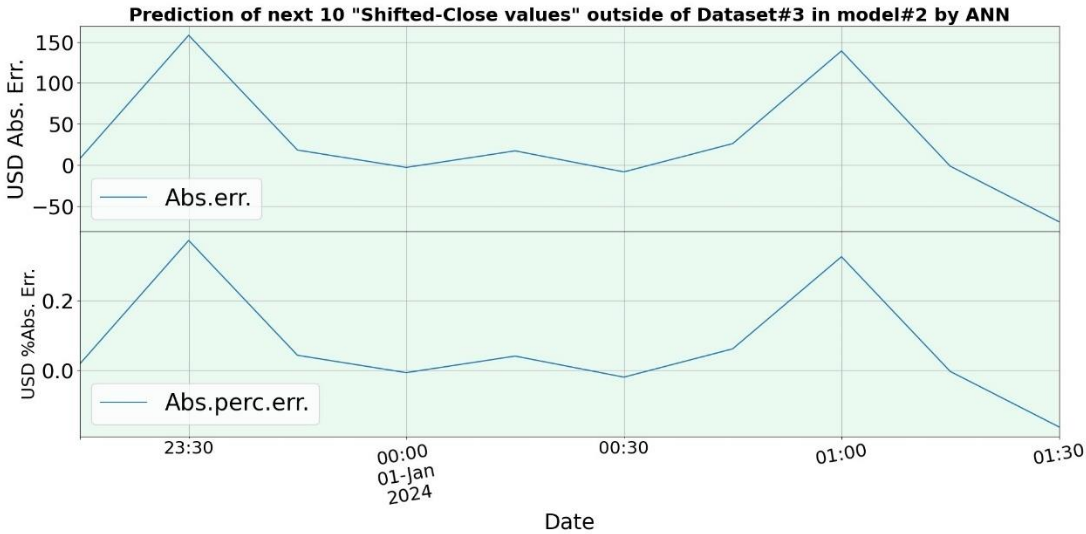
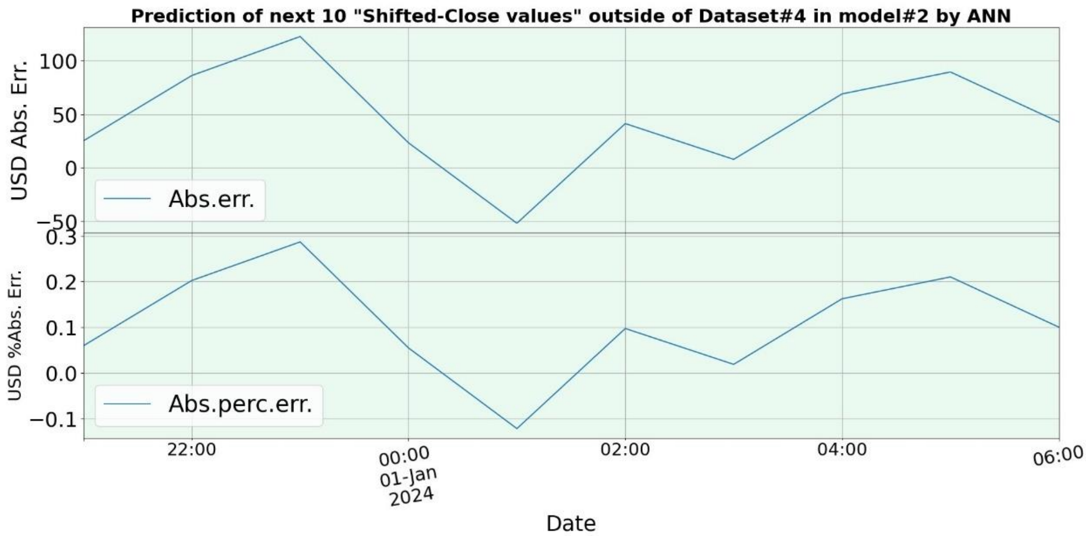


Figure 6.3. ANN-Results of our <u>second prediction model</u> (for <u>model#2</u>) in predicting future values (for the prediction of the next 10 data beyond the last data in our our datasets).


