UNIVERSIDADE DO VALE DO RIO DOS SINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

EDUARDO REMOR DE SOUZA

CRYPTOCURRENCY TRADING BOT ASSISTED BY ARTIFICIAL INTELLIGENCE

Novo Hamburgo
2022

Eduardo Remor de Souza

CRYPTOCURRENCY TRADING BOT ASSISTED BY ARTIFICIAL INTELLIGENCE

Artigo apresentado como requisito parcial para obtenção do título de Bacharel em Sistemas da Informação, pelo Curso de Sistemas da Informação da Universidade do Vale do Rio dos Sinos (UNISINOS)

: Prof. Cristiano Andre da Costa

Novo Hamburgo
2022

CRYPTOCURRENCY TRADING BOT ASSISTED BY ARTIFICIAL INTELLIGENCE

Eduardo Remor de Souza[1] Cristiano Andre da Costa[2]

Abstract:

Cryptocurrencies are getting more popular day after day as an alternative investment option. With this hype, multiple exchanges are available and it is easy to set up an account and start trading in the crypto market. However, trading itself is not an easy task and depends on technical knowledge and emotional control if a person wants to be a profitable trader over time. Most people do not take time to study and get ready for the market and, therefore, end up with losses in a short period. An automated algorithm - a trading bot - following a good trading strategy could take the technical and emotional requirements aside and potentially be profitable, without needing expertise from the user. The market is dynamic and has its different moments, unlike a trading bot, which follows a fixed strategy and is tied to the market's movements. Artificial intelligence using a convolutional neural network (CNN) architecture could assist the trading bot by analyzing the macro scenario in a set of cryptocurrencies and indicating which ones are in a suitable moment for the trading strategy. The project's evaluation was done by comparing the financial returns of a trading bot set up with a strategy based on Keltner Channels and RSI with the trading bot assisted by a CNN-based model. The results were interesting as the CNN model did not perform well enough to improve the result from the trading bot, but the labeling method proved to be efficient. The trading bot without assistance had an average return of 14.43%. With AI's assistance, the average returns dropped to 13.28%. Considering the benchmark AI labels, the average return increased to 24.58%.

Keywords: Cryptocurrencies. Trading. Artificial Intelligence. Convolutional Neural Networks.

1 INTRODUCTION

The cryptocurrency market is exponentially growing over the world. 13% of Americans traded cryptocurrencies in 2020 according to a survey conducted by University of Chicago (2021). In Brazil, Hashdex (2022) reports a growth of 1266% in the number of investors in cryptocurrency funds over 2021. As this is a new type of asset and it is highly volatile, the risk of trading cryptocurrencies is high. As it is common knowledge and had been proved by studies (BARBER et al., 2017), most people that do short-term trades lose money and quit trading in a short period of time. This does not happen only due to the lack of technical knowledge, but also because of the emotions that affect the investors' decision-making process.

With that in mind, it is possible to think about different ways to resolve these issues to, possibly, have profit in short-term trades in cryptocurrencies. Using an automated algorithm with a previously defined trading strategy, it would be possible to not only cover the lack of technical knowledge but also to eliminate emotions in the decision-making process. This would

[1]Graduando em de Sistemas de Informação pela Unisinos. Email: eduardo.remors@gmail.com
[2]Mini-currículo do orientador. Email: cac@unisinos.br

happen as there would not be any human interference while the algorithm is running.

There are studies with different automated trading strategies solutions, as developed by Souza (2019). There are also approaches that use machine learning to know when to buy, sell or hold an asset, such as presented in *Stock Trading Classifier with Multichannel Convolutional Neural Network* (NASCIMENTO; COSTA; BIANCHI, 2020). However, in papers focused on short-term trades, thousands of trades happen in a short period of time, which makes the bot not profitable due to the high taxes caused by over-trading. This is a gap that can be explored by applying a trading management strategy to the trading bot that is not only focused on the monetary value but also on the number of trades. Another gap is related to papers using machine learning: none of the papers use AI to assist the trading bot by telling it when the macro scenario is favorable to the trading strategy rather than executing the trades itself. An AI algorithm capable of identifying these moments in the macro scenario would be helpful not only to trading bots but also to traders that want to know when the market is better to trade using their strategy.

This paper aims to create an automated robot capable of executing a trading strategy and use the assistance of artificial intelligence (AI) to learn and decide which cryptocurrencies are in better conditions to trade at a given moment. The specific objectives were split into three points:

- Create a trading bot to follow a strategy using Keltner Channels and RSI;
- Use a Convolutional Neural Network AI architecture to learn the best moments to trade in a macro scenario;
- Compare the trading bot results with the bot with AI assistance, with the assistance of labels generated by the labeling algorithm and a buy-and-hold strategy.

This article presented the main concepts to understand the research in the next section. In the Related Work section, papers with related themes were described. In the Methods section, the development and details of each module within the project were described. The fifth section showed the results and a discussion about them. Lastly, section 6 concluded the research and debated future improvement.

2 BACKGROUND

In this section, the theoretical basis of the research was presented. The definition of principles and terms within trading, cryptocurrencies, and artificial intelligence areas were explained.

Trading, candlesticks, graphics, and technical indicators were addressed in the first subsections. Next, cryptocurrencies and what are they used for today were explained. The last sub-section described artificial intelligence, some of its concepts, and models, one of which was applied in practice for the research.

2.1 Trading

Trading is the act of buying or selling a given asset in a financial market (CAPITAL RESEARCH BRASIL, 2020). This asset can be a stock, future indexes, currency pairs, cryptocurrencies, and others. These buy and sell operations are done by a trader, who does it through a trading platform (CAPITAL RESEARCH BRASIL, 2020). There are retail traders, who are natural people, and institutional traders, who are the banks, financial institutions, and funds, responsible for most part of the negotiation volume of any market.

The simplest trading strategy is Buy and Hold. It consists in buying an asset and holding it for a long period of time (BEERS, 2020). Another simple and popular strategy is Dollar Cost Averaging (DCA). In DCA, the investor divides the total amount to be invested in equal parts that will be used to purchase an asset periodically, regardless of the current price (HAYES, 2022). This period can be weekly or monthly, for example. The goal is to reduce the impact of volatility on the overall purchase.

Trading can be automated and executed by trading bots. These are robots that can trade by themselves based on algorithms, without human interference. Also called algo trading, these automated tools can be configured according to parameters informed by its human users (YAPPI, 2020).

One type of trading bot is High-Frequency Trading (HFT). These are high-frequency algorithms that operate in financial markets, mainly by banks and financial institutions. According to Ismar (2021),

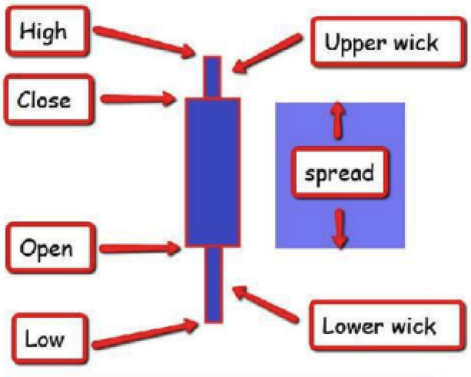
> High-Frequency Trading comes from a principle that reminds tape reading operations. However, the computational engineering behind these robots allows this to be done in very short time intervals. These intervals are within the microsecond order. This is only possible due to the high technology applied to the architecture of the robots.
> Robots of this type are more frequently used by institutional investors, such as banks and other financial institutions. This recourse is usually not available to small investors, mainly because the HFT strategy relies on high negotiation volume.

2.2 Candlesticks

Coulling (2013) explains that there are seven key elements that shape a candlestick: the open, high, low, and close, the upper and lower wicks, and the spread, as shown in Figure 1.

The open, close, high, and low points show the price at the moment within the candlestick's time frame. This time frame can be minutes, hours, days, weeks, or months. In a 5 minutes time frame, the open shows the price at time 0, the close shows it at time 5 minutes, and high and low show the highest and lowest prices within the 5 minutes. If the price change in the time frame was positive (open > close), it is a bullish candle, usually represented by green, blue or

Figure 1 – Candlestick



Source: Coulling, p.26 (2013)

white. If it was negative (open < close), it is a bearish candle, usually represented by red or black. Figure 2 shows a graph of 15 minutes candlesticks.

Figure 2 – 15 minutes candlestick graph

Source: TradingView (2021).

2.3 Technical Indicators

Nelogica (2019), the company responsible for the most popular trading platform in Brazil, explains that technical indicators are visual representations of formulas applied to the prices. They help traders to identify situations and read the market, in addition to the candlesticks. These are some of the main technical indicators:

2.3.1 Moving Average

The moving average is one of the most common technical indicators used in technical analysis. It is meant to identify trends in the price (FERNANDO, 2021). There are two main types of moving averages: the arithmetic (or simple) moving average and the exponential moving average.

