FORECASTING THE FUTURE PRICE MOVEMENT OF CRYPTOCURRENCY ASSETS BY CONVOLUTIONAL NEURAL NETWORK

A Thesis

by

Aysel Çetin

Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Data Science

Özyeğin University
May 2023

FORECASTING THE FUTURE PRICE MOVEMENT OF CRYPTOCURRENCY ASSETS BY CONVOLUTIONAL NEURAL NETWORK

Approved by:

Asst. Prof. Erinç Albey, Advisor
Dept. of Industrial Eng.
Özyeğin University

Prof. Mehmet Güray Güler
Dept. of Industrial Eng.
Yıldız Technical University

Asst. Prof. Mehmet Önal
Dept. of Industrial Eng.
Özyeğin University

Date Approved: May 24, 2023

To my loved ones and friends who supported me throughout my journey.

ABSTRACT

Digital or virtual currency known as cryptocurrency uses cryptography for security and is not controlled by a central bank. Cryptocurrencies control the issue of new units and record transactions using decentralized technology, such as blockchain. Cryptocurrencies are entirely digital and have no physical form, in contrast to traditional currency, which is real and backed by a government or financial institution. Although Bitcoin was the first and best-known cryptocurrency, there are now thousands of other coins in use, including Ethereum,Tether, BNB, XRP etc. Bitcoin's value can be extremely unstable and it is frequently utilized as an investment or speculative asset. In some locations, it can also be used to make purchases of products and services, and some companies even accept it as payment. Due to the formation of this sector in relation to the growth in earnings and followers, all focus turned in the direction of the cryptocurrency market. Despite the abundance of studies that have been done in the past for this area, less image processing research has been done for the next movement prediction.

This paper uses Bitcoin (BTC) dataset and tries to create a tool to project the upcoming price direction. The target variable is a binary type as the next movement will decrease or increase direction. The challenge of forecasting the next day's price movement involves learning from the information from the previous day by transforming them to the images. Since the main goal is using image processing for prediction, Convolutional Neural Networks, one of the most well-known deep learning techniques, and human judgment will be used. As a result, it is aimed to create an algorithm that makes a successful buy-sell decision on BTC and to achieve profitability. However, if the algorithm created for this study has adequate performance and structure on

BTC, it will be simple to adapt it to other cryptocurrency kinds in the future.

ÖZETÇE

Kripto para olarak bilinen dijital veya sanal para birimi, güvenlik için kriptografi kullanır ve herhangi bir merkezi finans kuruluşu tarafından kontrol edilmez. Kripto para birimleri, blok zinciri gibi merkezi olmayan teknolojiyi kullanarak yeni birimlerin çıkarılmasını ve işlemlerin kaydedilmesini kontrol eder. Kripto para birimleri, gerçek olan ve bir devlet yönetimi veya finans kurumu tarafından desteklenen geleneksel para biriminin aksine, tamamen dijitaldir ve fiziksel bir formu yoktur. Bitcoin ilk ve en iyi bilinen kripto para birimi olmasına rağmen, şu anda kullanımda olan Ethereum, Tether, BNB, XRP vb. binlerce başka madeni para da var. Bitcoin'in değeri son derece istikrarsız olabilir ve sıklıkla bir yatırım aracı veya spekülatif varlık olarak kullanılabilir. Bazı lokasyonlarda ürün ve hizmet satın almak için dahi kullanılabilmektedir, dolayısı ile bazı firmalar bunu ödeme aracı olarak kabul etmektedir. Bu alanın kazanç ve takipçi artışı yani popülaritesine bağlı olarak gelişmesi nedeniyle tüm odaklar kripto para piyasası yönüne çevrilmiş durumdadır. Geçmişte bu alan için yapılan çalışmaların çokluğuna rağmen bir sonraki yön tahmini için yeterince görüntü işleme modeli kurarak araştırma yapılmamıştır.

Bu çalışma, Bitcoin (BTC) veri setini kullanır ve gelecek fiyat yönünü tahmin etmek için bir algoritma oluşturmaya çalışır. Bir sonraki hareket yönü azalacak veya artacak şeklinde olduğu için hedef değişken ikili tiptedir. Bir sonraki adımın fiyat hareketini tahmin etmenin zorluğu, önceki hareketlere ait bilgileri resimlere dönüştürerek öğrenmeyi içerir. Temel amaç tahmin için görüntü işlemeyi kullanmak olduğundan, en iyi bilinen derin öğrenme tekniklerinden biri olan Konvolüsyonel Sinir Ağları ve iş bilgisi kullanılacaktır. Sonuç olarak BTC üzerinde başarılı bir al-sat kararı veren bir algoritma oluşturmak ve bu karar neticesinde karlılığa ulaşmak

amaçlanmaktadır. Ancak bu çalışma için oluşturulan algoritma BTC üzerinde yeterli performansa ve yapıya sahip olursa gelecekte diğer kripto para türlerine uyarlaması da mümkün olacaktır.

ACKNOWLEDGEMENTS

I would like to start by thanking Erinç Albey, my adviser, for his assistance and direction throughout my data science studies. His knowledge of the subject and professional experience were really helpful to me while I carried out my research. Working with him was very instructive and enjoyable for me. I would like to express my gratitude to my committee members for letting my defense. I am also appreciative of the love, continuous support and understanding that I have received from my family, especially from my husband Mustafa Çakır, my mother Fatma Sayınbaş, and my sister Büşra Çetin when undertaking my research and writing my project. I also want to express my gratitude to my friends for standing by me all these years. Last but not least, I would like to express my gratitude to those who have passed away for making me who I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation and Contributions

Nowadays machine learning (ML) and artificial intelligence (AI) make life easier in many areas thanks to the remarkable improvement of technology, data accessibility, data diversity and data storability. With the basic definition, ML is a field of study devoted to comprehending and developing techniques that use data to learn and improve performance on a certain set of tasks. It is also seen as a part of AI. On the other hand, To create intelligent software and systems, AI is achieved by examining the cognitive process and the patterns of the human brain. ML and AI applications are used in different areas with the development of technology such as recommendation engines, surveillance systems, advertising and business intelligence and facial recognition etc.

Especially for prediction in financial markets, new and powerful AI-driven methodologies are supported by the emergence of algorithmic high-frequency and high-volume trading. Besides, all the attention shifted to the direction of the cryptocurrency market due to the emergence of this area regarding the increase in profits and followers. However, rising crypto-currency markets have highly scaled fluctuations compared to other financial equities. All these reasons give a great opportunity to use AI-driven tools for crypto-currency predictions.

Bitcoin (BTC) is subject to prediction in this project as it is counted as the locus classicus of this market in line with the popularity, large market capitalization, and history. In addition to this, the tool developed with this study will be easily applicable to other cryptocurrency types afterwards if it has sufficient performance and structure on BTC.

In this study, the next price direction is predicted for the BTC. Next price movement prediction is the problem of learning from the past days' information to predict the next day's price movement. For this purpose, convolutional neural networks, which is one of the most well-known algorithms of deep learning, and expert judgment will be used. Although there are so many studies in literature for this aim, there is a lack of research using image processing.

The Binance [1] API might be used as the primary data source for any apps or services that require real-time market data and trading capabilities on the Binance exchange. Additionally, Binance provides historical data so that users can analyze earlier market data. Therefore, BTC data was provided form Binance in this study. Afterwards, it was transformed for image processing with CNN model as detailed in Chapter 3.

In addition, descriptive analysis was applied in section 4 in order to understand if the dataset prepared is available and sufficient to develop a statistical model regarding the target variable and explanatory variable decided to use. Finally, the CNN model was built as described in Section 5.

1.2 Organization of the Thesis

The rest of the thesis is organized as follows: Chapter 1.3 provides a literature review of the relevant topics, including cryptocurrency and Bitcoin technology, and the methodologies used in the study (Convolutional Neural Network method). Chapter 2 explains detailed problem description regarding the decision of graph type, predictor variable,target variable and expert judgment to use in this study. Chapter 3 describes the data architecture in detail thoroughly. Chapter 4 presents the results of the descriptive analyses of the target variable and explanatory variables, and Chapter 5 describes the CNN model development steps including image pre-processing and building the CNN model. Chapter 6 presents area estimation results for this study by giving final model results and comparisons between alternatives. Finally, Chapter 7 represents conclusion and discussion on the results and potential improvements for this study.