Prediction of next 10 "Shifted-Close values" outside of Dataset#2 in model#2 by ANN
USD Abs. Err.
Abs.err.
USD %Abs. Err.
Abs.perc.err.
23:30
00:00
11-Dec
00:30
01:00
01:30
Date


Prediction of next 10 "Shifted-Close values" outside of Dataset#3 in model#2 by ANN
USD Abs. Err.
Abs.err.
USD %Abs. Err.
Abs.perc.err.
23:30
00:00
01-Jan
2024
00:30
01:00
01:30
Date


Prediction of next 10 "Shifted-Close values" outside of Dataset#4 in model#2 by ANN
USD Abs. Err.
Abs.err.
USD %Abs. Err.
Abs.perc.err.
22:00
00:00
01-Jan
2024
02:00
04:00
06:00
Date

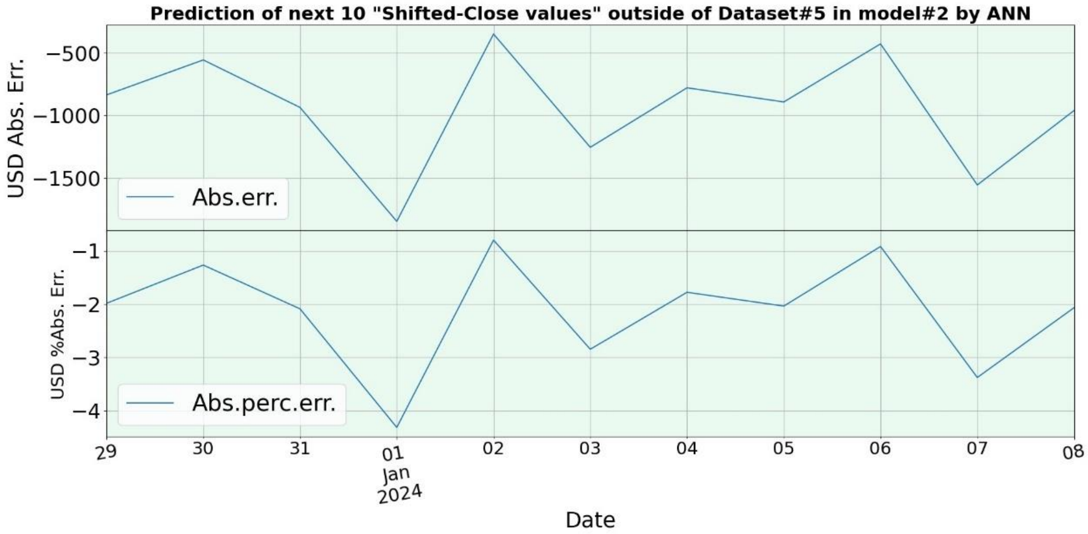


Figure 6.4. ANN-Absolute error and %Absolute error graphs of the ANN results of future predictions in model#2.

7. CONCLUSION

In this work, we studied the cryptocurrency concept and fundamental properties of blockchain in the context of the cryptocurrencies. We also studied the two fundamental ML regression models in the AI, namely, MLR and the ANN and how to use them to predict the cryptocurrencies. We studied here two models to predict the future values of BTC by using the candlestick diagrams of the past values. In the first model, we assume that “Open”-“High”-“Low” values of each candle can be used to determine the “Close” value of the same candle. In the second model, we assume that all four values (“Open”-“High”-“Low”-“Close”) of each three subsequent candles (making a total of twelve input values) can be used to predict the “Close” value of the next (the fourth) candle. We applied 20%-to-%80 test-train split to these two models firstly to prepare the training by the MLR and the ANN applications.