- **Arithmetic Moving Average (SMA):** The arithmetic moving average is a simple average of the prices of each candle for a given period. This price can be set as the open, close, high, or low price. The most commonly used is the closing price. In an 80-period moving average, for example, the closing price of the latest 80 candles is summed and then divided by 80, so we have the current price of the moving average.

$$SMA = \frac{P1 + P2 + ... + Pn}{n}$$

  Where:
  P = Price at each candle within *n*
  n = number of time periods

- **Exponential Moving Average (EMA):** The exponential moving average gives more weight to recent prices, being more responsive to new information. Its calculation consists in applying a smoothing factor to the SMA for a given period.

$$EMA_d = EMA_{d-1} + \frac{2}{n+1} * (P_d - EMA_{d-1})$$

  Where:
  P = Price at each candle within *n*
  n = number of time periods
  d = current candle

2.3.2 Keltner Channels

Created by Chester Keltner, the Keltner Channel is a volatility-based indicator that can assist in determining the direction of a trend. It consists of three lines plotted in the graph: the middle one is an exponential moving average, and the upper and lower lines are calculated using the same EMA and a multiplier or the ATR of the asset, which is a volatility indicator. (MITCHEL, 2021).

The standard way to use Keltner Channels is with a 20-period EMA and an ATR multiplier of 2, but these values can be changed according to the trader's preference.

$$UB = EMA + ATRm$$

$$LB = SMA - ATRm$$

Where:
UB = Upper band
LB = Lower band
EMA = exponential moving average of a given period
ATR = Average True Range
m = ATR multiplier

2.3.3 Relative Strength Index (RSI)

The Relative Strength Index (RSI) is a momentum indicator. It measures the speed and magnitude of the recent price changes of an asset. The RSI is displayed as a line graph that ranges from 0 to 100. Usually, if the line is above 70, it means the asset is overbought - therefore it would be a good moment to sell - and if it is below 30, means the asset is oversold - a good moment to buy.

The calculation of the RSI happens in two parts:

$$FirstStepRSI = 100 - \frac{100}{1 + \frac{AverageGain}{AverageLoss}}$$

Where:
Average Gain = average percentage gain during a look-back period
Average Loss = average percentage loss during a look-back period

$$SecondStepRSI = 100 - \frac{100}{1 + \frac{PreviousAverageGain*m+CurrentGain}{PreviousAverageLoss*m+CurrentLoss}}$$

2.4 Cryptocurrencies

Cryptocurrencies are digital and decentralized money, based on blockchain (ASHFORD; SCHMIDT, 2022). Unlike US Dollars or Brazilian Reais, there isn't a central authority to manage and maintain the value of the cryptocurrency. Cryptocurrencies are available for trading as any other assets available in a stock exchange and with the popularization of crypto trading inherited all the existing characteristics of classic stock trading, such as candlestick graphics and technical indicators.

Bitcoin was the first cryptocurrency and is the most famous one, created in 2008. Satoshi Nakamoto (2008), its creator, defines it as an "electronic payment system based on cryptography proof instead of trust". These cryptography proofs are registered and verified in a blockchain, which is "a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network" (IBM, 2022).

It is possible to use cryptocurrencies for payments today, however, it is still not what they are commonly used for today (ASHFORD; SCHMIDT, 2022). Currently, cryptocurrencies are more popular as an alternative investment option. Trading is also very popular. Anyone can invest and trade cryptocurrencies in exchanges, similar to how it works for stocks, commodities, and other investment instruments.

2.5 Artificial Intelligence (AI)

Considered the father of AI, John McCarthy (2004) described it as the following:

> It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

Machine learning, which is one of the applications of AI, can be powered by neural networks. The neural networks are an artificial model inspired by the way the biological neural networks process information in the human brain, as explained by Fumo (2017) in Towards Data Science blog. Neural networks can learn from data patterns such as stock or cryptocurrency data and their technical indicators, which are data in a time series format, widely used in the data science field.

While biological neural networks have neurons connected by synapses, artificial ones have nodes. These nodes receive inputs from other nodes or external sources and compute outputs. Each input has an associated weight based on its relative importance to other inputs and the node applies a function to a weighted sum of its inputs. The concept is that the weights are learnable and can control their positive or negative influence over other nodes.

The architecture of a neural network is composed of different parts (FUMO, 2017):

- **Input layer:** A block of input nodes with the purpose of passing the information to the next layer.

- **Hidden layer:** The hidden layer is where the processing and computation of data starts. The nodes in this layer receive the inputs from the input layer and then transfer the calculated weights to the next layer, which can be the output layer or another hidden layer. A neural network can also work without a hidden layer. However, it is not the case for this project.

- **Output layer:** The nodes in the output layer apply an activation function to the received inputs and output the result in the expected format.

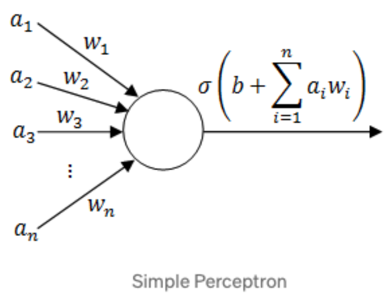
- **Activation function:** Given an input or set of inputs, a node's activation function defines the node's output.

2.6 Feedforward Neural Network

A feed-forward neural network is an artificial neural network where information moves only forward, from the input layer, into the hidden layer and then to the output nodes (FUMO, 2017). The information does not cycle through the layers. They are split into two types:

- **Single-layer Perceptron:** This type does not contain a hidden layer, so there is only one layer that works as an input and output layer. Figure 3 shows its design.

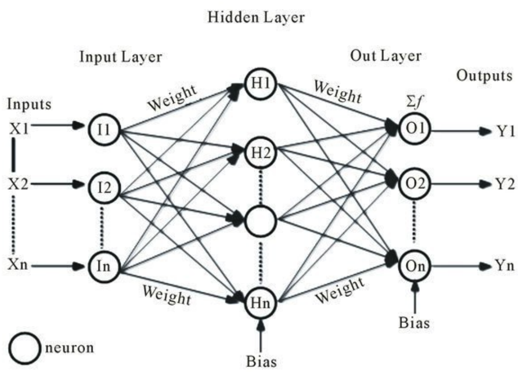
Figure 3 – Simple Perceptron



Source: Towards Data Science (2017).

- **Multi-layer perceptron (MLP):** This type contains at least one hidden layer of nodes that can learn non-linear representations. Figure 4 shows a visual representation of an MLP architecture.

Figure 4 – Multi-layer perceptron

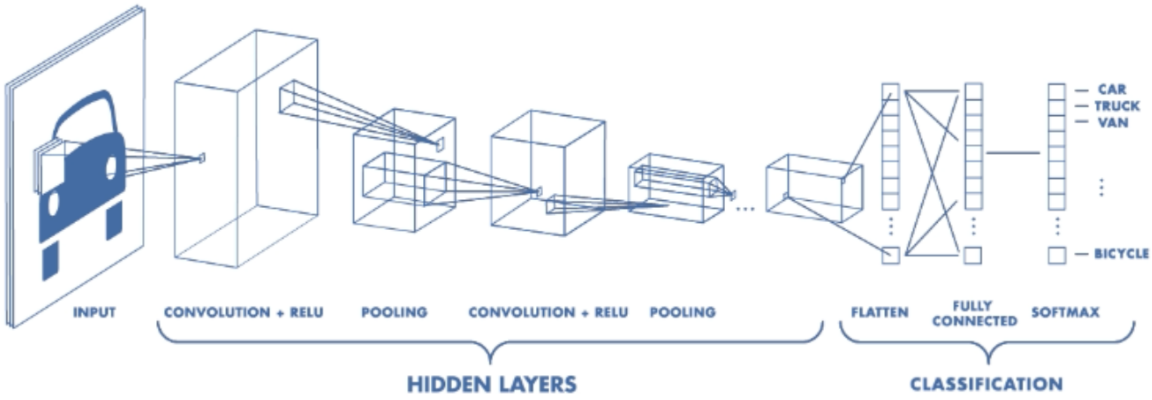


Source: Towards Data Science (2017).

2.7 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is designed to process data formatted as grid-like images (MISHRA, 2020). Figure 5 shows the layers that form a standard CNN.