1.3 Previous Work

There are many factors that can affect the market price of a product and forecasting in this area has been difficult for people throughout history by taking all these factors into account. The estimation process with the help of traditional technical analysis methods has evolved into more modern advanced analytical methods with the development of ML / AI techniques. People have always tried to clarify this problem with scientific studies. Therefore, there exists a large body of literature devoted to this subject.

1.3.1 Cryptocurrency and Bitcoin (BTC)

The definition of a blockchain is a public ledger where all transactions are recorded in blocks of chain, and the chain expands each time a block is added to it [2]. In a nutshell, blockchain serves as the basis for cryptocurrencies. It is essentially a collection of linked informational blocks on an electronic ledger. A group of transactions are contained in each block, and they have all been individually validated by every validator on the network. On the other hand, cryptocurrency is a decentralized digital currency that is built on blockchain technology and does not rely on banks to validate transactions. Being a peer-to-peer system that allows anybody, anywhere to send and receive payments is its most crucial characteristic. Because it is an internet platform, a public ledger keeps track of all transactions involving cryptocurrency when they are made. Cryptocurrency is stored in digital wallets.

Bitcoin (BTC) is one of many cryptocurrencies, a digital asset intended to serve as money and a means of payment that eliminates the need for third parties to be involved in financial transactions. An anonymous developer or group of developers called by the name Satoshi Nakamoto created Bitcoin in 2009.

With the increase in popularity and value in recent years, trading with such cryptocurrencies has come into prominence. There are a couple of main advantages and disadvantages of trading cryptocurrencies [3] as listed in below.

Advantages:

- Drastic fluctuations: While the quick swings in intraday prices might provide traders fantastic opportunities to make money, they also carry a higher level of risk.
- 7-d/24-h market: There are no time and place restrictions for trading in the cryptocurrency market.
- Near anonymity: Using cryptocurrency to make online purchases of products and services eliminates the need to reveal one's identity.
- Peer-to-peer transactions: There is no need for intermediaries from financial institutions, which can lower transaction costs.
- Programmable "smart" capabilities: Some cryptocurrencies can also offer its owners restricted ownership and voting privileges as well as represent a portion of ownership in tangible goods like artwork or real estate.

Disadvantages:

- Scalability problem: There is a huge number of transactions and the speed of transactions.
- Cybersecurity issues: Cryptocurrencies are vulnerable to cyber security lapses and can get into the wrong hands.
- Regulations: Not enough regulation for this market.

As a common drawback, the cryptocurrency market is generally speculative which means it is affected by traders behaviors in line with majority movement and huge trading volumes. Therefore, the literature on the issue of BTC price formation includes some empirical studies to examine its driving factors. Author in [4] initially assumed that the Bitcoin market was made up entirely of speculative traders. In order to prove that he assessed the relationship between Bitcoin and the volume of searches on Wikipedia and Google Trends. His findings revealed a clear asymmetry in the relationship between BTC price and search queries, pointing to speculation and trend-chasing as the primary forces behind BTC's price dynamics in the cryptocurrency market.

Below figure presents the percentage of total market capitalisation over years [3]. As seen in the figure, BTC remarkably dominates other coins. Therefore, BTC is subject to prediction in this project due to its popularity, large market capitalization, and history. In addition to this, the tool developed with this study will be easily applicable to other cryptocurrency types afterwards if it has sufficient performance and structure on BTC.

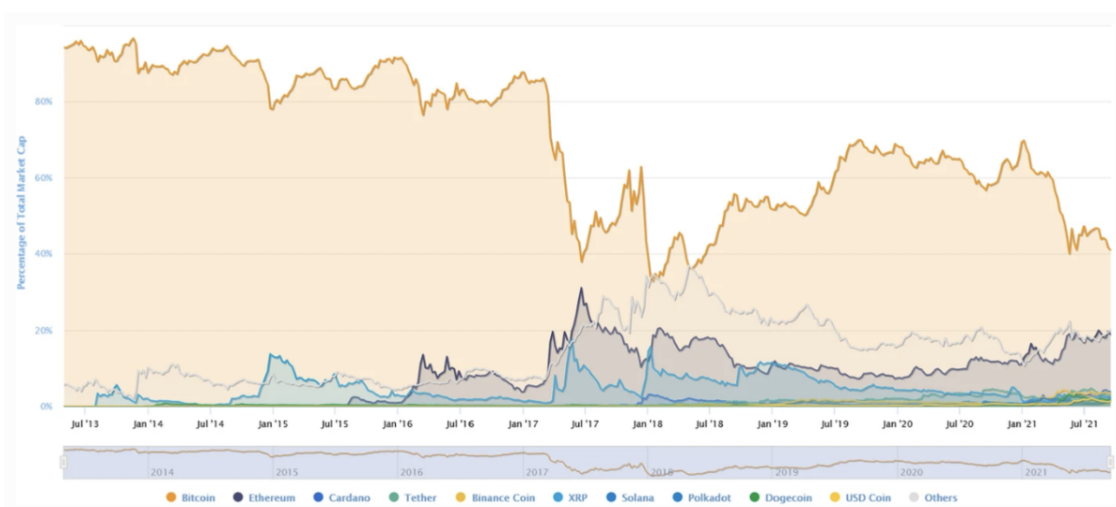

Figure 1: Percentage of total market capitalisation

1.3.2 Convolutional Neural Network (CNN)

Currently, ML and AI has become a frequently used method in finance, as in many other fields. These methods generally focus on solving two types of problems that are classified as supervised or unsupervised depending on the variable to be estimated. Briefly, the target variable is known in advance and given to the model during development in the supervised method while it is not in the unsupervised method. On the other hand, supervised methods can be divided basically into models with binary targets or models with continuous targets, depending on the type of target variable defined at the beginning of the modeling process. A supervised method with binary target will be used for this study.

There are classical machine learning algorithms such as logistic regression and they are known as easier to understand and explain. For example, Authors in [5, 6, 7] proposed models using several techniques such as linear regression, bayesian regression, SVM regression, logistic regression and SVM and analyzed BTC price prediction.

On the basis of confirmed historical data, time series models can also be used to forecast events. Moving average, smooth-based, and ARIMA are examples of common types of these models. The fundamental idea behind the ARIMA model is to evaluate the trend and seasonality of the series before removing them to create a stationary series. Techniques for statistical forecasting can be applied in this series. The predicted values would then be scaled back to the original values by applying trend and seasonality limitations [8]. However, it is more appropriate to use these models when a complex model is not desired and when working with less complicated data structures with less frequency. In such cases, the use of these models can produce a good-performing solution. But sometimes there may be complex problems or very high frequency datasets that such simple and classical methods cannot solve with high performance.

In these cases, deep learning methods come into play with their complex structures and very high performance solutions. They are also called black-box models because of the complexity of the model structure and the inability to clearly see the effects/coefficients of the variables on the final decision. Especially in recent years, the increase in the accessibility of such methods and the rise in the availability of exponentially increased data frequency in different types enable us to encounter many examples in this field as well as create notable popularity for these methods. An artificial neural network called Long Short-Term Memory (LSTM) is employed in deep learning and artificial intelligence. LSTM features feedback connections as an alternative to typical feed-forward neural networks. By maintaining a more stable error, LSTM helps the network to learn more about several time increments. As a result, the network can learn to establish lasting trust. The forget and remember gates of the LSTM cell enable the cell to select which information to transmit or block based on its strength and relevance. Therefore, as can be seen in many researches, the LSTM method generally outperforms simpler time series estimation methods such as ARIMA. [8, 9, 10]. Also, hybrid models were also tried to obtain better prediction performance. In [11], they provide a brand-new hybrid model that ensures both a descriptive and a predictive angle. For this reason, they employed LSTM networks and hidden markov models to characterize historical cryptocurrency movements and forecast future movements. As a result, they obtained a more successful model than the simpler structured models they compared in their study.

In some studies, CNN CNN (convolutional neural network) was employed for time series prediction .For instance [12] presents an innovative method for predicting the direction of the bitcoin price based on 1D CNN models. The suggested models forecast whether the price of bitcoin will be higher or lower than the most recent value in the time series after z days. Contrary to the fact that CNN models are typically used for image identification, 1D CNN models can be used for prediction tasks using time series as applied in this study. Additionally, various cutting-edge deep learning techniques, including CNNs, ResNets, a hybrid of CNNs and RNNs (CRNN), DNNs, and their ensemble models for predicting BTC prices are examined and compared in [13].