In our MLR study, we obtained the regression coefficients and compare the graphs of the prediction results with the graphs of the BTC data in our dataset. “Close” values are successfully predicted with a great fitting in the figures. After the training, we obtained very high learning scores as shown in Table 5.1 (for model#1) and Table 5.2 (for model#2) for both train and test data indicating that there is no need for regularization for both models. We performed it by using five different datasets involving diverse range of data (small and big data in various ranges) for our two-studied prediction models. For future predictions, we tried to find the “Close” values of the next 10 data outside our datasets. Actually, our intention is to check fitting of the two models for the future prediction. We used the “Open-High-Low” values of the next 10 data to predict the “Close” values of the same future candles in model#1. In our second model, no information about the future candles as input data is used as. Results show that these two models can be used for further analyses for future predictions. Graphs obtained in our results in Chapter 5 also show correct prediction of tendencies of the BTC (trends on increasing or decreasing behaviour).

We note here that since model#1 is used to establish a linear relationship between “Close” value and all three other values (“Open”, “High”, and “Low”) for the same candle, we can not actually use model#1 as a future prediction in the current form directly. Figure 5.3 and Figure 5.5 predicts the “Close” values of the next ten candles assuming that all three inputs (“Open”, “High”, “Low”) for the related ten subsequent candles are known (or predicted exactly) before these candles are created in real time. It also explains the great success in the future price prediction of BTC in Figures 5.3 and 5.5. By using the obtained regression coefficients in Table 5.1 in equations (3.2) and (3.7), desired MAE, MSE, RMS, and R^2 scores can be calculated. We calculated these scores for model#1 via our Python codes as given in Table 7.1. These scores also explain the correctness of the future predictions for model#1 in our graphs in Section 5.

Table 7. 1. Error estimate scores of the application of MLR in our prediction model#1.

Scores of Model#1 by LReg (within the whole dataset)					
Data-set#	Total Data#	MAE	MSE	RMSE	R^2
1	96	21.159441	819.873780	28.633438	0.992575
2	672	25.009852	1332.290655	36.500557	0.998667
3	2976	24.054195	1295.390907	35.991539	0.999372
4	8676	31.867127	2542.778822	50.425974	0.999927
5	3393	153.029427	104459.118125	323.201358	0.999601
Scores of Model#2 for future price prediction of BTC (For prediction of the next 10 data outside the dataset being trained) by LReg					
Data-set#	Total Data#	MAE	MSE	RMSE	R^2
6	10	9.464484	128.168695	11.321161	0.934201
6	10	61.241631	14472.470947	120.301583	0.974871
6	10	16.494012	484.398956	22.009065	0.928612
6	10	32.896315	1552.242752	39.398512	0.902524
6	10	434.496632	262429.725323	512.278953	0.838643

Model#2, contrary to model#1, is designed for predicting the future value ("Close" value) of the upcoming candle directly. In Figure 5.2, we can see that our MLR model predicts the "Close" value of the next candle by legging one candle behind the true "Close" value for each candle, hence making a future prediction. Graphs of future predictions (for the "Close" values of the next 10 candles) in Figure 5.4 and 5.6 represents our model prediction in continuum. To illustrate, for the prediction of the "Close" value of the next 7^{th} candle, we think that all the four Candle values of the last three candles ("no future data!") are known. Consequently, Figure 5.4 and 5.6 represents our model for the prediction of "Close" value of the nth candle where all the four values of the previous three candles (candle#: $n-1, n-2,$ and $n-3$) are known. If the $n+1, n+2, n+3, \ldots$ indexed candles are also to be predicted (as far-future prediction), each predicted value can be used to predict the "Close" value of the next (upcoming) candle iteratively, decreasing the accuracy given here. Consequently, model#2 is appropriate to predict the close value of just the next candle correctly. But far future values can be calculated iteratively. Here we predicted them in continuum for the prediction of just the next candle for each of the next 10 candles.

Regression coefficients regarding our MLR application is given in Table 5.2. All twelve inputs of the previous three candles can be used with these coefficients to find the "Close" value of the next (upcoming) candle. Again, by using the obtained regression coefficients in Table 5.2 in equations (3.2) and (3.7), desired MAE, MSE, RMS, and R^2 scores can be calculated. We calculated these scores for model#2 via our Python codes as given in Table 7.2. These scores also explain the correctness of the future predictions for model#2 in our graphs in Section 5.

Table 7.2. Error estimate scores of the application of MLR in our <u>prediction model#2</u>.

