Bitcoin Price Prediction with 2D Convolutional Neural Networks

Jerry Fanelli
Advisor: Dr. Francesco Grossetti

Bocconi University
Master of Science in Data Science and Business Analytics

10/15/2021

What is Bitcoin?

"A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution"

Satoshi Nakamoto (2008)

Bitcoin is based on cryptographic proofs where transactions are recorded in the **Blockchain**, a distributed database consisting of a chain of blocks containing transactions.

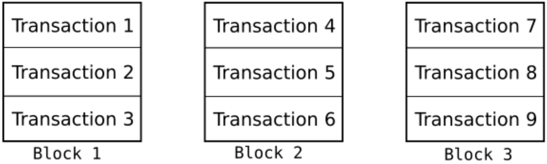


Figure 1: Blockchain structure

Issue of Predicting Bitcoin Price

Problems with Financial Time Series data:

1. Inherently noisy
2. Non-stationary
3. Deterministically chaotic

Deep Learning models are emerging as the best performing models to predict Financial Time Series, while cryptocurrency combined with Financial Time Series Forecasting is still a very novel and unexplored field.

The goal of this thesis is to test an innovative and untested approach to classify the price of Bitcoin in a way that can later be used on *algorithmic trading scenarios*.

Statement

Bitcoin Financial Time Series price prediction as a binary classification problem.

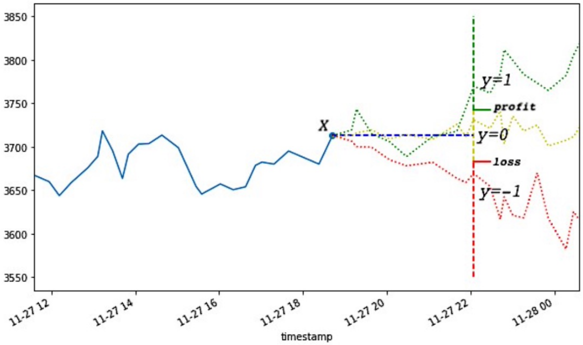


Figure 2: Financial Time Series price binary classification

Research Design

Predition task approached similarly to a computer vision problem, leveraging a **2D Convolutional Neural Network**.

Data used:

- Open, High, Low, Close (OHLC) Bitcoin price
- On-Chain Features and Metrics

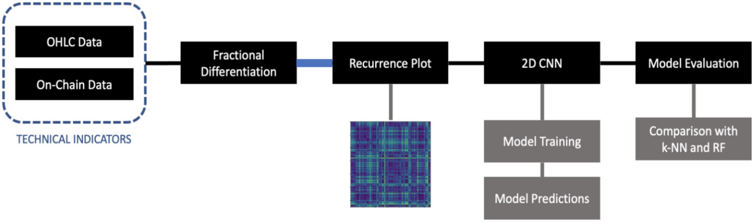
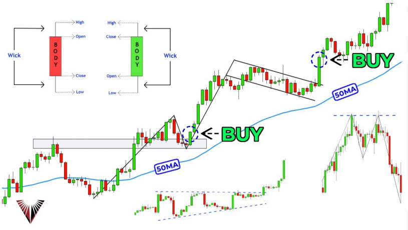


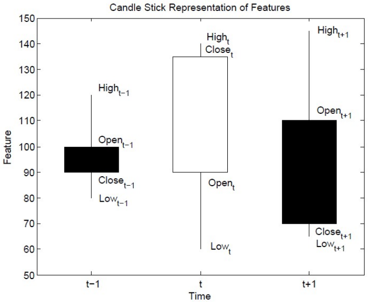
Figure 3: Methodology pipeline

The Data — OHLC Data

Technical Analysis: A trading discipline employed by traders to identify trading opportunities in price trends and patterns;

Open, High, Low, Close (OHLC) data is the standard type of price data used on technical analysis and it is usually represented through candle sticks.





