Ringrazio mamma e papà per avermi permesso di e spronato a studiare. Ringrazio anche zia Gili e tutta la famiglia per il costante supporto e incentivo.

Abstract

With a total market capitalization that almost reached the peak of 3 trillion dollars, the Crypto currency market has attracted a lot of investors from all over the world. A peculiar feature of such a market is that the its volatility is something that is difficult to find on a regular basis in others asset classes. Also, a second peculiar feature of this market is that the entry barriers are very thin: just think that in some countries it is more easy (or convenient) to have a crypto wallet than a bank account. These factors combined shows that the small investor, that usually operates with a buy and hold strategy, can hurt themselves due to the massive volatility of this market. This is the reason that brought me to the purpose of this research, which is to investigate a market neutral strategy in order to formulate and compare different approaches with the goal of formulating investment strategies in the crypto market that are able to minimize the systematic risk. In order to do that, theories and models from time series analysis and machine learning will be implied. The results have shown that some of the proposed approach have proven themselves to be good, at the same time other and simpler approaches has proven to be bad investment strategies. The machine learning problems formulated in the research produced models that can actually provide the investor with a real hedge over the market. The results also have shown that the mean reversion market neutral strategy have better results in moments in which the market is not in a bull run environment. This is not a bad news since most trend following strategies have great results when markets are in bull runs or in euphoria states.

Contents

Chapter 1

Introduction

1.1 Cryptocurrency markets and systematic risk

Wikipedia provides the following definition of cryptocurrency:

"A cryptocurrency, crypto-currency, or crypto is a digital currency designed to work as a medium of exchange through a computer network that is not reliant on any central authority, such as a government or bank, to uphold or maintain it." [14]

The decentralized structure is the main characteristic of blockchain, and the applications built on top of blockchain have intrinsic characteristics of security and privacy that no central authority can provide. The potential of the blockchain in terms of applications to various fields is huge and goes from the gaming industry to insurances and banking services. However, as of today, the main use case that we see blockchain actually used is investment in Bitcoin and various altcoins[1]

The field of investing itself is a huge topic in the crypto space because there are multiple opportunities offered by the several decentralized finance (DEFI) protocols that never existed before, and if they did they were issued by centralized authorities and they offered a return on investment much smaller. Some of those opportunities are:

- Liquidity Provision
- Farming
- Staking
- Mining
- Lending

[1] altcoins is a term used to refer to every cryptocurrency that is not bitcoin

And the list goes on. However, the way in which most investors approach this market is the "buy and hold" strategy, meaning buy a crypto asset -that can be any cryptocurrency- hoping that in the future its dollar price will be higher.

Just like every other asset class, crypto market has its own inherent risk, and even if one builds a portfolio of crypto assets very well diversified, one could never protect itself with a buy and hold strategy from the systematic risk. Systematic risk refers to the risk inherent to the entire market or market segment; it is also known as "undiversifiable risk," "volatility" or "market risk," and it affects the overall market. [6]

Just like the average investor, in this research it won't be taken into account the investment possibilities offered by the DEFI, instead the goal will be to present and investigate a statistical arbitrage strategy in order to minimize the systematic risk when investing in the cryptocurrency market.

1.2 Cryptocurrencies futures contract

In finance, futures contracts (sometimes called futures) are standardized legal agreements between parties who do not know each other to buy or sell something at a given price at a particular time in the future [1]. The pre-agreed price at which a party buys or sells an asset is called the forward price. The specified time in the future, that is, the time of delivery and payment, is called the delivery date. Futures contracts are derivative products because this is a function of the underlying asset. The underlying financial instrument of a forward or futures contract can be any asset, such as equity, a commodity, a currency, an interest payment, cryptocurrencies or bonds.

Futures contracts are instruments built to serve specific purposes:

- **Hedging and Risk Management**: for example, a company may sell futures contracts for their products to ensure they get a certain price in the future, despite unfavorable events and market fluctuations.
- **Leverage**: a mechanism investors can use to increase their exposure to the market by allowing them to pay less than the full amount of the investment
- **Short Exposure**: Futures contract allow investors to take short exposure to an asset.