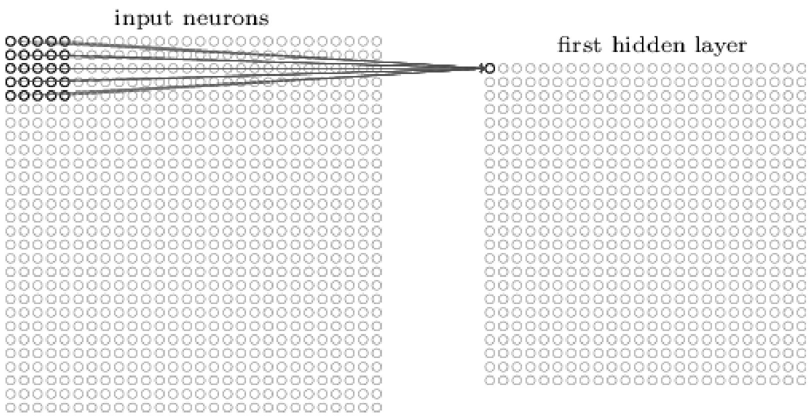
Figure 5 – Convolutional Neural Network Architecture



Source: Towards Data Science (2020).

- **Convolution Layer:** This is the main layer of the CNN. It is responsible for the most computational work done by the neural network. The convolution layer work as a filter (kernel) that take small portions of the input grid and gets the most important parts of it. The kernel creates a feature map that is smaller than the input grid, as shown in Figure 6 (ALVES, 2018).

Figure 6 – Convolution Layer

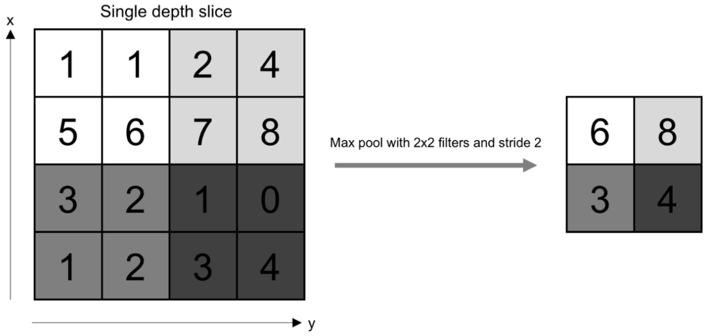


Source: Neuronio BR (2018).

The kernel is formed by weights that are randomly set for each new input through back-propagation.

- **Pooling Layer:** The pooling layer is responsible to reduce the spatial size of the grid by simplifying the information from the previous layer. It works by dividing the grid received in the pooling layer in neighborhoods and applying a function to it. The most commonly used process is max pooling, which gets the highest element of the neighborhood and adds it to the output grid, as shown in Figure 7 (MISHRA, 2020).

Figure 7 – Max pooling



Source: Towards Data Science (2020).

- **Fully Connected Layer:** The nodes in this layer are connected to all nodes from the previous and the next layer (MISHRA, 2020). It helps in the learning of non-linear combinations of the outputs from the previous layers.

- **Non-linearity Layers:** These layers are usually just after a convolution layer, as convolution is a linear process, and the data CNN processes usually are not (MISHRA, 2020). There are different non-linear functions, such as Sigmoid, Tanh, Softmax, ReLU, and others. Today ReLU is getting more popular for being more precise and computationally efficient compared to the other functions (ALVES, 2018).

3 RELATED WORK

This section presents papers selected based on their similarity to this research. The papers were found in Google Scholar using searching terms like *"trading bots"*, *"algorithmic trading"*, *"AI trading bot"* and *"CNN trading bot"*. The author analyzed different papers and the ones presented here are the most relevant to the research. Not all papers use artificial intelligence but their work was solid and presented interesting inputs.

Souza (2019) proposed a trend follower trading bot to be applied in Brazilian future mini USD contracts (WDO) and future mini IBOV index contracts (WIN). To detect a trend and take a decision, the algorithm processes the current value of different indicators, which have different weights according to De Souza's setup, and if the result of the total weight is either 1 or -1, it takes a long or short position respectively. The same indicators are used to decide when to close the trade. The trading bot acted on a simulation account in the month of April 2019, in a 5 minutes time frame. The best version of the WDO algorithm resulted in a 103% profit in the period, which is great. The algorithm for WIN had a 15% profit in the same period. At first sight, these results seem very impressive. However, the author did not consider the cost of each trade if they were taken in a real trading account. Over 1600 trades were done for WDO and almost 900 for WIN in one month of tests. In case the stock exchange taxes were applied by Souza, most likely neither of the setups would be profitable as the trading bot did many operations and the taxes are applied individually to each trade.

Santos (2020) developed another project for the Brazilian market. The project is composed of different parts: the first part is a bot created to follow a strategy using RSI, Stochastic, and MACD and buy, sell or do nothing based on the current value of the indicators. The second part is the integration between this bot and the machine learning algorithms of the third part. These algorithms are Support Vector Machines (SVM) and Multilayer Perceptron Artificial Neural Networks (MPL ANN). The bot from the first part was initially used to train the machine learning algorithms and then the machine learning algorithms were tested trading on their own. The backtest results were impressive: trading BBAS3 using a 1-minute time frame over a period of 3 months and an initial balance of 4.729,00 BRL, the SVM had a 2.853,00 BRL profit, and the MPL ANN a 5.378,00 BRL profit against 1.121,00 BRL profit of the bot without any ML. These amounts, however, do not consider the taxes, which would most likely wipe the profits from the ML algorithms as both did thousands of trades in the period.

The work from Nan, Perumal e Zaiane (2019) took a different approach. They implemented machine learning to gather the market sentiment of news headlines related to Microsoft, Amazon, and Tesla. Along with this information, their trading bot considered the current account balance, current stocks being held, the opening price of the day, and a short and long average of opening prices from previous days to take a decision for a given day. The bot was trained with four years of data (Jan 2014 to Dec 2017) and then tested with one year of data (2018). The bot was profitable in all 3 stocks and the results were slightly better when the market sentiment was applied to the bot's decision of buying, selling, or holding than when it was not considered.

Singh (2019) developed a trading bot using Python for his master's degree thesis in 2019. Two indicators were used to guide the bot: a 30 periods RSI and a 30 periods Bollinger Band. The trades were done in the hourly time frame. Once a trade is open, the bot will close it once it reaches either 1% profit or loss. Singh informs the bot ended up having small profits and suggests using smaller time frames to increase the number of trades and therefore the amount of profit.

A project conducted by Andersson et al. (2021) in 2021 compared the performance of a moving average strategy and of a random trade strategy with the performance of a trading bot that uses Long Short Term Memory (LSTM) to predict future prices to decide whether to buy or sell, all in Bitcoin. Focusing on the LSTM implementation, it was trained with data from 2017 to 2021 and its predictions were set to the next hour, meaning if the current time is 15:00, it will predict the likely Bitcoin price to 16:00. They concluded that the price predictions were mostly accurate and that the trading bot was more likely to make a profit when trading in shorter time frames. It makes sense as the price predictions are for the price one hour ahead, therefore if the bot takes trades in a time frame of minutes, it will be more likely to get the complete expected price movement.

Nascimento, Costa e Bianchi (2020) applied a Multichannel CNN (MCNN) architecture to determine where to sell or buy trying to get a financial result superior to a buy-and-hold strategy. The feature sets were built based on the stationary closing price of the asset along with four indicators. These multiplied by an 11-day window formed a 5x11 feature. They also created an algorithm to label the data. The CNN architecture consisted of an input tensor, two convolutional layers, two max-pooling layers, and an output layer. The training happened in some stocks in the Brazilian Stock Exchange (BM&F) in 1000 days period. Out of four assets tested, three of them performed well above a buy-and-hold strategy and the other was slightly more profitable than the buy-and-hold.

3.1 Comparison Table

To compare the papers, some elements were selected as criteria:

- Market: the market that the bot traded on;
- Assets: which assets within the market the bot traded on;
- Time Frame: time frame in which the bot traded on;
- Strategy: trading model applied by the bot;
- Indicators: list of indicators used in the paper;
- AI: A brief explanation of how AI was applied in the work.

Table 1 shows these elements for each paper.

All the selected papers that use AI have it to replace the trading bot, depending only on the model's precision to identify where to buy or sell a given asset. Also, in most trading bots presented, there is an over-trading issue. The execution of too many trades generates a huge cost and usually wipes all the profit from the operations. A gap that has not been explored yet is to use AI to assist a trading bot that follows a proven trading strategy. This assistance would tell the bot when the macro scenario is appropriate for the strategy, making sure it only takes trades during the most profitable moments in the market. This would involve a macro scenario analysis as the AI would analyze longer periods of time and the bot would take trades in shorter periods. A CNN would be able to analyze the macro scenario based on indicators, similarly to what Nascimento, Costa e Bianchi (2020) developed. However, the labeling would happen according to the results of the trading bot in each period of time, so the AI can learn from the trading bot results and the indicators in a larger time frame what are the best macro scenarios for trading.