CNN with image processing is decided to use in this study, which is one of the deep learning algorithms, since there is a very high frequency data available to use and image processing is aimed which is a very complex problem.

CNN is a subset of deep learning. It is also one of the various types of artificial neural networks (ANN) which are used for different applications and data types. Below graph presents the last 13 years from google trends and compares CNN and ANN over years to see how much the usage rate has improved. As can be seen in the graph, CNN has become a more popular method in recent years although it emerged later and is a types of ANN.

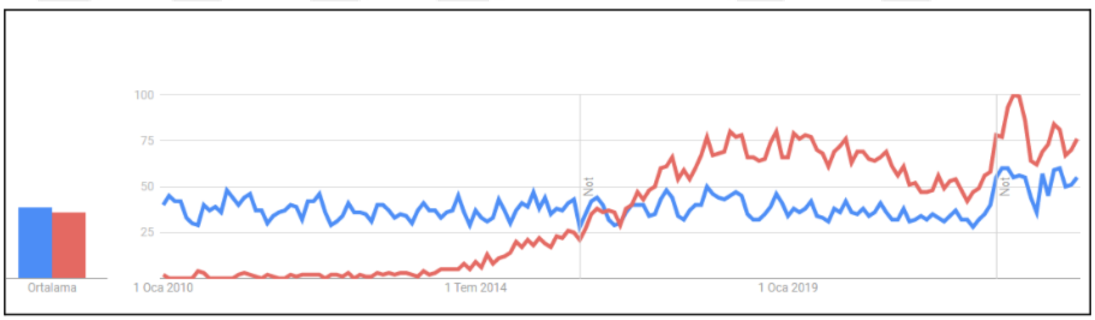

Figure 2: Google trends results, blue CNN and red ANN

A CNN is a particular type of network design for deep learning algorithms that is utilized for tasks like image recognition and pixel data processing. The CNN algorithm can be applied to signal, image, audio, and time series data. A CNN uses concepts from linear algebra, such as matrix multiplication, to find patterns in an image. Similar to an ANN architecture, a CNN architecture is designed to mimic the human brain and consists of several intricate neurons to analyze visual or acoustic data.

By calibrating the input weights and biases, the CNN algorithm learns through optimization. The basic objective is to categorize training data and generalize it to new datasets in order to determine the appropriate value for these weights. To find the proper weights, there isn't a closed-form solution, though. As a result, iteration setup is required to optimize weights, reduce loss function, and increase model prediction accuracy. Both forward and backward propagations were used for this aim. In brief, forward propagation is the process of feeding input data through a network in a forward direction to produce the output with initial weights, while backward propagation is the process of moving from the output to the input layer and ensures that weights are updated to minimize loss function. The technique is essentially depicted in the diagram below. This study uses the CNN algorithm to forecast the future price direction for bitcoin.

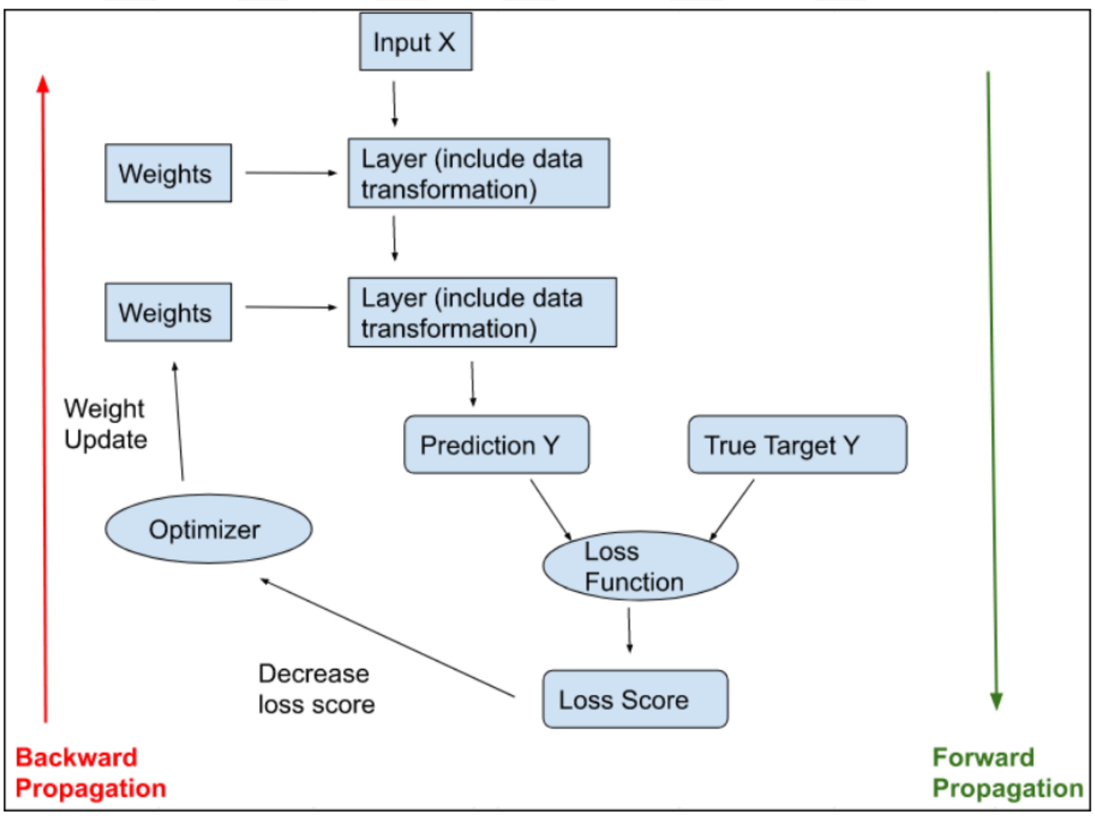

Figure 3: Basic illustration of learning process

CHAPTER II

DETAILED PROBLEM DESCRIPTION

Identifying the movements correctly for cryptocurrency prices might be very profitable. There are so many classical methods built to cover this aim in this area and an example of this could be technical analysis metrics. Technical analysis seeks to predict price movements by examining historical data, mainly price and volume. These metrics were mostly leveraged techniques like statistical analysis and behavioral economics. For instance; simple moving average (SMA), relative strength index (RSA), bollinger bands, moving average convergence divergence (MACD) etc.

On the other hand, estimating next price movements and trends in the stock or cryptocurrency market is a classical, challenging, overwhelming or esoteric issue for investors. It is hard to investigate market psychology to identify opportunities to profit. Moreover, using a particular trading approach may not be enough and it may be necessary to follow many signs, which is quite a tedious task. Also, it is not easy to develop a disciplined strategy that you can follow without letting emotions or second-guessing get in the way. In addition to this, it is necessary to have sufficient experience and self-confidence in this field in order to use these technical analysis methods. Last but not least, it may be necessary to constantly follow the trends in order not to miss any sign. All in all, the issue of how to make the decision and how to protect from great losses has always been a challenging problem for investors.

Thus, the main aim of this study is to prevent all above problems by using advanced analytical methods and to predict the next price movements using the historical data of BTC, which is the most favorite one of the cryptocurrencies. These kinds of advanced analytical models have started to be used for price predictions with the emergence and widespread use of ML and AI methods as discussed in the previous section. Specifically, the CNN algorithm and image processing were tried to use for this purpose in this study. In our study, we will focus on four main challenges of the BTC price prediction with CNN:

- Decision of graph type to use
- Preparation of predictor variable structure
- Adjustment of target variable
- Adding correct expert judgments to the model result in order to strengthen the model performance

2.1 Decision of Graph Type to Use

In this study, it was assumed that the visual of historical data for a specific period length can be used to predict the direction of next price movement. In addition, some technical analysis metrics can be added to these images to be helpful for prediction.

The candlestick graph was employed for visualization of historical prices and the below figure briefly presents and explains the structure of this graph. As can be seen below, candlestick charts display the high, low, opening, and closing prices of a security for a specific period.