An important feature of traditional futures contracts is the expiration date. When the contract expires, a process called settlement begins. Traditional futures contracts are typically settled monthly or quarterly. At the time of settlement, the contract price will converge to the spot price and all open positions will be forfeited. In traditional futures,

when the contract expires, the buyer gets delivered the actual goods underlying the contract (e.g. Oil barrels)

Perpetual contracts, as a difference from traditional futures contracts, never expires. Investors can hold positions with no expiration date and do not need to track different delivery dates. As a result, trading cryptocurrencies perpetual contracts is very similar to trading pairs in the spot market. Because of the fact that they are not settled in the traditional sense, exchanges need a mechanism to ensure that perpetual futures contracts and the underneath asset prices converge on a regular basis. This mechanism is also known as the funding rate, and it consists in periodic payments either to traders that are long or short based on the difference between perpetual contract markets and spot prices. Therefore, depending on open positions, traders will either pay or receive funding.

Besides all the pro of the futures instrument, it must be declared that these contracts are also very risky. In order to prevent investors from having a negative equity, crypto exchanges offering futures adopts a mechanism called Liquidation. A liquidation consist in force-exit a losing position without the consesus of the trader to prevent traders from falling into negative equity. With this mechanism exchanges can protect themselves from reckless investors.

1.3 The rationale of Pairs Trading

As mentioned in section 1.1, this research will be focusing on a statistical arbitrage strategy, in particular the focus will be on pair trading.

Pair trading is a trading strategy that allows the investor to limit systematic risk in the sense that it does not need market directionality in order to function. This strategy falls into the category of strategies called statistical arbitrage, this means that it uses a statistical relationship observed in the past to decide future trades. The simplest approach is to find two assets belonging to the same asset class that have moved in a very similar way in the past and monitor their mutual movement, when the two assets diverge more than a predetermined value then it is time to buy the one that had a bearish movement and sell short the one that had the bullish movement.

Using this strategy we have two contrasting positions at the same time and in two assets belonging to the same asset class. What this strategy is based on is the relationship between the two (or more) assets, not the directionality of the market, that is why it is called a "market neutral" strategy and is the optimal way to minimize systematic risk when investing in financial markets.

Chapter 2

Dataset

The dataset required for building and testing a trading strategy can be very difficult to obtain. Luckily, an upside of the crypto space, with respect to the stock market is that the market data is easier and cheaper to obtain. As a matter of fact all excanges offer a free API with wich is possible to query past market data.

The assets that were chosen for this research were all the available perpetual futures contracts on FTX exchange starting from january 1 2020, up to january 1 2022. The choice of the dataset was due to the following reasons

- Futures assets were selected because the proposed strategy need to have short selling positions
- Perpetual futures don't need to take into account the expiration of the contracts, which would require to change the underlying asset, making the data gathering very painful.
- It was chosen the exchange with most perpetual futures available. Also, one of the ones with the highest volumes.

2.1 Dataset Retrieval

All available market data between the start date and the end date were pulled from the exchange. The format of the data is the canonical format for financial data: Open, High, Low, Close, Volume, Time. The time frequency of the data were of 1 minute.

2.2 Data cleaning and resampling

Amongst the 167 assets price series, lot of data were missing, but here a distinction must be made. In some cases the missing data was an error, and the missing data points were

not more than five, in other cases the data were missing because the exchange listed[1] the asset after the starting date of the research, and the result was a hole of missing data from the starting date up to the listing date.

In the first case, the error was covered by replacing the missing data with an average of the first close data points available backword and onward. But since the actual research was carried out over datapoints of 1 Hour frequency Close prices, this was not really an issue. The whole dataset was indeed resampled in 1 hour time frequency.

The second case, as it is well explained in the section 4 of the research, was addressed by designing the testing framework in a way that the entire testing period were divided in several sub-periods, in each of which a basket of assets is chosen from the universe of all available assets. With such a design, the issue of missing data only translate in the universe of available asset growing larger as the times goes onward.

[1]"To List" an asset means make an asset tradable on an exchange

Chapter 3

Methodology

This section is meant to provide a dissertation about the mathematical principles that will be used in the next sections.

3.1 Stationarity

A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary.

More formally, There are two types of stationarity: weak stationarity or covariance stationarity, and stationarity.