Table 1 – Comparison of related papers

Paper	**Market and Assets**	**Time Frame**	**Strategy**	**Indicators**	**AI**
Robot Model for Trend Following Operations in the Brazilian Stock Exchange, 2019.	BM&F: WDO and WIN	5 min	Trend Follower	SMA, Hilo Activator, MACD, RSI, Bollinger Bands	–
Study and Development of a Trading Bot Using SVM and MLP for Stock Purchase and Sale Management at BM&F BOVESPA, 2020.	BM&F: BBAS3	1 min	AI	RSI, Stochastic, MACD	Separated SVM and MLP algorithms trained based on a trading bot using some indicators and bot's decision to buy, sell or do nothing and the profit of the operation (positive or negative).
Sentiment and Knowledge-Based Algorithmic Trading with Deep Reinforcement Learning, 2019.	NASDAQ: MSFT, AMZN and TSLA	1 day	Based on previous openings and market sentiment	SMA	Q-learning algorithm applied to decide the next action based on market sentiment from news headlines and the account balance.
Trading Bot, 2019.	Crypto: LTC/USD	1 min	Confluence between Bollinger Bands and RSI	Bollinger Bands and RSI	–
Crypto Proxima - An analysis of autonomous Bitcoin trading, 2021.	Crypto: BTC	multiple	Based on price prediction from LSTM	SMA	LSTM for price prediction using price as a parameter to teach the AI.
Stock Trading Classifier with Multichannel Convolutional Neural Network	BM&F: BOVA11, ITUB4, VALE3 and PETR4	1 day	AI using indicators	SCP, RSI, WR, MI, MACD	MCNN to analyze features created from time series stock data and define buy and sell points.

Source: The author (2022).

4 METHODS

4.1 Methodology

This section describes how the trading bot was built and how the AI algorithm was trained and integrated with the trading bot, detailing all modules developed for the research.

All the cryptocurrency data was taken from an exchange. Binance was chosen for this purpose as it is the biggest exchange in the world, with the highest trading volume, and has a well-documented API that fulfills all the needs of the project. From the OHLCV data provided by Binance, the indicators used in the model will be calculated. These are the indicators that will be used:

- 21 period Keltner Channels with a 0.38 ATR multiplier
- 2 period RSI
- 8 period EMA
- 80 period EMA

At the AI end, a Convolutional Neural Network architecture was used to decide whether the macro scenario is in a good moment for trading or not. The CNN is powered by Keras, a neural network library that runs on TensorFlow, an open-source platform for machine learning.

The data processing, the trading bot, feature-building logic, and the CNN Model was written in Python. The language was selected as all the libraries required to build the project are available and documented in Python. Also, it is a modern and widely used language nowadays. The technical analysis library used was Pandas-TA, due to its ease of use and versatility. The backtesting library chosen was Backtesting.py as it has only what was needed for this project and is simple to be implemented.

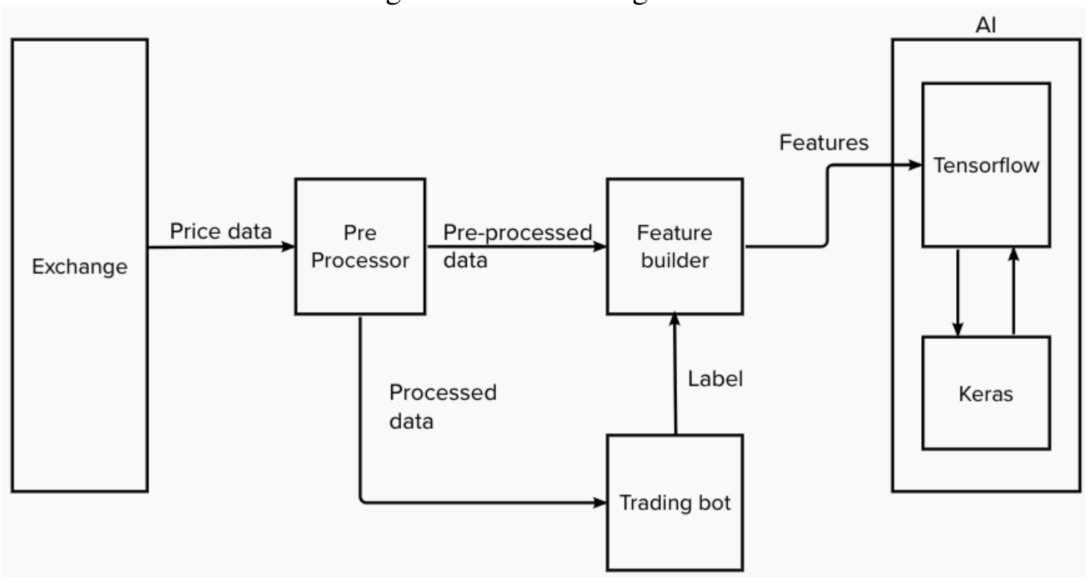
The model was tested in five different cryptocurrencies:

- **Bitcoin (BTC)**: The first and most famous cryptocurrency. Currently holds over 40% of the total crypto market cap (COINMARKETCAP, 2022).
- **Ethereum (ETH)**: Second most famous cryptocurrency and second largest in market cap (COINMARKETCAP, 2022).
- **Binance Coin (BNB)**: Currency created by the largest exchange in the world, Binance. Currently, the 6TH largest crypto in market cap (COINMARKETCAP, 2022).
- **Cardano (ADA)**: Blockchain created as a more environment-friendly alternative to Ethereum. Currently, in the top 10 largest cryptocurrencies (COINMARKETCAP, 2022).
- **Solana (SOL)**: An open-source project created to support DeFi solutions. Currently holds 11th place in crypto market cap (COINMARKETCAP, 2022).

4.2 Implementation

There are two major building blocks in the project: the AI Training block and the Crypto Trading block. The AI Training block is responsible to train the Convolutional Neural Network algorithm. Its flow is shown in Figure 8:

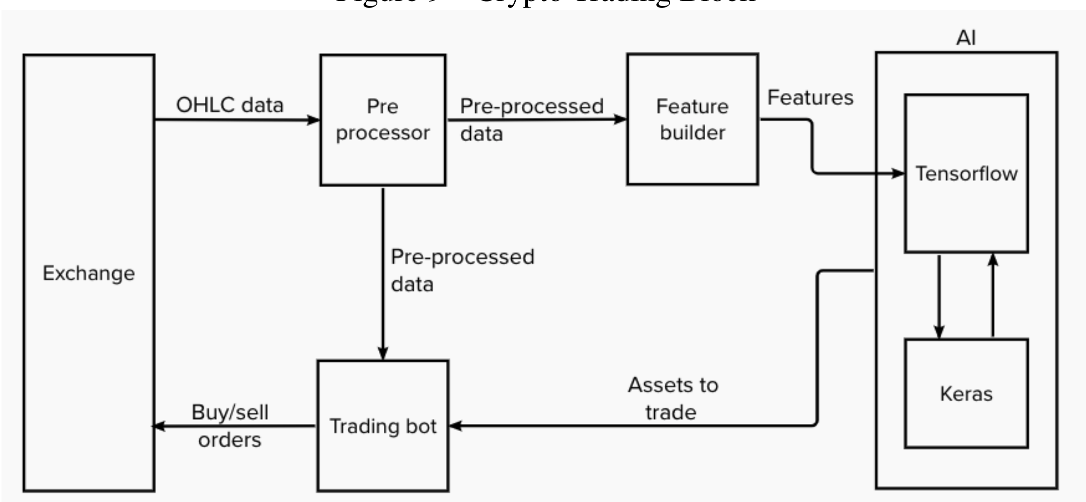
Figure 8 – AI Training Block



Source: The Author (2022).

The Crypto Trading Block is responsible for using the CNN Model trained in the AI Training Block to predict the data for the backtest and then run the backtests. Figure 9 shows an overview of the block. The blocks are explained in detail in the following sub-sections.

Figure 9 – Crypto Trading Block



Source: The Author (2022).

4.2.1 Pre-Processor

The pre-processor is responsible to get the raw OHLCV data provided by Binance in a JSON format and transform it into a Pandas data frame with the columns required by the Backtesting.py library to perform the backtests. The OHLCV data was previously taken using Binance's API calls via Postman. The data was saved into files so for each test, it was already available locally.

The pre-processor gets data for 15-minute intervals from the exchange. Once the processing is done, the OHLCV values are available to be used in any other module.

4.2.2 Trading Bot

The Trading Bot block uses the Backtesting.py and Pandas-TA libraries. It is the one that contains the trading strategy and is responsible to calculate the technical indicators and run the backtests.