Historically, Japanese rice merchants and dealers used candlesticks to observe market prices and daily movement. Typically, traders use candlesticks to track patterns and indications on charts for any liquid financial instrument, including stocks, foreign exchange, cryptocurrencies, and futures. Additionally, it is adaptable and appropriate for any trading times, including daily, hourly, or minute-long cycles.

The aim of this study is to automate the prediction process by eliminating the difficulty of analyzing the candlestick chart with traditional methods and making a buy and sell decision on it.

Figure 4: Candlestick graph structure

2.2 Preparation of Predictor Variable Structure

Since the aim is creating an image processing model, the predictor variable needs to be an image. In this study, images should reflect all required information to the model because the CNN model will learn everything from them. For example, it should be determined what the data frequency will be, the colors should be separated according to the decrease and increase in the images, there should be no unnecessary visuals or texts in the images, it should be clarified how many data point to go backwards in each image, what additional information can be found in the images other than the observation point (such as technical analysis indicators) etc. All these decisions will be discussed and made in the next section.

2.3 Adjustment of Target Variable

Although the aim of this study is to predict the next price movement, it is not easy to make a strong prediction directly after a data point due to the volatility of the data in this market and possible delays in the trend. That is, it would be more logical to predict the movement in the near future rather than knowing directly what the movement will be at the next data point. For this reason, it may be necessary to make a couple of adjustments on the target variable to be used while modeling in order to increase and stabilize the predictive power of the model. All these decisions will be discussed and made in the next section.

2.4 Final Result Expert Judgment

In fact, expert judgments are very important and often used by traders during their trading since success in this field is very closely related to experience. It is also not very easy to obtain a truly successful model alone in such image processing studies. In these cases, it may be necessary to add expert judgment that will strengthen the predictions of the model and increase the model performance. These would be for example just use the 1 or 0 prediction for very high or low probabilities respectively or avoid excessive losses above risk appetite by using stop loss etc. All these decisions will be discussed and made after model prediction in this study.

CHAPTER III

DATA ARCHITECTURE

Blockchain data is freely accessible to anyone, but using it for image processing is not a method that has been extensively studied in the literature. The Binance [1] API, which provides access to raw data, is what we've chosen to use in this study. To create a dataset suitable for predictive modeling, the raw data given by the API must be meaningfully altered.

3.1 Architecture

The final dataset is composed of one data source. We only utilize Python [14] programming language to fetch data. The Python library of Binance.Client was used to collect BTC time series datasets automatically from the Binance Exchange Company to the Python by using apithe _I a a key and secret_key.

3.2 Data Source

The only data source of the study is Binance [1]. Changpeng Zhao established Binance in 2017, which is one of the biggest cryptocurrency exchanges in the world. Due to its user-friendly platform, wide selection of trading pairs, minimal trading costs, and strong liquidity, Binance has grown in popularity. By establishing numerous programs to encourage the adoption of cryptocurrencies and blockchain technology, the exchange has also established itself as a prominent participant in the cryptocurrency market.

Users of Binance can trade a variety of cryptocurrencies, such as Bitcoin, Ethereum, Tether, BNB, XRP and many more, as well as a number of trading pairs with fiat currencies, including USD, EUR, and others. For more advanced functionality, the exchange also provides futures trading, staking, and other advanced capabilities.

Binance provides a number of additional services and products in addition to trading, including as a decentralized exchange (DEX), a cryptocurrency wallet, and Binance Smart Chain, a blockchain platform that enables developers to create decentralized applications. (DApps).

3.3 Fetching Data

Any apps or services that need access to real-time market data and trading capability on the Binance exchange can use the Binance [1] API as a main data source. It is possible to access a number of services through the Binance API that give real-time data, including order book depth, ticker prices, trading history etc. It is also possible to use programming to carry out trades, manage orders, and do other tasks. Additionally, Binance also makes historical data available to access historical market data for analysis. Candlestick chart data, trade data, and order book snapshots are just a few examples of the historical data. For anyone wishing to create software or services that involve trading or market analysis on the blockchain, the Binance API is a comprehensive data source.

Binance API was utilized to fetch the data. For this aim, we employed the Python library of Binance.Client to take BTC time series datasets directly from the Binance to the Python by using api_key and secret_key.

3.4 Transforming Data

Basically, the dataset provides date, open time, close time, quote asset volume, number of trades, taker by base asset volume, taker by quote asset volume, high, low, open, and close prices with volumes for a given frequency, which is set to 1-hour in this project. From the source, daily close price information with respect to Tether (USDT) values are collected (which is named as BTCUSDT). Since Tether is the stable coin that mirrors the value of the U.S. dollar, Tether exchange values are used as the timeseries.

A transformation phase is necessary to prepare the data for modeling because the data pulled from source is in a raw format. As a result below listed transformation stages are applied:

- Split the whole dataset as train and test
- Create predictor variable (x label)
- Create prediction target variable (y label)

3.4.1 Train and Test Samples

The data period is traditionally divided into training and test samples in order to assess the model performance with out of time dataset. Theoretically, train and test samples must be independent, otherwise, the quality metrics will be unduly overestimated. In order to ensure a robust model and test the model performance with the out of sample dataset, the whole dataset mentioned above was split as train and test dataset, and division was done by time information with ratios of 70% and 30% respectively. It should be noted that the most recent period was used for development in order to obtain a model that could reflect more recent periods. As a result, the train dataset has the period between 04-03-2018 and 17-05-2019 (10.500 number of observations) while the test dataset covers the time horizon between 27-08-2017 and 04-03-2018 (4.500 number of observations).

3.4.2 Predictor variables (x labels)

Since it is aimed to use candlestick charts for image processing, the variables used in the candlestick principle are considered as explanatory variables in this study. These fields can be listed as high, low, open, and close prices for a given frequency.

Additionally, some strong and remarkable technical analysis indicators are taken into account as explanatory variables in this study since traders mostly make their decisions with technical analysis indicators in this type of technical analysis evaluations and this information was wanted to add to the model. These indicators are SMA13, SMA21, SMA34, SMA55, Bollinger Bands, RSI.

Since the model used in this study is CNN, the predictor variable is matrices of digital images corresponding to the quotes charts on certain time intervals - data slices.

The Japanese candlestick charts are used to create these images which is the one of the most common chart types that accounts for all price data (open price, close price, high and low) in this area. The decrease and increase information of each observation is given with the color code in the pictures as well as the numerical values (high, low, open, and close prices for a given frequency). In this study, the green color of the candle indicates that the observation for that frequency is completed with an increased movement, while the red color means that the observation is completed with a decreased movement. It should be noted that the choice of colors, however, is not critical, as long as they are contrasting.

From the point of view of digitization, an image is a collection of pixels arranged in a given order. Each pixel is characterized by its intensity, measured in the range from 0 to 255 (from white to black for monochrome). To digitize color images the RGB scheme is often used. In this case, each pixel is described by a three-dimensional vector containing the color intensity in three channels - red, green and blue. All other colors are obtained by combinations of color intensities in different channels.

The intensity value in each channel that describes a pixel corresponds to one variable, therefore the total number of input variables depends on both the number of channels that describe a pixel and the picture resolution (dpi, or digits per inch). The speed of calculations is increased but some information may be lost due to the drop in resolution and number of channels (and, consequently, a decrease in the dimension of the feature space).

For image processing, one of the most important decisions, it is necessary to decide how many candles should be in each picture as an input. Furthermore this decision had to be made with expert judgment since there is no definite approach on this issue. For instance, authors in [15] used the input of the models consisting of the index values from the previous 14 days and the output is the next-day value of the index. Therefore, traditional trading methods and approaches were examined again in this study to decide that. Usually, traditional traders use Fibonacci numbers, which have an important place in technical analysis for such frequencies. Small Fibonacci numbers such as 3, 5, 8, 13 are used if rapid market reactions are expected, while larger Fibonacci numbers such as 21, 34, 55 are taken into account for late market reactions. In this study, different numbers were tried for this purpose and it was decided to use the number 21, which gives better results and is considered relatively average. As a result, below steps are followed briefly for input image preparation for the model;

- Each picture created for model prediction and testing, there are 21 candles with a frequency of 1 hour as observation information.
- In the pictures, each candle is given in green and red colors, respectively, according to the increase, decrease of closing status.
- Some technical analysis indicators (SMA13, SMA21, SMA34, SMA55, Bollinger Bands, RSI) have been added to the pictures in order to increase the predictive power.
- Model estimation with these pictures is made in the form of an increase or decrease for the closing price of the next observation(s).