For every generic random variable we can define its moments as:

$$Mean : E(Y_t) = \mu_t \tag{3.1}$$

$$Variance : E[(Y_t - \mu_t)^2] = \sigma_t^2 \tag{3.2}$$

$$Autocovariance : E[(Y_t - \mu_t)(Y_{t+j} - \mu_{t+j})] = \gamma_t(j) \tag{3.3}$$

$$Autocorrelation : \frac{\gamma_t(j)}{\sigma_t \sigma_{t+y}} = \rho_t(j) \tag{3.4}$$

The stochasting process who generated the time series is said to be covariance stationary if:

$$E(Y_t) = \mu \quad \forall t \tag{3.5}$$

$$E[(Y_t - \mu)(Y_{t+j} - \mu)] = \gamma(j) \quad \forall t \tag{3.6}$$

i.e. the first two moments do not depend on time. The this fact is very useful because it means that μ and $\gamma(j)$ can be estimated using the sample counterparts. From now on, whenever the word stationarity will be mentioned, it will refer to the concept of covariance stationarity.

How can one check wheter a time series generating process is stationary? Different ways to do that can be carried out.

As first thing one may want to check wheter equations (3.1) and (3.2) holds true, and if it is the case, than the process is considered to be stationary.

A second way to check for stationarity leverages the **Wold Decomposition Theorem**, which states that any zero-mean nondeterministic, regular, covariance-stationary process

$$y_t \quad ; \quad t \in Z$$

can be decomposed as:

$$y_t = \sum_{j=0}^{\infty} \psi_j \mu_{t-j} \qquad \mu_t \simeq WN(0, \sigma_\mu^2) \tag{3.7}$$

Where :

$$\psi_0 = 1 \quad ; \quad \sum_{j=1}^{\infty} \psi_j^2 < \infty \tag{3.8}$$

By representing the process as (3.7) and by checking that the (3.8) hold, we can determine whether a process is stationary or not.

A third way which applies to auto regressive processess would be to check the roots of the following AR polynomial are "outside the circle":

$$(1 + \theta_1 z + ... + \theta_q z^q) = 0 \tag{3.9}$$

The roots are the values of z that solve the equation (3.9). There are p of them, although some of them may be equal. Some of these roots may be complex numbers. If the

roots all are real numbers (that is, none of the roots are complex numbers), then we can say that y_t is stationary if the absolute values of all of these real roots are greater than one. If a root equals one or minus one, it is called a unit root. If there is at least one unit root, or if any root lies between plus and minus one, then the series is not stationary. About this fact it is worth mentioning that in ARMA models, the stationarity only depends on the autoregressive part.

In all these aforementioned cases we are able to check for the stationarity of a given model. However if we want to check for the stationarity of a data generating process that we do not know, we must adopt a statistical testing framework and use the only available thing that we have in order to determine its stationarity: a sample of realized data. More about that in forecoming chapters

Stationarity can be useful in a context in which we want to o understand the relationship between two variables using regression analysis: in such a case we need to assume some sort of stability over time Stationarity is a property that regards the joint distributions of a process as it moves through time.

However, stationarity by itself is not sufficient to use OLS. We need a property called Weakly dependence.

3.2 Weak Dependent time series

The concept of weak dependency in time series places restrictions on how strongly related the random variables x_t and x_{t+h} can be as h gets large. A stationary time series is weakly dependent if the correlation between x_t and x_{t+h} goes to zero "sufficiently quick" as $h \to \infty$. This means that as the variables get more distant in time, their correlation become smaller and smaller.

$$Cov(x_t, x_{t+h}) \to 0 \qquad h \to \infty \tag{3.10}$$

Weak dependence is important for regression analysis because it replaces the assumption of random sampling in implying that the law of large numbers (LLN) and the central limit theorem (CLT) hold. And as it will be shown in the next sections, regression analysis will play an important role in this research, and this is because the statistic tests that will be carried out are based on regression. Note that stationarity does not replace such assumptions, we specifically need weak dependence.

It is very typical that time series are not weakly dependent, but exhibit strong dependence, or "high persistence". When the time series sequence is persistent, our assumptions for OLS are misleading.

Weakly Dependent processes are said to be integrated of order zero, or I(0). Such series can be used without transformation since can be used the standard OLS estimators.