The trading strategy is based on 21 periods Keltner Channels with an ATR multiplier set to 0.38 and a 2-period RSI. A buy happens when a bearish candle closes above the Keltner Channels and the RSI is below 20. The stop loss is set in the previous bottom and the take profit is set as twice the size of the stop. Figure 10 shows examples of buy entry points and their respective stop and take points.

Figure 10 – Buy Trade Example

Source: The Author and TradingView (2022).

A sell happens when a bullish candle closes below the Keltner Channels and the RSI is above 80. The stop loss is set in the previous top and the take profit is set as twice the size of the stop. Figure 11 shows examples of sell entries and their stop loss and take profit points.

Figure 11 – Sell Trade Example

Source: The Author and TradingView (2022).

During the development of the project, it was noticed that for consolidated regions, trades tend to be less assertive, therefore an extra condition was added for entering a long or short position: the distance between the upper and lower Keltner Channels must be higher than a specific trigger. This trigger was determined individually for each crypto with the assistance of an optimizer available in Backtesting.py library.

For the AI Training module, the trading bot module does the backtesting of the trades. The data of each trade is then available to be used by the Feature Builder, which will define the labels of the training features.

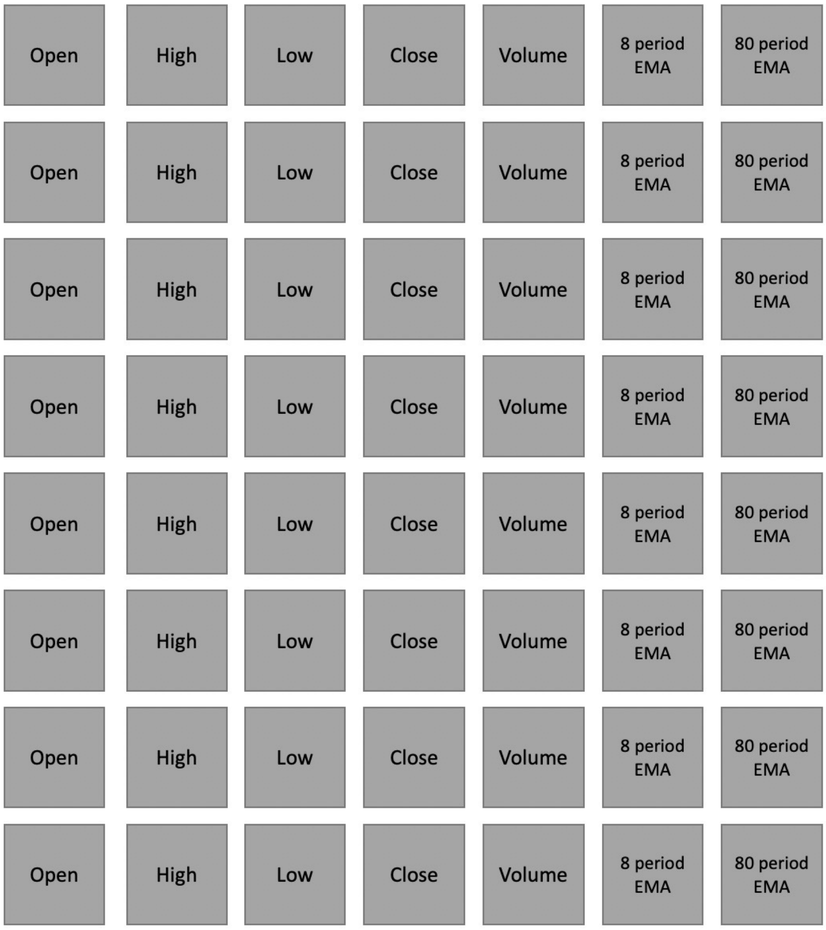
To use the inputs from the CNN model, the trading bot reads the predicted labels and knows if a trade can be done or not based on that.

For each interval of 2 hours, the net result will define the label. If the strategy was profitable during the 2 hours, the label sent to the Feature Builder is 'profit'. If it was not profitable, the label sent is 'loss'. These 2 hours intervals will start every 15 minutes, so it matches the data that the Feature Builder will receive from the Pre-Processor.

4.2.3 Feature Builder

The Feature Builder is in charge of labeling the features and organizing the data to be sent to the AI block as the input data. The features are composed of 8 sets of OHLCV data, and 8 and 80-period EMAs meaning they will represent a 2-hour market data in a 15-minute time frame. Figure 12 shows a visual representation of a feature.

Figure 12 – Feature Example



Source: The Author (2022).

From the second feature and beyond, every 30-minute interval will build a new feature, discarding the 2 oldest rows and adding the fresh new data in the two first rows. For each set of 8 rows, a function called *searchTrade* is called to find if a trade happened in that 2-hour window. Whenever a trade happens and has a negative return, a label is set as 0, which means **do not trade**. If there are no trades or the trade has a positive return, the label for that 8-row set is marked as 1, meaning **clear to trade**.

The Feature Builder then returns a NumPy array of the features and another one of the labels, both with the same length. Figure 13 shows the Feature Builder code:

Figure 13 – Feature Builder Code

```python
def getIndicators(data):

    data.ta.ema(length=8, append=True)
    data.ta.ema(length=80, append=True)

    return data

def searchTrade(candle,trades):

    entry = trades[trades['EntryBar'].between(candle,candle+8)]
    exit = trades[trades['ExitBar'].between(candle,candle+8)]

    if exit['ExitBar'].any() and (entry['ReturnPct']<0).any():
        return 0
    else:
        return 1

def buildFeature(asset):

    trades = go(asset)
    asset.reset_index(inplace=True,drop=True)
    asset = getIndicators(asset)
    asset.drop('Volume', axis=1, inplace=True)

    features = []
    results = []

    i = 0

    while i < len(asset) - 7:
        features.append(asset.iloc[i:i+8])
        results.append(searchTrade(i, trades))
        i+=2

    del features[:40]
    del results[:40]

    return features, results
```

Source: The Author (2022).

4.2.4 TensorFlow and Keras (AI)

This module is where the CNN model was built. As explained at the beginning of this section, the Convolutional Neural Network architecture was powered by Keras through TensorFlow. The goal of AI was to classify the current macro scenario as good to trade or bad to trade.

The hidden layer of the CNN model contains two blocks of 2d convolution layer, relu activation layer, and 2d max pooling layer. After that, for the classification, a flatten layer and two dense layers are used. Crossentropy loss and adam optimizer was used in the model.

CNNs are usually used to classify images and their input is a feature frame. The features generated by the Feature Building block were first used for training and validation. Each cryptocurrency had 7450 features with data ranging from Dec 10th, 2020 to May 10th, 2021. 10% of these features were used for validation. The training parameters were empirically tested until it reaches a good accuracy. A batch size of 32 was used along 100 epochs.

Tests started once the AI training was finished. The testing data ranged from July 21st, 2021 to November 21st, 2021. The outputs from the features were used as a flag to trade or not trade in the period by the Trading Bot. The results are presented in the next session.

5 RESULTS AND DISCUSSIONS

The trading data for each cryptocurrency was predicted by the CNN model individually. Accuracy, precision, recall and F1 metrics for each cryptocurrency are shown in Table 2:

Table 2 – Accuracy of CNN model

Cryptocurrency	Accuracy	Precision	Recall	F1
ADA	90.78%	95.51%	94.71%	95.11%
BNB	91.70%	95.25%	95.99%	95.62%
BTC	92.09%	98.96%	92.94%	95.84%
ETH	93.96%	95.13%	98.76%	96.91%
SOL	89.22%	93.67%	94.66%	94.16%

Source: The author (2022).

All the metrics had a very high percentage, which indicates the CNN model was able to precisely identify moments that historically tend to present entry points that would end up in a loss. However, upon crossing the data on Table 2 with the financial results of the trading bot assisted by AI, it is possible to see that the CNN model was not as effective as it initially seems.

Table 3 shows the financial return for the cryptocurrencies between July 21st, 2021 and November 22nd, 2021 for the trading bot (125 days), the trading bot assisted by AI, the trading bot using the actual data used as the benchmark for the CNN model predictions and a buy-and-hold strategy. All the tests were made with a starting capital of 100,000.00 USD. Binance taxes were not considered as the cost for each 100,000.00 USD traded is only 1.08 USD (BINANCE, 2022). If they were considered, the returns would be approximately 0.3% lower.

Table 3 – Financial results and number of trades

Cryptocurrency	Bot without AI	Bot with AI	Bot w/ AI benchmark data	Buy-and-hold
ADA	17.25%	18.21%	25.07%	51.37%
BNB	16.26%	11.62%	19.57%	93.91%
BTC	11.33%	8.54%	19.34%	77.01%
ETH	12.31%	15.17%	15.60%	111.39%
SOL	15.02%	12.86%	43.31%	714.66%

Source: The author (2022).