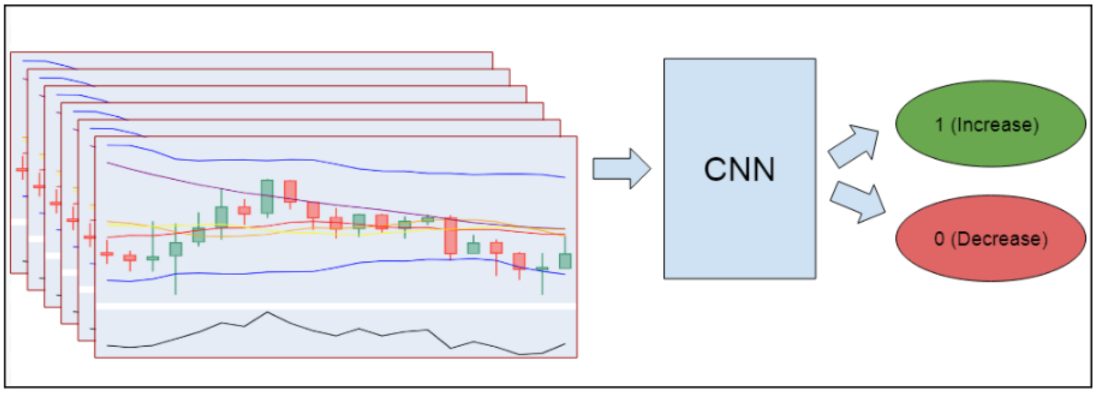

Figure 5: CNN model building structure

3.4.3 Prediction target (y label)

Prediction target of this model was decided to be binary as 1 or 0. Firstly a binary variable is created for this aim; this variable takes the value 1 if the closing price of the next observation is greater than the closing price of the current observation, not 0. Secondly, this variable is adjusted to capture and reflect the lagging movements in the target variable and to create a more stable target variable via expert judgment. As a result, the final target variable was created; the mean of the next 5 binary variables is calculated, and if this value is greater than 0.5 (average), the final target variable takes the value 1 if not 0.

It should be noted that, since CNN is a sensitive model in this respect, the price and date information on the axes have been removed in order not to mislead the pictures with any writing. Therefore, final model parameters can be summarized as:

- **Dataset – frequency:** BTC data – 1 hour
- **Train period:** 04/03/2018 – 17/05/2019
- **Number of observations in train set:** 10,500
- **Testing period:** 27/08/2017 - 04/03/2018
- **Number of observations in testing set:** 4,500

- **Percentage of test set:** 30%
- **Number of candlesticks in each picture:** 21
- **# of obs. whose average is calculated to determine the target variable:** 5
- **Technical indicators used in pictures:** SMA13, SMA21, SMA34, SMA55, Bollinger Bands, RSI

CHAPTER IV

DESCRIPTIVE ANALYSIS

For this analysis, the period between 27-08-2017 and 17-05-2019 was collected for BTC overall. As a result the dataset covers 15.000 number of observations. Below figure depicts the historical prices of the BTC.

Figure 6: Whole (Train + Test) dataset over time

Obviously, high fluctuations are observed in some periods throughout the horizon and the dataset follows a non-stationary pattern. Briefly, it is seen that the price showed a downward trend from January 2018 to April 2019, while it smoothly entered an increasing trend after this date. It should be noted that no scaling was needed and applied due to these fluctuations for this study in order to give the model the opportunity to cover this issue within the algorithm. In the sub-sections below, we will analyze target classification of datasets and descriptive statistics of explanatory variables in detail.

4.1 Target classification

Machine learning classification problems need us to predict a discrete target after receiving some input (independent variables). The probability that the distribution of discrete values will deviate greatly is very high. The algorithms have a tendency to get biased towards the majority values present and perform poorly on the minority values as a result of this discrepancy in each class. This is also known as class imbalance problem i.e., that one class occurs far more frequently than the other classes than are present. In other words, the target has a bias or skewness in favor of the dominant class. In a binary classification, for instance the majority target class has 1,000 rows and the minority target class has just 10 rows. In that situation, the ratio is 100:1, meaning that there is only one minority class present for every 100 members of the majority class. This is a general issue in literature especially for rare cases like fraud detection or medical diagnosis.

This variation in class frequencies has an impact on the model's overall predictability. It is not extremely difficult to achieve good accuracy on these questions, although success is not usually measured in terms of a high score. Therefore, it is not desired to have an imbalanced dataset for modeling. There can be different approaches to consider balancing data such as removing some data from rows of data from the classes with the majority of the data, copying minority labels from data rows or synthesizing new data depending on the implications of existing data. Below table presents target variable proportions over train and test datasets that were prepared for this study. Fortunately, no such problem was observed in both train and test datasets and the distributions seem to be almost balanced in both data (around %50). Additionally, there is no imbalance between train and test datasets' distributions and they are align as well.

Table 1: Target distribution over train dataset

$Class$	$\#ofobservation$	$\%ofobservation$
$Increase(1)$	$5,389$	51.35%
$Decrease(0)$	$5,106$	48.65%
$Total$	$10,495$	100%

Table 2: Target distribution over test dataset

$Class$	$\#ofobservation$	$\%ofobservation$
$Increase(1)$	$2,364$	52.53%
$Decrease(0)$	$2,136$	47.47%
$Total$	$4,500$	100%

4.2 Descriptive statistics of explanatory variables

As mentioned before, there are four explanatory variables in this model which are listed as open, high, low, and close values of each observation point. Below table summarizes descriptive statistics of these variables for train dataset.

	open	high	low	close
count	10495.00	10495.00	10495.00	10495.00
mean	6042.98	6071.45	6012.03	6042.59
std	1809.90	1822.23	1795.84	1809.21
min	3172.62	3184.75	3156.26	3172.05
25%	3996.32	4009.32	3985.84	3996.33
50%	6403.54	6428.59	6375.30	6403.00
75%	7225.92	7270.84	7189.64	7225.33
max	11666.00	11710.00	11640.00	11669.08

Figure 7: Descriptive statistics of predictor variables in train dataset

On the other hand, descriptive statistics are given in the below table for the test dataset. Descriptive statistics of train and test datasets seem discordant due to yearly fluctuation observed before.

	open	high	low	close
count	4500.00	4500.00	4500.00	4500.00
mean	8945.63	9049.84	8833.91	8947.06
std	4101.87	4164.16	4028.14	4101.42
min	2870.90	2950.00	2817.00	2919.00
25%	5321.76	5376.50	5284.54	5323.22
50%	8270.56	8368.97	8210.75	8275.82
75%	11535.83	11644.75	11417.71	11540.47
max	19709.50	19798.68	19552.00	19709.50

Figure 8: Descriptive statistics of predictor variables in test dataset

Finally, overall yearly distribution of explanatory variables are checked in the figure below. It is obviously seen that there is a decreasing trend for variables' descriptive statistics parallel to the overall distribution.