The Data — On-chain Metrics

On-Chain Analysis: Emerging field aiming at analyzing the available data recorded on the Blockchain to facilitate decision making. It is mostly applied for trading and investment purposes.

Bitcoin: Realized HODL Ratio

glassnode

Data Labels — Triple Barrier Labeling

Data Labeling

Data Labeling is the task of detecting and tagging data with labels to provide context so that a machine learning model can learn from it.

Marco Lopez de Prado (2018) proposed the **Triple Barrier Labeling** where each observation is labelled based on the first barrier touched on three barriers.

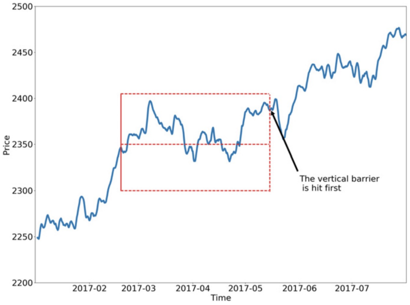


Figure 5: Triple Barrier Labeling

Comparing Returns: Triple Barrier vs Buy-and-Hold

Buy-and-Hold Strategy

The Buy-and-Hold strategy consists of buying the asset at time t_0 and holds it until time t_{0+h} where h coincides with the last day of the data.

Naive Trading Algorithm

The Naive Trading Algorithm consists in buying the maximum amount of assets when $y_t = 1$, so when there is a buying opportunity, and sell all the asset in our possession when $y_t = 0$, so when there is a good selling opportunity.

Comparing Returns: Triple Barrier vs Buy-and-Hold

Strategy	Total Return	Sharpe Ratio
Buy-And-Hold	$\times 5,264$	0.0640
Triple Barrier Based	$\times 402,543$	0.4318

Table 1: Buy-and-Hold vs. Triple Barrier Strategy

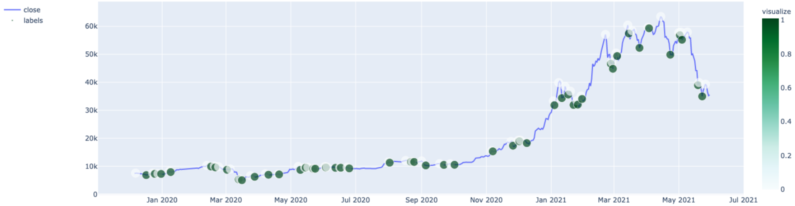


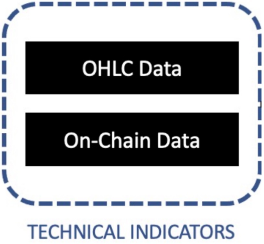
Figure 6: Triple Barrier Strategy - last 500 days sample

Technical Indicators

Technical Indicators

Tecnical indicators are mathematically-based tools used to analyze the past and anticipate future price trends and patterns.

A model trained on both OHLC and On-Chain data transformed with technical indicators, should be comparable to a group of traders who jointly perform technical analysis on not only the Bitcoin price, but also on all On-Chain metrics.



Stationarity and Differentiation

Modelling of Financial Time Series is extremely difficult mostly because of two properties:

1. Remarkable amount of noise;
2. Great amount of trends that shift the series' mean over time;

Stationarity Time Series

A stationary time series is one whose properties do not depend on the time at which the series is observed. The statistical properties of the process generating the time series do not change over time.

Stationarity and Differentiation

Usually to make a time series stationary, we apply a first differentiation which in the case of a time series of the price of an asset corresponds, to its returns.

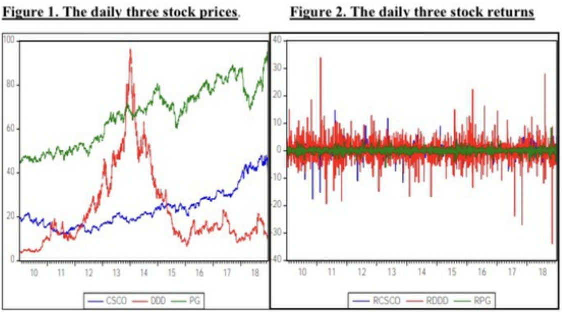


Figure 7: Example Price and Returns stationarity

Stationarity by Fractional Differentiation

Through **Fractional Differentiation**, proposed by Hosking (1981), is it possible to apply a differencing with a degree of differencing $d < 1$ by exploiting the lag operator. In this way is possible to obtain a stationary time series, without completely removing memory.