As the market was bullish in the period used for testing and the trading management strategy is based on a 2 to 1 take-profit to stop-loss ratio rather than closing the trade when a specific condition happens, the best results were expected to be from the buy-and-hold strategy, which was the case.

The Trading Bot by itself had a positive outcome in all five cryptocurrencies, averaging 14.43% return in a period of four months. However, considering the buy-and-hold return for the same period, it is a poor performance. This shows that the trading strategy set up in the bot is not ideal for extremely bullish market conditions.

When assisted by AI, the average dropped to 13.28%. The returns dropped considerably in all cryptocurrencies but ADA, which increased by almost 1%, and ETH, which increased by nearly 3%. Considering the benchmark labels for trading and applying them with the Trading Bot, the results get better, with an average of 24.58%. In this scenario, ETH increased approximately the same as with the AI, while all the other cryptocurrencies had a greater increase in performance compared to the AI, especially SOL, which increased by 27.05%.

These results show that identifying the best moments to trade can improve trading returns. However, the implemented CNN model was not effective to replicate the results even in a macro scenario similar to the one it was trained.

Breaking down each cryptocurrency result individually, we can see some trends:

- Cardano (ADA):

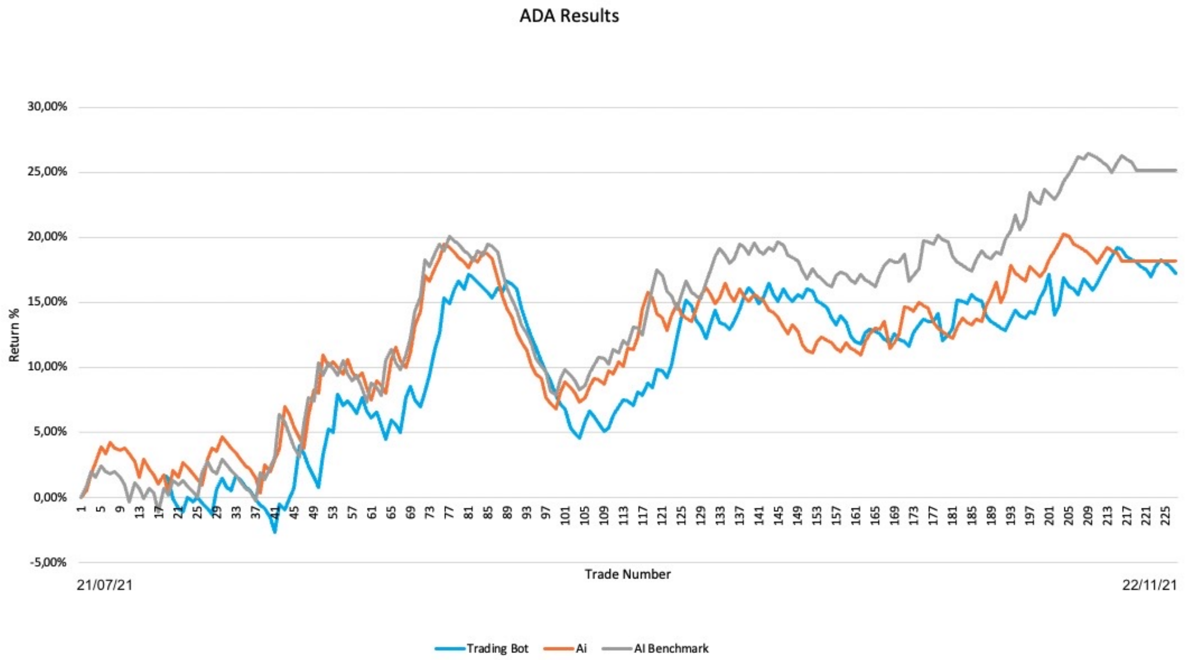
Table 4 – ADA Results

Trading Model	Return	Number of Trades	Win Rate
Bot without AI	17.25%	223	38.57%
Bot with AI	18.21%	210	39.05%
Bot with AI Benchmark Data	25.07%	184	41.85%

Source: The author (2022).

For ADA, the return had a little increase comparing the trading bot without and with AI's assistance. Compared to the bot trading with the AI benchmark data, the returns increase by almost 8%. It is possible to see the win rate increasing and the number of trades decreasing, therefore the AI predictions and AI benchmark indeed prevented the trading bot from doing trades that, without their assistance, would have resulted in a loss. Figure 14 shows a graphic of the trade-by-trade evolution of capital over time.

Figure 14 – ADA Results



Source: The Author (2022).

- Binance Coin (BNB)

Table 5 – BNB Results

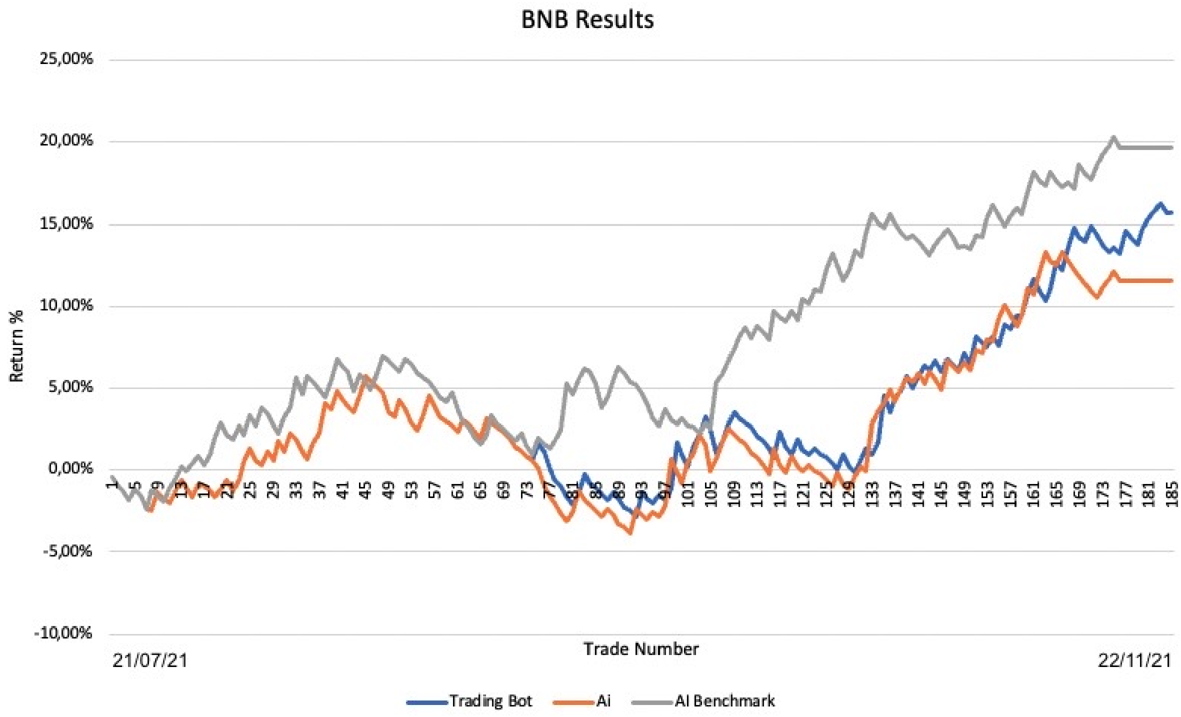
Trading Model	Return	Number of Trades	Win Rate
Bot without AI	16.26%	184	38.59%
Bot with AI	11.62%	176	36.93%
Bot with AI Benchmark Data	19.57%	147	42.18%

Source: The author (2022).

BNB presented a negative outcome for the AI-assisted bot: the return decreased by 4.64% in comparison with the non-assisted bot. The number of trades decreased as well, however, the win rate decreased with it, showing that the trades that did not happen due to AI's prediction were mostly winning trades. On the other hand, with the assistance of the AI benchmark data, the return increased by 3.31%, along with a lower number of trades and a higher win rate. Figure 15 shows the visual representation of the trade-by-trade evolution of capital over time.

- Bitcoin (BTC)

Figure 15 – BNB Results



Source: The Author (2022).