Scores of Model#2 by LReg (within the whole dataset)					
Data-set#	**Total Data#**	**MAE**	**MSE**	**RMSE**	**R^2**
1	96-3	46.300407	3412.149795	58.413610	0.963986
2	672-3	66.083321	8990.630242	94.818934	0.990627
3	2976-3	64.413679	9549.576759	97.721936	0.995320
4	8676-3	76.594070	16697.543909	129.218977	0.999521
5	3393-3	373.609472	609877.841960	780.946760	0.997673
Scores of Model#2 for future price prediction of BTC (For prediction of the next 10 data outside the dataset being trained) by LReg					
Data-set#	**Total Data#**	**MAE**	**MSE**	**RMSE**	**R^2**
6	10	11.990558	244.004644	15.620648	0.881670
6	10	13.646864	312.382895	17.674357	0.999352
6	10	4.622796	28.515760	5.340015	0.997746
6	10	5.061966	35.744355	5.978658	0.997799
6	10	126.616119	26887.319038	163.973532	0.986616

Since model#2 is appropriate for direct use in predicting the next (upcoming) "Close" value of the BTC, we also studied it in our ANN model as described in Section 6. We studied the ANN regression model only for our prediction model#2. Train and test scores of the datasets in our ANN application is given in Table 6.1. Although our prediction graphs in Section 6 show consistence with the exact results, in comparison to the graphs of the LReg graphs given in Section 5, the ANN model we studied is not much successful as the MLR model for future prediction. To illustrate, estimation scores of dataset#1 of the ANN model we studied is computed via our Python codes as given in Table 7.3. We here give only the results of Dataset#1 since it involves the smallest number of data (96-3=93 data for our present dataset and 10 data to predict as future dataset) with respect to the others. By comparison the values in Table 7.3 with the values in Table 7.1 and 7.2, we see the explanation of the weak prediction by the ANN model we studied with respect to the MLR model. Consequently, other ANN models (with other layer and/or neuron numbers with other activation functions) as well as advanced DL techniques such as CNN, LSTM can be designed for better prediction with the second prediction model we studied here. We see that model#2 can be used to predict future price value of BTC (as the "Close" value of the upcoming candle in the near future) directly by using any regression model within the AI but model#1 can be used indirectly, such as predicting the "Open"-"High"-"Low" values of the next (upcoming) candle first.

Table 7.3. Error estimate scores of the application of ANN for dataset#1 in our prediction model#2.

Scores of Model#2 by LReg (within the whole dataset#1)					
Data-set#	**Total Data#**	**MAE**	**MSE**	**RMSE**	**R^2**
1	96-3	74.664861	10292.309598	101.451021	0.891369
Scores of Model#2 for future price prediction of BTC (For prediction of the next 10 data outside the dataset being trained (=dataset#1)) by ANN					
Data-set#	**Total Data#**	**MAE**	**MSE**	**RMSE**	**R^2**
6	10	30.040234	1092.228810	33.048885	0.470322

REFERENCES

[1] Milutinović Monia. (2018). Cryptocurrency. Ekonomika, 64(1), 105-122.

[2] Kakavand, H., Kost De Sevres, N., & Chilton, B. (2017). The Blockchain Revolution: An Analysis of Regulation and Technology Related to Distributed Ledger Technologies. SSRN Electronic Journal.

[3] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

[4] Hileman, G. & Rauchs, M. (2017). Global cryptocurrency benchmarking study. *Cambridge Centre for Alternative Finance Reports*.

[5] Kent, P., & Bain, T. (2022). Bitcoin For Dummies (2nd ed.). Wiley.

[6] Antonopoulos, A. M. (2022). Mastering Bitcoin (2015th ed.). O'Reilly.

[7] Sushma, M., & Anamika, R. (2020). *A Brief Survey of Cryptocurrency Systems*, *11*(1), 76–82.

[8] Mukhopadhyay, U., Skjellum, A., Hambolu, O., Oakley, J., Yu, L., and Brooks, R. (2016). A brief survey of Cryptocurrency systems. 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand. pp. 745-752, doi: 10.1109/PST.2016.7906988.