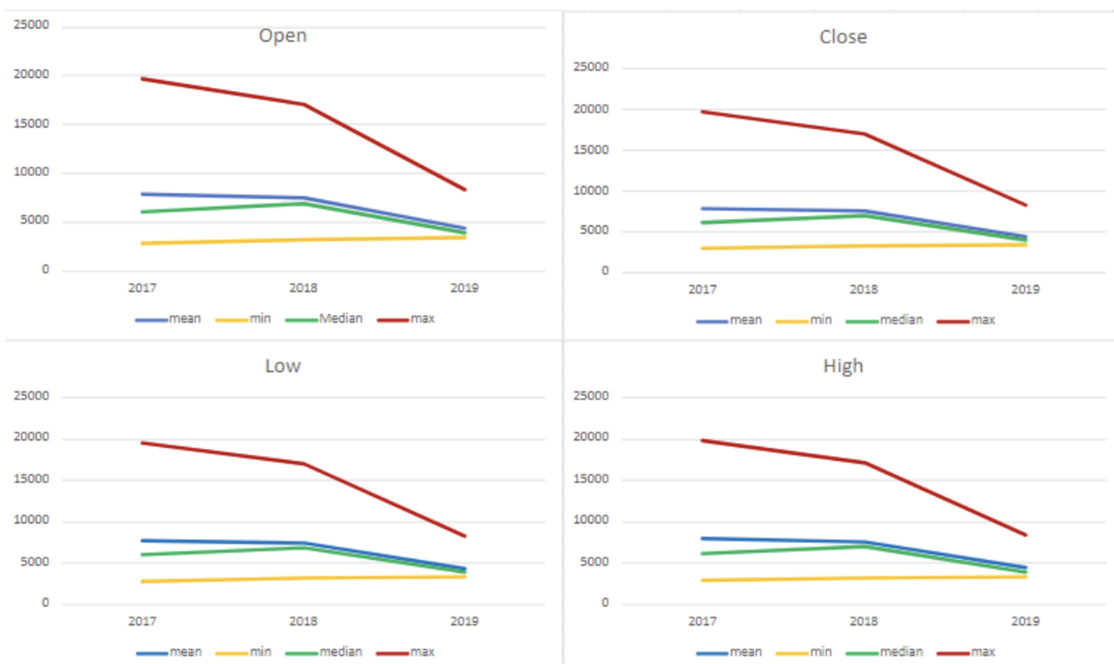

Figure 9: Descriptive statistics of predictor variables over years

CHAPTER V

CNN MODEL FOR PRICE MOVEMENT PREDICTION

As mentioned earlier, BTC was chosen for modeling in this study. The reason behind this is that BTC's popularity, significant market capitalization, and track record make it a subject in this project. Additionally, if the proposed tool created for this study has adequate performance and structure on BTC, it will be simple to adapt it to other cryptocurrency kinds in the future.

This section starts with explanation of image pre-processing for CNN in order to explain employed image pre-processing methods and its selected hyper-parameters. Afterwards, it continues with building CNN model to summarize modeling steps and selected hyper-parameters for this study.

5.1 Image Pre-Processing for CNN Model

A Python interface for artificial neural networks is provided by the open-source software package known as Keras. The TensorFlow library interface is provided by Keras. As well as a variety of tools to make dealing with picture and text data easier, Keras includes multiple implementations of widely used neural-network building blocks like layers, objectives, activation functions, and optimizers. This helps to simplify the coding required to create deep neural networks. Convolutional and recurrent neural networks are supported by Keras in addition to traditional neural networks. To pre-process the data for CNN model in this study, the library of keras.preprocessing.image was employed [16]. Basically, this library utilizes image preprocessing and augmentation. Therefore, below sub-libraries were used for image pre-processing in this study.

ImageDataGenerator:

This library essentially creates batches of real-time data augmented tensor image data. It offers a variety of augmentation methods, including standardization, rotation, shifts, flips, brightness changes, and many others. The benefit of this approach is that it makes an attempt to model by taking into consideration the various permutations of images that you desire while each image generates an input to the model. This not only adds richness to the model but also enhances model performance. The fact that ImageDataGenerator uses less RAM is another benefit. This is true since if we didn't utilize this technique, we'd load every image at once. However, by loading the photographs in batches, we are able to save a significant amount of memory.

Below list of hyper-parameters were adjusted and taken into account during this process:

Table 3: Hyper-parameters of ImageDataGenerator

$Hyper-parameter$	$Value$
$Rescale$	1./255
$Shear\_range$	0.2
$Zoom\_range$	0.2
$Horizontal\_flip$	$True$

Rescale: Rescaling factor. After doing all other transformations, we multiply the data by the value provided if it's not None or 0, in which case no rescaling is performed. This setting helps in bringing the image's color scale into balance. 255 was used to normalize because there are 255 possible color tones for images.

Shear_range: Shear Intensity (Shear angle in counter-clockwise direction in degrees).

Zoom_range: Range for random zoom. Any value smaller than 1 will zoom in on the image whereas any value greater than 1 will zoom out on the image.

Horizontal_flip: Randomly flip inputs horizontally. It is generally employed to improve model performance and avoid overfit.

Flow_from_directory:

As the neural network model learns from the training data, you can read images straight from the directory and edit them and this method helps in this part. During this procedure, the following list of hyper-parameters was modified and taken into consideration:

Table 4: Hyper-parameters of Flow_from_directory

$Hyper-parameter$	$Value$
$Target\_size$	$(64, 64)$
$Batch\_size$	32
$class\_mode$	$'binary'$

Target_size: Tuple of integers (height, width). The size that will be used to resize all images that are discovered.

Batch_size: Size of the batches of data and it controls accuracy of the estimate of the error gradient. Default value is 32 and it was used for this study.

Class_mode: One of "categorical", "binary", "sparse", "input", or None. It determines the type of label arrays that are returned. Binary was set for this parameter since the target variable is binary for this study as 0 or 1.

5.2 Building CNN Model

A free and open-source software library for artificial intelligence and machine learning is called TensorFlow [17]. It can be used for a variety of applications, but focuses particularly on deep neural network training and inference. The library of tensorflow was used in order to build the CNN model in this study. Basically, this library ensures an end-to-end AI process in python for users. Below two figures briefly summarized the steps followed in this part and general architecture.

tf.keras.layers.Conv2D: 2D convolution layer (e.g. spatial convolution over images). Below list of hyper-parameters were adjusted and taken into account during this process:

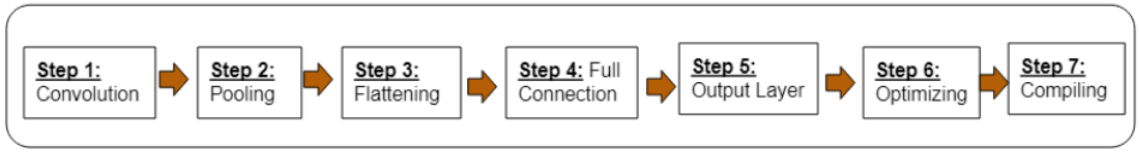

Figure 10: CNN model building steps

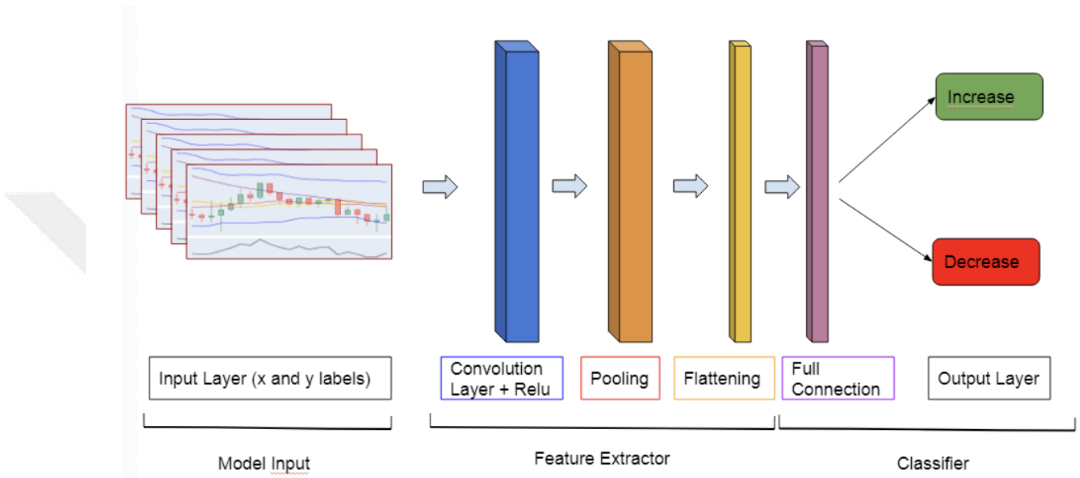

Figure 11: CNN model architecture

Filters: nteger, the dimensionality of the output space (i.e. the number of output filters in the convolution). 32 was set in this study.

Kernel_size: An integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window. Can be a single integer to specify the same value for all spatial dimensions. 3 was set in this study.

Activation: Activation function to use. Relu was set in this study. Briefly, ReLU (rectified linear unit) activation function is an activation function defined as the positive part of its argument in the context of artificial neural networks. It is also commonly activation function to use in these kinds of exercises.