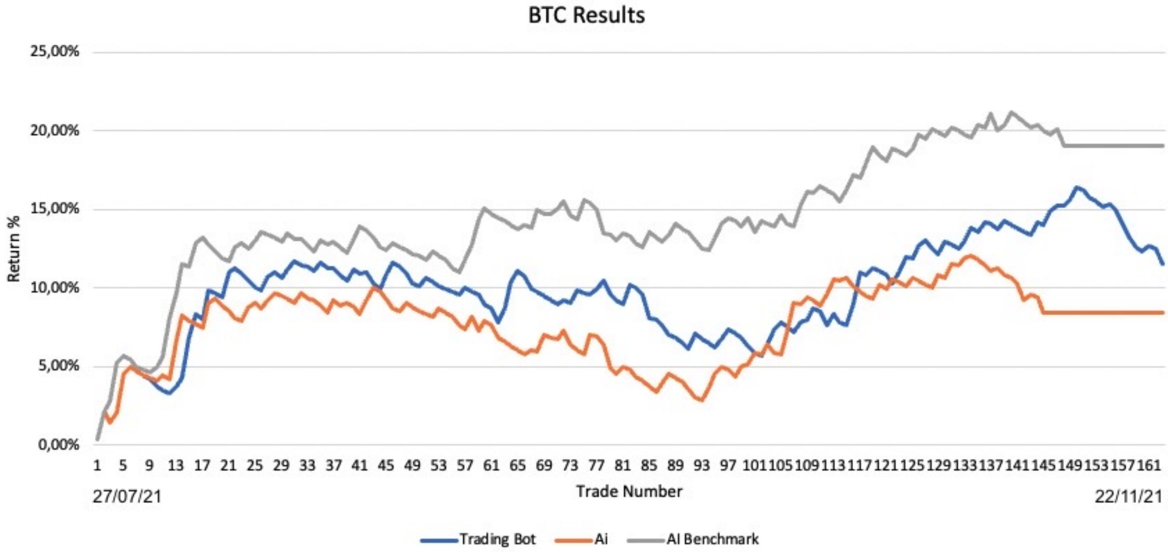
Table 6 – BTC Results

Trading Model	Return	Number of Trades	Win Rate
Bot without AI	11.33%	163	38.04%
Bot with AI	8.54%	145	37.24%
Bot with AI Benchmark Data	19.34%	148	41.89%

Source: The author (2022).

BTC had a close outcome to BNB: the AI-assisted bot was outperformed by the non-assisted bot by 2.79% even though the number of trades decreased, showing again the CNN model was not able to efficiently identify moments that are not good for the trading strategy. The results of the AI's benchmark data-assisted bot were excellent, with over 8% increase in return with fewer trades than the non-assisted bot but a few more than the AI-assisted one and a higher win rate, which again shows the efficiency of the model. Figure 16 shows the visual representation of the trade-by-trade evolution of capital over time.

Figure 16 – BTC Results



Source: The Author (2022).

- Ethereum (ETH)

Table 7 – ETH Results

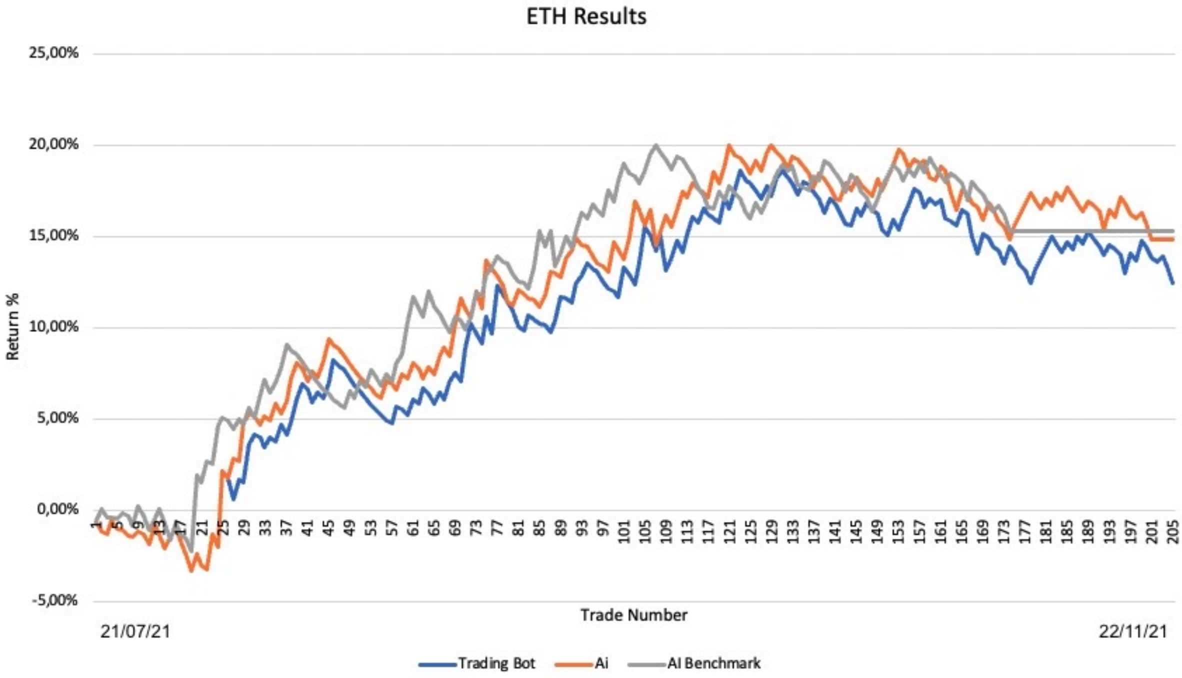
Trading Model	Return	Number of Trades	Win Rate
Bot without AI	12.31%	205	36.10%
Bot with AI	15.17%	201	36.81%
Bot with AI Benchmark Data	15.60%	174	39.08%

Source: The author (2022).

The results for ETH were at some level close to ADA's. The return increased by almost 3% for the AI-assisted bot against the non-assisted bot. The number of trades decreased by only a few trades and the win rate increased only by 0.71%. The difference between ADA appears in the bot assisted with AI's benchmark data: although the number of trades had a significant decrease and the win rate increased by almost 2.5%, the return is virtually the same as the AI-assisted bot. This indicates that the trades avoided by the benchmark data, even being mostly failed trades, were not financially significant.

This can happen as the stop-loss and take-profit are variable, depending on the previous pivot points, therefore we can have failed trades with a low percentage loss - and successful trades with low percentage gains as well. Figure 17 shows the visual representation of the trade-by-trade evolution of capital over time.

Figure 17 – ETH Results



Source: The Author (2022).

- Solana (SOL)

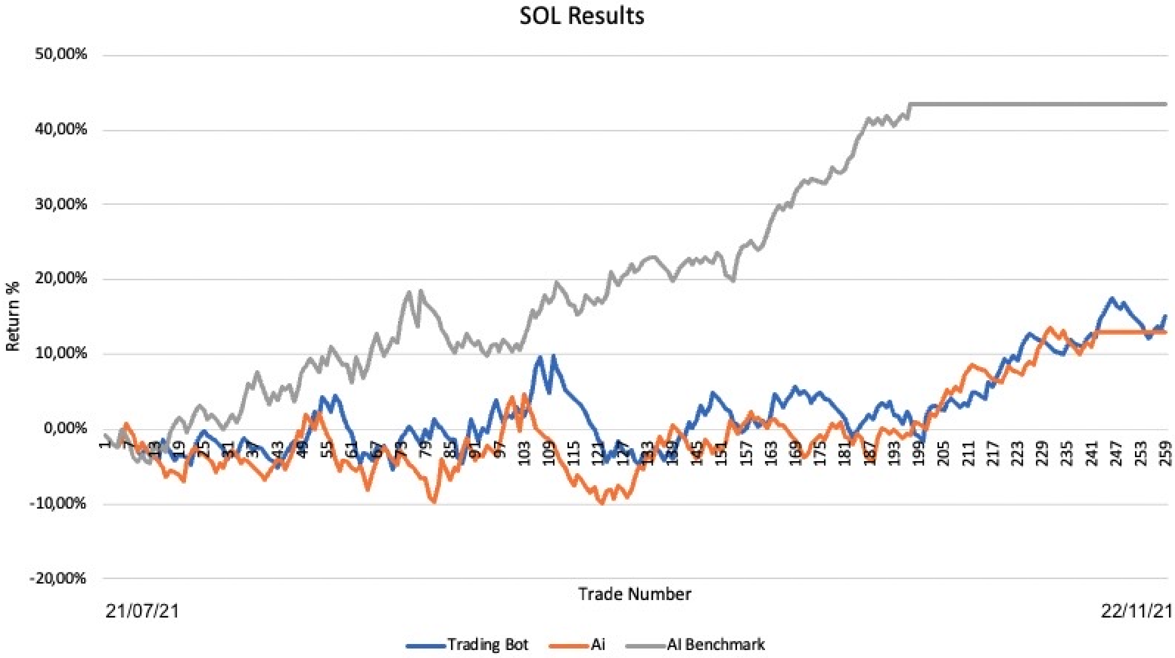
Table 8 – SOL Results

Trading Model	Return	Number of Trades	Win Rate
Bot without AI	15.02%	259	39.75%
Bot with AI	12.86%	242	39.67%
Bot with AI Benchmark Data	43.31%	197	47.72%

Source: The author (2022).