[9] Investopedia. (2016, October 15). Cryptocurrency. Retrieved from Investopedia: http://www.investopedia.com/terms/c/cryptocurrency.asp

[10] Cryptocurrency. (2023, December, 1). In *Wikipedia.* https://en.wikipedia.org/wiki/Cryptocurrency

[11] Bitcoin. (2023, December, 1). In *Wikipedia.* *https://en.wikipedia.org/wiki/Bitcoin*

[12] Legality of cryptocurrency by country or territory. (2023, December, 1). In *Wikipedia.* https://en.wikipedia.org/wiki/Legality_of_cryptocurrency_by_country_or_territory

[13] Böhme, R., Christin, N., Edelman, B., & Moore, T. (2015). Bitcoin: Economics, Technology, and Governance. Journal of Economic Perspectives, 29(2), 213–238. https://doi.org/10.1257/jep.29.2.213

[14] Tschorsch, F., & Scheuermann, B. (2016). Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. In IEEE Communications Surveys

& Tutorials (Vol. 18, Issue 3, pp. 2084–2123). Institute of Electrical and Electronics Engineers (IEEE).

[15] Singh, A., & Krishna, V. (2022). A New Investment Opportunity: Bitcoin & Ethereum Cryptocurrency. International Journal of Scientific Research in Engineering and Management (IJSREM), 6(10), 1–19. https://doi.org/10.55041/IJSREM16525.

[16] Tapscott, D., Tapscott, A. (2016). Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. NY. Penguin.

[17] Narayanan, A., Bonneau, J., Felten, E., Miller, A., Goldfeder, S., 2016. Bitcoin and Cryptocurrency Technologies, Princeton University Press.

[18] Crosby, M., Pattanayak, P., Verma, S. and Kalyanaraman, V. (2016) Blockchain Technology: Beyond Bitcoin. Applied Innovation, 2, 71.

[19] Şekil Avunduk, H., & Aşan, H. (2018). Blok Zinciri (Blockchain) Teknolojisi ve İşletme Uygulamaları: Genel Bir Değerlendirme. Dokuz Eylül Üniversitesi İktisadi İdari Bilimler Fakültesi Dergisi, 33(1), 369-384. https://doi.org/10.24988/deuiibf.2018331746

[20] Gokoglan, K., Cetın, S. & Bilen, A. (2022). Blockchain Technology and Its Impact on Audit Activities. Journal of Economics Finance and Accounting, 9 (2), 72-81. DOI: 10.17261/Pressacademia.2022.1567.

[21] Ünal, G., & Uluyol, Ç. (2020). Blok zinciri teknolojisi. INTERNATIONAL JOURNAL OF INFORMATICS TECHNOLOGIES, 13(2), 167–175. https://doi.org/10.17671/gazibtd.516990

[22] Watanabe, H., S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu and J. Kishigami. (2016). Blockchain contract: Securing a blockchain applied to smart contracts: IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, pp. 467-468, doi: 10.1109/ICCE.2016.7430693.

[23] Makridakis, S., & Christodoulou, K. (2019). Blockchain: Current Challenges and Future Prospects/Applications. Future Internet, 11(12), 258. https://doi.org/10.3390/fi11120258

[24] Chu, J., Nadarajah, S., & Chan, S. (2015). Statistical analysis of the exchange rate of Bitcoin. PLOS ONE, 10(7), e0133678. https://doi.org/10.1371/journal.pone.0133678

[25] Hua, Y. (2020). Bitcoin price prediction using ARIMA and LSTM. E3S Web of Conferences, 218, 01050. https://doi.org/10.1051/e3sconf/202021801050

[26] Kervancı, I. S. & Akay, M. F. (2020). Review on bitcoin price prediction using machine learning and Statistical Methods. *Sakarya University Journal of Computer and Information Sciences*, *3*(3), 272–282. https://doi.org/10.35377/saucis.03.03.774276.