Values $\widetilde{X}_t$ of the transformed time series are defines as:

$$\widetilde{X}_t = \sum_{k=0}^{\infty} \omega_k X_{t-k}$$

$$\omega_k = -\omega_{k-1} \frac{d-k+1}{k}$$

Time-Series Data to 2D Images with Recurrence Plot

Recurrence Plot

A **Recurrence Plot** is a square matrix $RP_{i,j}$ where the values of $RP(i,j)$ are equal to 1 if the distance between the points $x(t_i)$ and $x(t_j)$ in phase space do not exceed a predefined value ϵ, otherwise $RP(i,j)$ is equal to 0.

$$RP(i,j) = \begin{cases} 1 & \|\overrightarrow{x_i} - \overrightarrow{x_j}\| \leq \epsilon \\ 0 & \|\overrightarrow{x_i} - \overrightarrow{x_j}\| > \epsilon \end{cases} \text{ for } i,j = 1,2...N$$

Recurrence plots can be represented as images, usually we assign to the values $RP(i,j) = 1$ the color black, while to the values $RP(i,j) = 0$ the color white.

Time-Series Data to 2D Images with Recurrence Plot

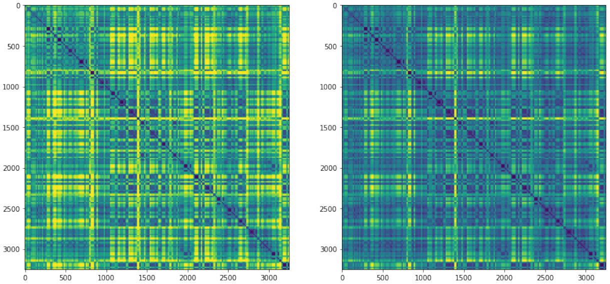


Figure 8: Left: joint recurrence plot of all days when it was convenient to buy bitcoin — Right: joint recurrence plot of all days when it was convenient to sell bitcoin.

Convolutional Neural Networks

The proposed model is an architecture based on a 2D **Convolutional Neural Network** (CNN), first introduced by Yann LeCun (1999).

Convolutional Neural Networks

Convolutional Neural Networks are specialized Neural Networks (NN) for processing data that have a grid-like topology, such as images (2D or 3D tensors). CNN employs a mathematical operation called convolution, which is a linear operation used in place of the general matrix multiplication used on standard NN

Convolutional Neural Networks

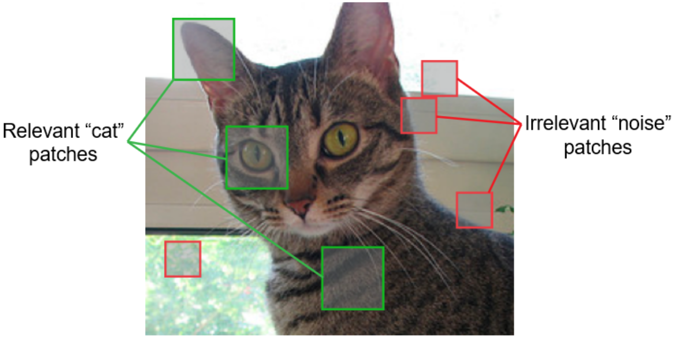


Figure 9: CNN are extremely efficient in finding local patterns anywhere in the image

Architecture of the proposed CNN Model

The proposed CNN model is composed of a two-stage structure for a total of 31,517,762 parameters. During each stage there is a feature learning phase using convolution, normalization, activation and pooling layers.