Results from SOL when for non-assisted and AI-assisted bots are alike to BNB's and BTC's: the return for the AI-assisted bot decreased even though fewer trades were made and the win rate decreased, meaning CNN predictions could not efficiently identify moments that are not good to trade. However, for the benchmark AI-assisted bot, the return is almost 3 times the return of the non-assisted bot and almost 3.5 times the return o the AI-assisted bot. The benchmark data was extremely efficient to identify bad moments to trade, as the number of trades decreased significantly and the win rate jumped from around 39% to almost 48%. As the trading strategy applied by the trading bot has a 2-to-1 profit-loss ratio, this jump in the win rate had an exponential increase in the financial return. Figure 18 shows the visual representation of the trade-by-trade evolution of capital over time.

Figure 18 – SOL Results



Source: The Author (2022).

Overall, it is possible to observe a relatively low number of trades by day, which is good as more trades means more taxes, ranging from an average of 2.07 to 1.16 trades per day.

Comparing the buy-and-hold results to the trading-bot results, with and without assistance and benchmark assistance, there is no clear relation between the results and the volatility of the cryptocurrency. ADA and SOL have the best returns on average with the trading bot, however, they had the lowest and highest returns in buy-and-hold, respectively. This makes sense as the trading model is focused on intra-day short-term trades, so the trend of a long period of time - 4 months in the tests - is not so relevant for the results.

6 CONCLUSIONS

The work showed an implementation of a Convolutional Neural Network aiming to identify moments in the macro scenario of the cryptocurrency market that is ideal for a specific trading strategy. An automated trading bot was developed to perform long and short trades using a strategy based on Keltner Channels and RSI in a variety of popular cryptocurrencies. The trading bot was tested with and without AI's assistance to validate the CNN model and find out its efficiency.

A limitation found was a way to increase the efficiency of the CNN model, as adjustments can be done in the layers to try to get better results. Also, the backtesting library used for testing can only trade full units of each asset, which means if BTC is at 40,000.00 USD and the total balance available is 100,000.00 USD, trades will happen with 2 units of BTC, meaning a total of 80,000.00 USD at stake, rather than the full 100,000.00 USD. This can affect the financial results. Another limitation was the type of market the AI was trained and tested on. Both were bull market runs. However different results might be presented if the AI is trained and tested in different conditions such as dull or bear market runs.

The contribution of this work is untold as none of the articles found had the same approach. An innovative way to use AI in trading was presented, using its power to instruct and enable/disable a trading bot with a proven working model instead of learning a model on its own. The paper explored this proposal of AI use with trading in a very specific manner, using CNN and features built in a grid-like format with cryptocurrency data. There are multiple fronts and possibilities that can be explored using the concept of a trading bot working with an AI rather than the AI working like a trading bot.

The results of the assisted trading bot were not positive, even though it presented a high percentage in all the presented metrics. This leaves an opportunity for future research, more focused on improving the CNN model to be able to have a greater financial return. Differently, it can also be trained in different market conditions, time-frames, and assets, including stocks. Other AI methods such as SVM, MLP, and LSTM can also be used for tests in research work with the same AI-trading model.

Lastly, the labeling process presented in this article proved to be efficient to identify the market conditions as all the returns increased with it. This showed that the AI-trading relation idealized here can be very positive if an AI model that can efficiently repeat this labeling process in a live market can make big improvements in trading results.

References

ALVES, G. **Entendendo Redes Convolucionais (CNNs)**. 2018. Available at: <https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>. Access in May 17th 2022.

ANDERSSON, S. et al. **Crypto Proxima: An analysis of autonomous Bitcoin trading**. Tese (Doutorado) — Uppsala Universitet, Uppsala, Sweden, 2021.

ASHFORD, K.; SCHMIDT, J. **What Is Cryptocurrency?** [S.l.], 2022. Available at: <https://www.forbes.com/advisor/investing/cryptocurrency/what-is-cryptocurrency/>. Access on May 13th, 2022.

BARBER, B. M. et al. **Do Day Traders Rationally Learn About Their Ability?** 34 p. — Graduate School of Management, University of California, Davis and Guanghua School of Management, Peking University and Haas School of Business, University of California, Berkeley, 2017.

BEERS, B. **Buy and Hold Definition**. [S.l.], 2020. Available at: <https://www.investopedia.com/terms/b/buyandhold.asp>. Access on June 17th, 2022.

BINANCE. **Trading Fees**. 2022. Available at: <https://www.binance.com/en-US/fee/futureFee>. Access on Nov 1st, 2022.

CAPITAL RESEARCH BRASIL. **O que é trading e como funciona essa forma de negociação?** [S.l.], 2020. Available at: <https://www.capitalresearch.com.br/blog/investimentos/trading/>. Access on May 12th, 2022.

COINMARKETCAP. **Gráficos Globais de Criptomoedas**. [S.l.], 2022. Available at: <https://coinmarketcap.com/pt-br/charts/>. Access on June 12th, 2022.

COULLING, A. **A Complete Guide To Volume Price Analysis**: Read the book then read the market. [S.l.]: Createspace, 2013.

FERNANDO, J. **Moving Average (MA)**. [S.l.], 2021. Available at: <https://www.investopedia.com/terms/m/movingaverage.asp>. Access on May 13th, 2022.

FUMO, D. **A Gentle Introduction To Neural Networks Series — Part 1**. 2017. Available at: <https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc>. Access in May 15th 2022.

HASHDEX. **Fundos cripto: número de investidores cresce 1.266% no Brasil em 2021**. [S.l.], 2022. Available at: <https://www.hashdex.com.br/conteudo/artigos-e-opinioes/fundos-cripto-numero-de-investidores-cresce-1266percent-no-brasil-em-2021>. Access on May 5th, 2022.

HAYES, A. **Dollar-Cost Averaging (DCA)**. [S.l.], 2022. Available at: <https://www.investopedia.com/terms/d/dollarcostaveraging.asp>. Access on June 17th, 2022.

IBM. **What is blockchain technology?** [S.l.], 2022. Available at: <https://www.ibm.com/topics/what-is-blockchain>. Access in May 13th, 2022.

ISMAR, B. **HFT: como funcionam as negociações de alta frequência?** 2021. Available at: <https://renovainvest.com.br/blog/hft-como-funcionam-as-negociacoes-de-alta-frequencia/#O_que_e_high_frequency_trading_HFT>. Access on May 12th, 2022.

MCCARTHY, J. **What is artificial intelligence?** Tese (Doutorado) — Computer Science Department, Stanford University, Stanford, CA, 2004.

MISHRA, M. **Convolutional Neural Networks, Explained**. 2020. Available at: <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>. Access on May 17th, 2022.

MITCHEL, C. **Keltner Channel**. [S.l.], 2021. Available at: <hhttps://www.investopedia.com/terms/k/keltnerchannel.asp>. Access on Aug 22nd, 2022.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. 2008.

NAN, A.; PERUMAL, A.; ZAIANE, O. R. **Sentiment and Knowledge Based Algorithmic Trading with Deep Reinforcement Learning**. Tese (Doutorado) — University of Alberta, Canada, 2019.

NASCIMENTO, D. G.; COSTA, A. H. R.; BIANCHI, R. A. C. **Stock Trading Classifier with Multichannel Convolutional Neural Network**. Tese (Doutorado) — FEI University Center and Universidade de Sao Paulo, Sao Paulo and Sao Bernardo do Campo, Brazil, 2020.

SANTOS, A. C. dos. **Study and Development of a Trading Bot Using SVM and MLP for Stock Purchase and Sale Management at BMF BOVESPA**. Tese (Doutorado) — Centro Universitário Padre Anchieta, Jundiai, Brazil, 2020.

SINGH, C. **Trading Bot**. Tese (Doutorado) — California State University, Northridge, CA, United States, 2019.

SOUZA, T. K. B. de. **Robot Model for Trend Following Operations in the Brazilian Stock Exchange**. Tese (Doutorado) — Universidade do Vale do Rio dos Sinos, Curitiba, Brazil, 2019.

UNIVERSITY OF CHICAGO. **More Than One in Ten Americans Surveyed Invest in Cryptocurrencies**. [S.l.], 2021. Available at: <https://www.norc.org/NewsEventsPublications/PressReleases/Pages/more-than-one-in-ten-americans-surveyed-invest-in-cryptocurrencies.aspx>. Access on May 5th, 2022.

YAPPI. **Trading Bots – O que são e como funcionam**. [S.l.], 2020. Available at: <https://yappi.com.br/trading-bots/>. Access on May 12th, 2022.