[27] Alahmari, S. A. (2019). Using machine learning ARIMA to predict the price of cryptocurrencies. The ISC International Journal of Information Security, 11(3), 139–144. https://doi.org/10.22042/isecure.2019.11.0.18

[28] Gullapalli, S. (2018), "Learning to Predict Cryptocurrency Price Using Artificial Neural Network Models of Time Series".

[29] Alkhodhairi, R. K., Aljalhami, S. R., Rusayni, N. K., Alshobaili, J. F., Al-Shargabi, A. A., & Alabdulatif, A. (2021). Bitcoin Candlestick Prediction with Deep Neural Networks Based on Real Time Data. Computers, Materials & Continua, 68(3), 3215–3233. https://doi.org/10.32604/cmc.2021.016881

[30] Wardak, A. B., & Rasheed, J. (2022). Uzun kısa vadeli bellek tekrarlayan sinir ağı kullanarak Bitcoin kripto para birimi fiyat tahmini. European Journal of Science and Technology. https://doi.org/10.31590/ejosat.1079622

[31] Greaves, A, Au, B. (2015). "Using the bitcoin transaction graph to predict the price of bitcoin". Department of Computer Science, Stanford University, California, United States of America, Scientific Report, 68.

[32] Atlan, F., & Pençe, İ. (2021). Forecasting of Bitcoin Price Using the Multilayer Perceptron Technique. Scientific Journal of Mehmet Akif Ersoy University, 4(2), 68-74.

[33] Guo, T., & Antulov-Fantulin, N. (2018), "An Experimental Study of Bitcoin Fluctuation Using Machine Learning Methods", arXiv preprint arXiv:1802.04065.

[34] Guo, T., Bifet, A., & Antulov-Fantulin, N. (2018), "Bitcoin Volatility Forecasting with a Glimpse into Buy and Sell Orders", In 2018 IEEE International Conference on Data Mining (ICDM), (989-994).

[35] Ferdiansyah, F., Othman, S. H., Radzi, R. Z. R. M., Stiawan, D., Sazaki, Y., & Ependi, U. (2019). A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market. *2019. International Conference on Electrical Engineering and Computer Science (ICECOS)*. https://doi.org/10.1109/icecos47637.2019.8984499

[36] Sezer, Ö. B., & Özbayoğlu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Applied Soft Computing, 70, 525–538. https://doi.org/10.1016/j.asoc.2018.04.024

[37] Özbayoğlu, A. M., Gudelek, M. U., & Sezer, Ö. B. (2020). Deep learning for financial applications: A survey. Applied Soft Computing, 93, 106384. https://doi.org/10.1016/j.asoc.2020.106384

[38] Santur, Y. (2023). A novel financial forecasting approach using deep learning framework. Computational Economics, 62(3), 1341–1392. https://doi.org/10.1007/s10614-023-10403-5

[39] Sarikaya, A., & Aslan, S. (2022). Deep learning and machine learning based sentiment analysis on BitCoin (BTC) price prediction. NATURENGS MTU Journal of Engineering and Natural Sciences Malatya Turgut Ozal University. https://doi.org/10.46572/naturengs.1182766

[40] Parekh, R., Patel, N., Thakkar, N., Gupta, R., Tanwar, S., Sharma, G., Davidson, I. E., & Sharma, R. (2022). DL-GuESS: Deep Learning and Sentiment Analysis-Based Cryptocurrency Price Prediction. IEEE Access, 10, 35398–35409. https://doi.org/10.1109/access.2022.3163305

[41] McCoy, M. E., & Rahimi, S. (2020). Prediction of highly volatile cryptocurrency prices using social media. International Journal of Computational Intelligence and Applications, 19(04). https://doi.org/10.1142/s146902682050025x

[42] Kaminski, J. (2014, June 30). Nowcasting the Bitcoin Market with Twitter Signals. arXiv.org. https://arxiv.org/abs/1406.7577

[43] Shen, D., Urquhart, A., & Wang, P. (2019). Does twitter predict Bitcoin? Economics Letters, 174, 118–122. https://doi.org/10.1016/j.econlet.2018.11.007

[44] Russell, S., & Norvig, P. (2003). Artificial intelligence - a modern approach, 2nd Edition. Prentice Hall series in artificial intelligence.