Consider the following AR(1) process:

$$x_t = \rho x_{t-1} + \varepsilon_t \qquad i \sim iid(0, \sigma^2) \quad |\rho| < 1 \tag{3.11}$$

We can show that it is weakly dependent:

$$Corr(x_t, x_{t-1}) \sim \rho \tag{3.12}$$

$$x_t = \rho^2 x_{t-2} + \rho x_{t-1} + \varepsilon_t \tag{3.13}$$

$$Corr(x_t, x_{t-2}) \sim \rho^2 \tag{3.14}$$

then:

$$|Corr(x_t, x_{t-2})| < |Corr(x_t, x_{t-1})| \tag{3.15}$$

We can therefore see that the auto correlation of the process (3.11) grows smaller at exponential rate with the lags. This fact however does not hold true when $\rho = 1$. In this case we have that the model is a random walk:

$$x_t = x_{t-1} + \varepsilon_t \quad , \quad i \sim iid(0, \sigma^2) \quad |\rho| = 1 \tag{3.16}$$

$$subsituting: \quad x_{t-1} = x_{t-2} + \varepsilon_{t-1} + \varepsilon_t \tag{3.17}$$

$$... \tag{3.18}$$

$$x_t = x_0 + \sum_{i=0}^{t-1} \varepsilon_{t-i} \tag{3.19}$$

$$Var(x_t) = \sum_{i=0}^{t-1} Var\varepsilon_{t-i} = t\sigma^2 \tag{3.20}$$

We have that the variance is a function of time, therefore we now know that the process is not stationary, but recall that we are interested in dependence:

$$Cov(x_t, x_{t+h}) = Cov(x_t, x_t + \sum_{i=0}^{h-1} \varepsilon_{t+h-i}) = Var(x_t) = t\sigma^2 \tag{3.21}$$

$$Corr(x_t, x_{t+h}) = \frac{Cov(x_t, x_{t+h})}{\sqrt{Var(x_t) \times Var(x_{t+h})}} = \frac{t\sigma^2}{\sqrt{t\sigma^2 \times (t+h)\sigma^2}} = \frac{t}{\sqrt{t(t+h)}} \tag{3.22}$$

$$Corr(x_t, x_{t+h}) = \sqrt{\frac{t}{t+h}} \tag{3.23}$$

For a fixed t, the correlation (3.23) goes to zero as h goes to ∞, however we can always choose a t arbitrarily large s.t. the correlation does not goes to zero. Therefore, we can not say that the correlation goes to zero "fast enough" in this case, therefore the process can't be considered weakly dependent.

3.3 Unit Roots and Orders of integration

Weakly dependent time series are integrated of order zero I(0). If a time series has to be differences one time in order to obtain a weakly dependent series, it is called integrated of order one I(1)

If a process is said to "have a unit root" I(1) it means that one of the roots of its AR part (3.9) is equal to 1 (unit), while the others are inside the unit circle. For every I(1) process, its first differences are I(0), meaning that the resulting process does not have unit root. Furthermore, like stated by Engle and Granger (1987): a series with no deterministic component which has a stationary, invertible ARMA representation after differencing d times is said to be integrated of order d.

According to the above statements, any stationary and invertible ARMA model is also an I(0) process. Note that I(0) implies stationarity, but the inverse is not true, follows an example.

The process:

$$y_t = \varepsilon_t - \varepsilon_(t-1) \qquad Where \quad \varepsilon_t \sim iit, (0, \sigma^2) \tag{3.24}$$

is stationary, but not invertible. It is infact I(-1)

We are interested at this point in determine whether or not a time series is weakly dependent - I(0) instead of stationary.

3.3.1 Dickey Fuller test

As mentioned before, in the real world we only have the data which are the realizations of the process in which we are interested. Therefore we must leverage the statistical test theory to determine with certain levels of confidence whether the data generating process is I(0). To this purpose the Dickey Fuller test is very useful.

Considering the model:

$$Y_t = \rho Y_{t-1} + \epsilon_t \tag{3.25}$$

Where

$$\varepsilon_t \sim iit, (0, \sigma^2); \quad |\rho| < 1;$$

If $\rho = 1$, the unit root is present in a time series, and the time series is non I(0).
A regression model can be represented as:

$$\Delta Y_t = (\rho - 1) y_{t-1} + \epsilon_t \quad = \quad \delta y_{t-1} + \epsilon_t \tag{3.26}$$

Where Δ is the difference operator and $\delta = \rho - 1$. By estimating the model (3.12) one can test for the presence of a unit root with the null: (H_0 : unit root) by testing $\delta = 0$.