Input_shape: It needs to be provided when using this layer as the first layer in a model, input_shape=(64, 64, 3) was set in this study for 64x64 RGB pictures in the dataset.

tf.keras.layers.MaxPooling2D: Max pooling operation for 2D spatial data. Below list of hyper-parameters were adjusted and taken into account during this process:

Pool_size: Window size over which to take the maximum. (2,2) will take the max value over a 2x2 pooling window. If there is just one integer is provided, the window length for both dimensions will be the same. 2 was set in this study.

Strides: Strides values. Indicates the amount by which the pooling window shifts for each pooling step. 2 was set in this study.

tf.keras.layers.Flatten: Flattens the input.

tf.keras.layers.Dense: Just your regular densely-connected NN layer. Below list of hyper-parameters were adjusted and taken into account during this process:

Units: Positive integer, dimensionality of the output space. 128 was set for full connection with expert judgment while 1 was set for output layer due to binary target in this study.

Activation: Activation function to use. Relu was set for full connection while sigmoid was set for output layer due to there being a binary target in this study. Briefly, ReLU (rectified linear unit) activation function is an activation function defined as the positive part of its argument in the context of artificial neural networks. On the other hand sigmoid function is a bounded, differentiable and monotonic function that has an S-shape and is commonly used for probability distribution between 0 and 1.

tf.keras.optimizers.Adam: Optimizer that implements the Adam algorithm. Below list of hyper-parameters were adjusted and taken into account during this process:

Optimizer: Adam was set in this study. A stochastic gradient descent technique called Adam optimization is based on adaptive estimate of first- and second-order moments.

Learning_rate: The learning rate controls the step size at each iteration of an optimization algorithm as it advances toward a minimum of a loss function. 0.001

was set in this study.

Compile: When the model is created, the model is needed to config. Below list of hyper-parameters were adjusted and taken into account during this process:

Loss: Binary_crossentropy was set in this study since there is a binary classification. Briefly, the actual class output, which can only be either 0 or 1 in this case, is compared to each of the predicted probabilities using binary cross entropy. The score that penalizes the probabilities based on how far they are from the predicted value is then calculated.

Metrics: Accuracy was set in this study since it is one of the most popular metrics to measure the model performance.

Fit: Once the model is created, the model is needed to fit. Below list of hyper-parameters were adjusted and taken into account during this process:

Epochs: A single epoch occurs when the neural network processes a complete dataset once, both forward and backward. 20 was set in this study.

Predict: Once the model is created, the model is needed to predict on a test dataset with this method.

It should also be noted that, given in all of the above hyper-parameters tried to be optimized by trying different values during this study and decided with expert judgment.

CHAPTER VI

AREA ESTIMATION RESULTS

In order to measure the performance of binary classification methods, confusion matrix generally created and employed. Therefore, model performance is measured with metrics calculated over this matrix. Below list summarizes the components of this matrix:

- **True Positive (TP);** means correctly predicted event values.
- **False Positive (FP);** means incorrectly predicted event values.
- **True Negative (TN);** means correctly predicted no-event values.
- **False Negative (FN);** means incorrectly predicted no-event values.

As a result, below formula gives accuracy of the model;

Accuracy = (TP + TN)/(TP + TN + FP + FN)

In this study, 20 epochs were tried to build the model and the results show that training has 63% accuracy while validation has 52% accuracy in the final epoch for next price movement prediction. The accuracy is the ratio of the number of correct predictions to the total number of input samples. Accuracy is the most common evaluation metric for classification models because of its simplicity and interpretation.

Figure 12 illustrates model accuracy and loss distribution over epochs for both train and test datasets. It is seen that model performance on train dataset increases over epochs while unfortunately this trend cannot be followed exactly for model performance on test dataset.

On the other hand, model prediction results were checked by pcut function of python as given in below tables in order to see model stability over datasets. They

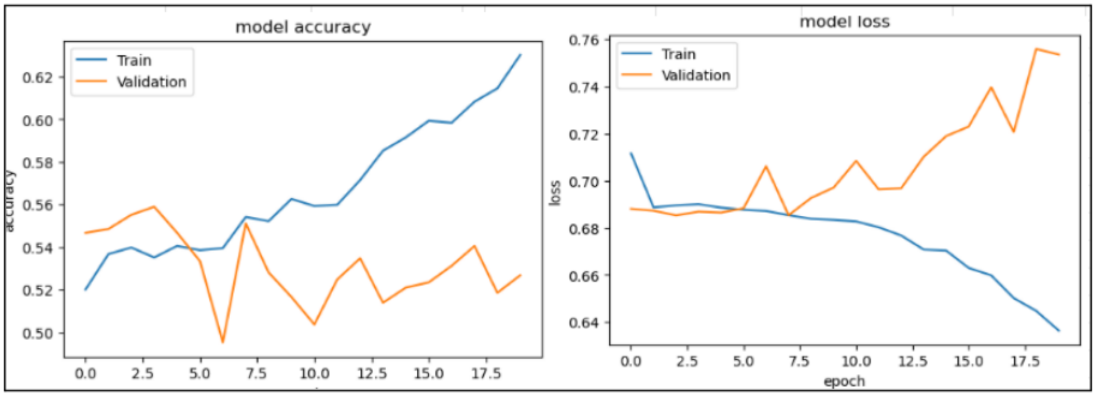

Figure 12: Model accuracy and loss distribution over epochs for both train and test datasets

show the number of observations in each prediction area and it seems they are mostly aligned for both datasets in line with both prediction areas and number of observations in each area. It is obviously seen that there is a concentration in the prediction areas around the 20 -80 bands as expected to reflect normal distribution shape.

Table 5: The number of observations in each prediction area for test dataset

$Pcut(predictionarea)$	$\#ofobservation$	$\%ofobservation$
$(0.00719, 0.203]$	187	4.16%
$(0.203, 0.398]$	873	19.40%
$(0.398, 0.594]$	1722	38.27%
$(0.594, 0.789]$	1431	31.80%
$(0.789, 0.984]$	287	6.38%

The model performance values obtained above unfortunately show that the performance of the model is not very good, with a performance of about 50% accuracy, such as flipping a coin. Because, this kind of image processing model is not very easy to use for this purpose with high performance. Using the model directly in this way will not provide the desired performance in terms of profit. For this reason, it has been tried to increase the model output performance and profit by adding expert judgment on the model estimations other than those applied when setting up the dataset or model. This expert judgment added to the model output can be explained

Table 6: The number of observations in each prediction area for train dataset

$Pcut(predictionarea)$	$\#ofobservation$	$\%ofobservation$
$(0.0129, 0.206]$	343	3.27%
$(0.206, 0.398]$	1777	16.93%
$(0.398, 0.59]$	4883	46.53%
$(0.59, 0.782]$	3089	29.43%
$(0.782, 0.975]$	403	3.84%

as putting a stop loss in order to minimize the loss for each buy or sell transaction made with the model output.

Afterwards, the performance of the transactions made with the model was also evaluated by comparison with the holding method with gold and BTC which means to buy the relevant product at the beginning of the relevant data period and to sell it at the end of the relevant data period. For this comparison the Return on Investment (ROI) metric was used. Since;

- ROI is a popular profitability metric used to evaluate how well an investment has performed.
- ROI is expressed as a percentage and is calculated by dividing an investment's net profit (or loss) by its initial cost or outlay.
- ROI can be used to make apples-to-apples comparisons and rank investments in different projects or assets.

For this comparison, initial investment of $ 10,000 made as an example for model and other products to buy-hold. In addition, cost/commission was taken into account for each buy or sell transaction made with the model output. ROI results for each alternative are given below and it is presented that our model seriously beats the other buy-hold alternatives for test and train dataset.