[45] Chollet, F. (2017). Deep Learning with Python. Manning Publications.

[46] Alpaydin, E. (2014). Introduction to machine learning. MIT Press.

[47] Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow 2, 3rd Edition.

[48] Ketkar, N., Moolayil, J. (2021). Introduction to Machine Learning and Deep Learning. In: Deep Learning with Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-5364-9_1

[49] Haykin, S. S. (2009). Neural networks and learning machines. Prentice Hall.

[50] Chapelle, O., Zien, A., & Schölkopf, B., (Eds.) (2006). Semi-supervised learning. Cambridge, MA MIT Press.

[51] Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Techniques and Tools to Build Learning Machines. O’Reilly Media.

[52] Charts-Blockchain Size (MB). (official web page of the BTC). https://www.blockchain.com/explorer/charts/blocks-size

[53] Wikipedia contributors. (2023, December 5). Blockchain. Wikipedia. https://en.wikipedia.org/wiki/Blockchain

[54] Finance-Yahoo, Cryptocurrencies (Selected and arranged by setting the Watchlist). Retrieved December 11, 20223, from https://finance.yahoo.com/crypto/

[55] Finance-Yahoo, Bitcoin USD (BTC-USD). Retrieved December 11, 20223, from https://finance.yahoo.com/chart/BTC-USD

[56] Kaabar, S. (Ed.). (2023). Mastering Financial Pattern Recognition Finding and Back-Testing Candlestick Patterns with Python (1st ed.). USA: O’Reilly.

[57] Nison, S. (2001). Japanese candlestick charting techniques: a contemporary guide to the ancient investment techniques of the Far East (2nd ed.). NY: NYIF.

[58] Deniz, C. (2023). Modern Computer Tomography with Artificial Intelligence and Deep Learning Applications. Artificial Intelligence Theory and Applications, 3(2), 123-136.

[59] Samuel, Arthur L. (1959). Some Studies in Machine Learning Using the Game of Checkers IBM Journal of Research and Development 44:1.2, 210–229.

[60] Mitchell, T. M. (1997). Machine learning (Vol. 1, No. 9). New York: McGraw-hill.

[61] What Are the Differences Between Supervised, Unsupervised and Reinforced Machine Learning? (2023, December 5). https://ded9.com/what-are-the-differences-between-supervised-unsupervised-and-reinforced-machine-learning/

[62] Montgomery, D.C., Peck, E.A. and Vining, G.G. (2001) Introduction to Linear Regression Analysis. 3rd Edition, Wiley, New York. P.81.

[63] Montgomery, D. C., & Runger, G. C. (2014). Applied statistics and probability for engineers (6th ed.). John Wiley & Sons.

[64] Deniz, C., Canbula, B. (2022.11.17). Bitcoin Price Prediction by Train Test Education Of Linear Regression Via Machine Learning In Python. 6th International Conference, ICENTE. Konya, Turkey.

[65] Deniz, C., Canbula, B. (2023.06.14). Suggestion of One and Four-Candle-Sick Models for Short-Term Bitcoin Price Prediction by Linear Regression in Machine Learning. 3rd International Graduate Studies Congress (IGSCONG'23).

[66] Ramasubramanian, K., Singh, A. (2019). Machine Learning Using R, 2nd edn., Apress, 2019.

[67] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review. 65(3), 386-408.

[68] Activation function. (2023, December, 1). In *Wikipedia.* https://en.wikipedia.org/wiki/Activation_function

[69] Atlan, F. (2019). Kripto para değerlerinin yapay zekâ teknikleri ile tahmini. Burdur Mehmet Akif Ersoy Üniversitesi, Sosyal Bilimler Enst., Burdur, Yönetim Bilişim Sistemleri AbD. (MS thesis).

[70] Salouhou, O. (2019). El Yazısı Karakter Tanıma ve Resim Sınıflandırmada Derin Ögrenme Yaklaşımları. Fatih Sultan Mehmet Vakıf Üniversitesi Lisansüstü Eğitim Enstitüsü, Bilgisayar Müh. AbD. (MS thesis, corrected thesis)