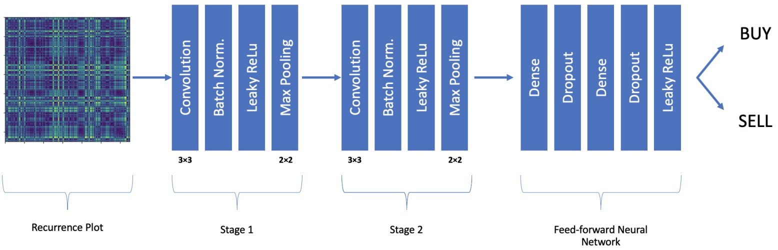


Figure 10: Architecture of the proposed CNN Model

Architecture of the proposed CNN Model

Layer (type)	Output Shape	Param
Conv 2D	(None, 254, 254, 32)	320
Batch Normalization	(None, 254, 254, 32)	128
Leaky ReLu	(None, 254, 254, 32)	0
Max Pooling 2D	(None, 127, 127, 32)	0
Conv 2D	(None, 125, 125, 64)	18496
Batch Normalization	(None, 125, 125, 64)	256
Leaky ReLu	(None, 125, 125, 64)	0
Max Pooling 2D	(None, 62, 62, 64)	0
Flatten	(None, 246016)	0
Dense	(None, 128)	31490176
Dense	(None, 64)	8256
Leaky ReLu	(None, 64)	0
Dense	(None, 2)	130

Table 2: Proposed CNN model Summary

Performance of the Proposed CNN Model

The proposed CNN model is trained for 100 epochs. Then, the trained model is tested on the testing dataset, corresponding to the days from May 27, 2020 to May 27, 2021.

To evaluate the performance of the model, we use **the macro average F1-score**, which is defined as the harmonic mean of the precision and recall.

$$\text{F1-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

The proposed CNN model shows promising results, achieving a Macro F1-score of 58% and outperforming the results obtained from various researches analyzed in the literature.

Financial Evaluation

Strategy	Total Return	Sharpe Ratio
Buy-And-Hold	329.63%	0.1291
Proposed CNN model Based	337.51%	0.1793

Table 3: Buy-and-Hold vs. Proposed CNN model based algorithm

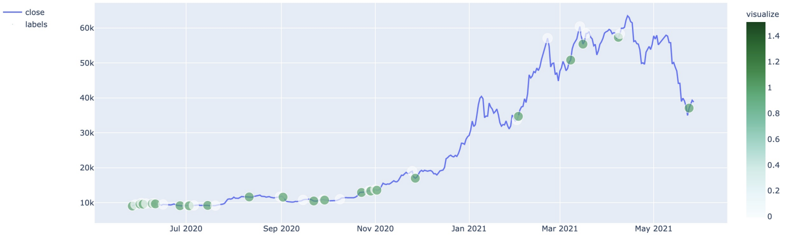


Figure 11: Proposed CNN model based algorithm - Predicted Labels

Other examined Machine Learning models

To compare the performance of the proposed CNN models, I also tested the performance of other machine learning models: the k-Nearest Neighbors (k-NN) and the Random Forest (RF) classifier.

ML Model	Macro F1-Score
k-NN	0.63
RF Classifier	0.48

Table 4: Macro F1-Score — Other ML Models

Strategy	Total Return	Sharpe Ratio
Buy-And-Hold	329.63%	0.1291
k-NN Based	405.31%	0.1947
RF Based	51.90%	0.0635

Table 5: Buy-and-Hold vs. Other ML Models

Conclusion

The proposed CNN model shows promising results, achieving a Macro F1-score of 58% and outperforming the results obtained from various researches analyzed in the literature.

It should be noted that the performance obtained by the k-NN surpasses those of the proposed model significantly, reaching a Macro F1-score of 63%.

It would be interesting to examine what are the reasons that lead to this increase in performance. The proposed model can be used in a trading algorithm.