Table 7: ROI comparison between alternatives

$Comparison$	ROI	ROI
$Alternatives$	$Train$	$Test$
$Model$	588%	1560%
$HoldGold$	-2.8%	0.2%
$HoldBTC$	-32%	162%

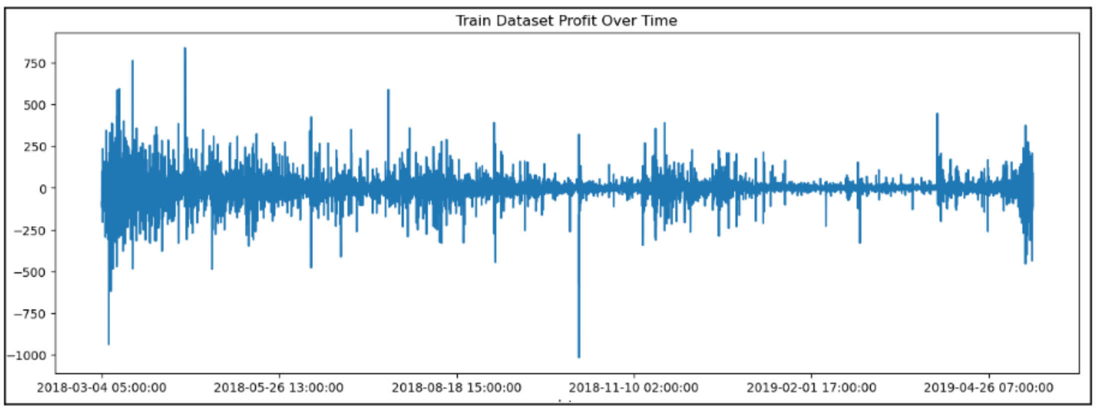

Figure 13: Train dataset profit overtime

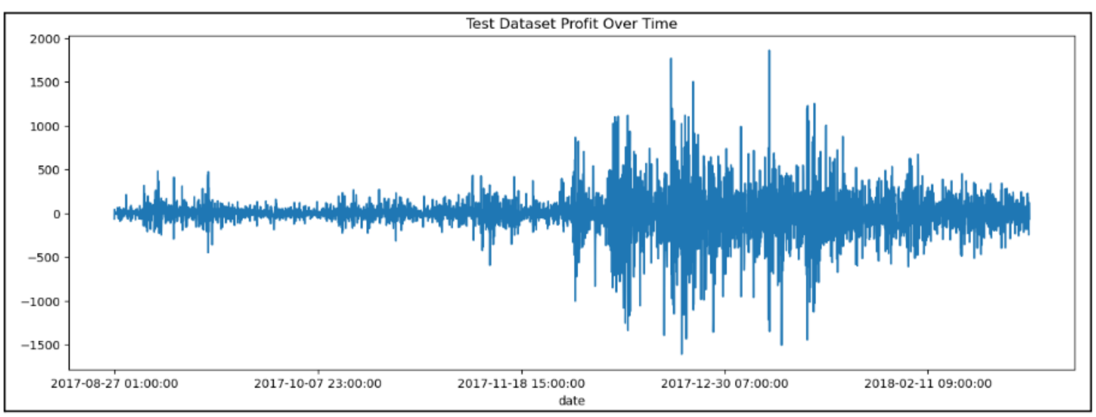

Figure 14: Test dataset profit overtime

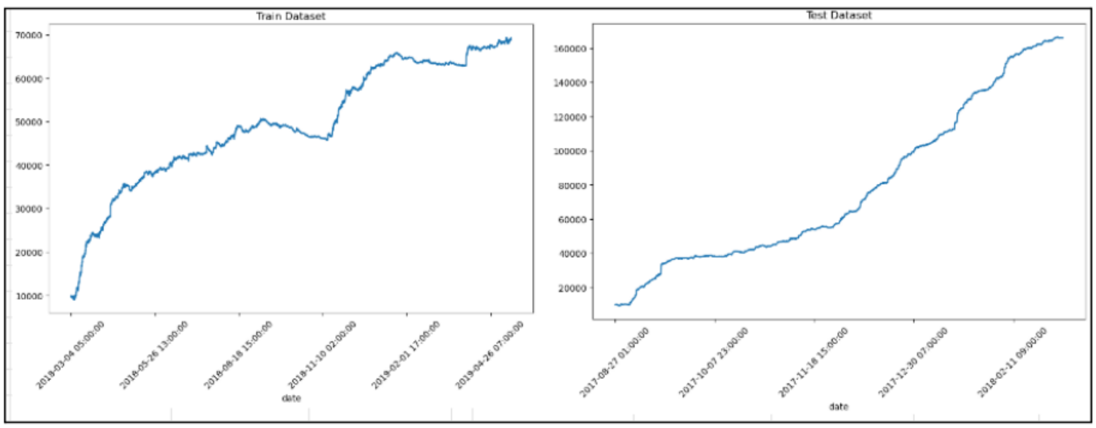

Figure 15: Cumulative balance for train and test datasets respectively

CHAPTER VII

CONCLUSIONS

In this paper, an image processing architecture for forecasting the future price movement of cryptocurrency assets and using it to successfully predict the next-hour direction of the price are proposed. Concretely, it was predicted the next hour value of BTC based on the information from the previous 21 hours. For this purpose, CNN model with expert judgements was employed by using BTC data. As a result, the data for each hour in the input layer is considered in the context of its preceding and following hours. Also, the usage of the technical analysis indicators leads to more informative features in the model. The results of the model proposed against benchmark buy-hold ways demonstrate the profitability of the proposed model via ROI metric. Additionally, the model achieves an accuracy rate of 63% and 52% respectively for training and test dataset.

For future work, the proposed deep learning architecture can be applied to other cryptocurrencies with different input structures or frequencies. Secondly, some ensemble model architectures could be tried. Thirdly, the final model output could be supported with different established policy rules; for example, taking into account only the ones with a high probability of increase or decrease in line with some limits, or making a decision to invest a higher amount in these cases. Fourthly, different parameter optimizations could be implemented for better performance.

REFERENCES

[1] Binance, "https://www.binance.com/tr," (accessed March 5, 2022).

[2] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," in *Inderscience*, vol. 14, p. 352–375, 2018.

[3] F. Fang, C. Ventre, M. Basios, L. Kanthan, D. Martinez-Rego, F. Wu, and L. Li, "Cryptocurrency trading: a comprehensive survey," in *Financ Innov 8*, vol. 13, pp. 5–9, 2022.

[4] E. Ladislav, "Bitcoin meets google trends and wikipedia: Quantifying the relationship between phenomena of the internet era," in *Sci Rep 3*, vol. 3415, pp. 1–7, 2013.

[5] A. Greaves and B. Au, "Using the bitcoin transaction graph to predict the price of bitcoin," in *Semantic Scholar*, vol. unknown, pp. 1–8, 2015.

[6] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Springer*, vol. 7859, pp. 1–24, 2013.

[7] I. Madan, S. Saluja, and A. Zhao, "Automated bitcoin trading via machine learning algorithms," in *Semantic Scholar*, vol. unknown, pp. 1–5, 2012.

[8] S. M. Raju and A. M. Tarif, "Real-time prediction of bitcoin price using machine learning techniques and public sentiment analysis," in *Arxiv*, vol. unknown, pp. 1–14, 2020.

[9] Y. Hua, "Bitcoin price prediction using arima and lstm," in *E3S Web Conf.*, vol. 218, pp. 1–5, 2020.

[10] C. H. Wu, C. C. Lu, Y. F. Ma, and R. S. Lu, "A new forecasting framework for bitcoin price with lstm," in *IEEE*, vol. 2018, pp. 168–175, 2018.

[11] I. A. Hashish, F. Forni, G. Andreotti, T. Facchinetti, and S. Darjani, "A hybrid model for bitcoin prices prediction using hidden markov models and optimized lstm networks," in *IEEE Xplore, 4th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 2019, pp. 721–728, 2019.

[12] S. Cavalli and M. Amoretti, "Cnn-based multivariate data analysis for bitcoin trend prediction," in *Applied Soft Computing Journal*, vol. 101, pp. 1–12, 2021.

[13] S. Ji, J. Kim, and H. Im, "A comparative study of bitcoin price prediction using deep learning," in *MDPI*, vol. 7, p. 898, 2019.

[14] Python, "https://www.python.org/," (accessed March 15, 2022).

[15] F. Kamalov, L. Smail, and I. Gurrib, "tock price forecast with deep learning," in *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pp. 1098–1102, 2020.

[16] Keras, "https://keras.io/," (accessed April 3, 2022).

[17] TensorFlow, "https://www.tensorflow.org/," (accessed May 12, 2022).

VITA

Aysel Çetin completed elementary school education in Göktürk İlköğretim Okulu, in June 2006, in İstanbul. The following September she started high school in Ali Rıza Özderici Lisesi, in İstanbul. Afterwards, she entered Yıldız Technical University, in September 2009, in İstanbul and graduated as a high honor student. The department was statistics. Finally, she started Özyeğin University in February 2021 to receive a degree Master of Science of Data Science.