It has to be considered though that if the tested process has mean different from zero, (3.11) will actually become:

$$Y_t = \alpha + \rho Y_{t-1} + \epsilon_t \tag{3.27}$$

Where

$$\varepsilon_t \sim iit, (0, \sigma^2); \quad |\rho| < 1; \quad \alpha \neq 0$$

If α is omitted while estimating, and actually $\alpha \neq 0$, then $\hat{\rho}$ won't be a consistent estimator. There are actually four different cases of the DF test to account for issues of this kind:

- **Case 1:** The estimated process does not contain a constant or a time trend and the true process is a random walk.
- **Case 2:** The estimated process contains a constant but not a time trend and the true process is a random walk
- **Case 3:** The estimated process contains a constant but not a time trend and the true process is a random walk with drift
- **Case 4:** The estimated process contains a constant and a time trend and the true process is a random walk with drift

Each cases has its own $T(\hat{\rho} - 1)$ and t-statistics values which are both valid tests and are based on the size of the sample, this specific distribution is known as Dickey-Fuller table. t-Student critical values can't be used because the test is based on residuals.

3.3.2 Augmented DF test

The ADF test builds on top of what the DF fuller test does, it still consist in a test of the null: (H_0 : unit root) in the time series; the difference with the DF test is that ADF takes into account larger and more complicated ARMA(p,q) models when testing, whereas classic DF test is only adequate for time series well represented by an AR(1) plus noise models.

The following is the model used for testing:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + ... + \delta_{p-1} \Delta y_{t-p+1} + \epsilon_t \tag{3.28}$$

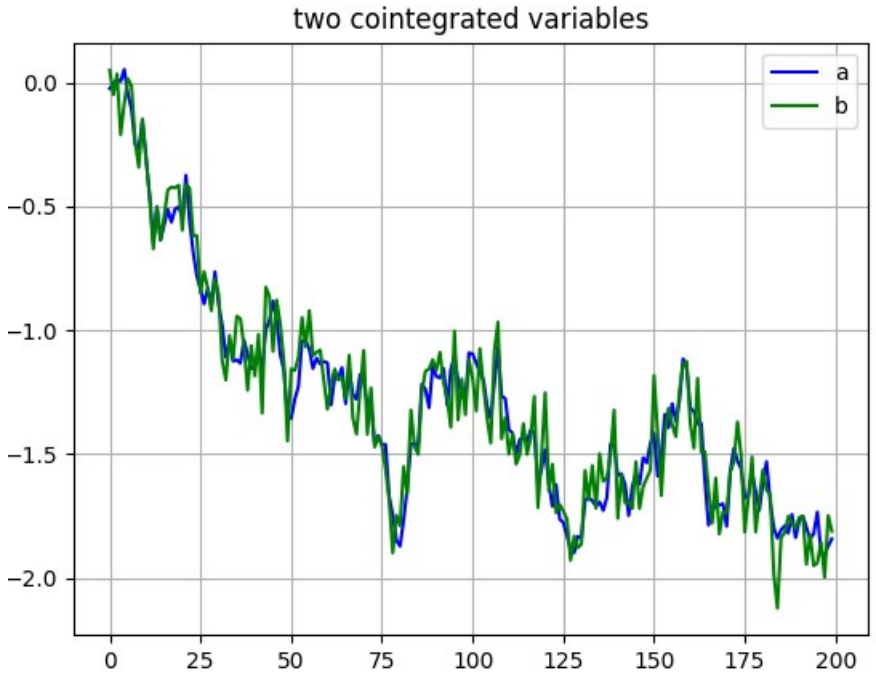
Where α is constant, β is the coefficient of a time trend, p is the lag order of an autoregressive process. By setting the parameters $\alpha = 0$ or-and $\beta = 0$, there can be formulated four different versions of the test, just like it happens in the classic DF test. By the fact that it is possible to include lags of order $p > 1$ it is clear that p must be determined when applying the test. There are two different approaches in determine p: one is to start from high orders and examine the t-values on coefficients while testing by a lower p. Another one require to perform order selection techniques. The unit root test is then carried out under the null hypothesis $\gamma = 0$ against the alternative hypothesis of $\gamma < 0$. The test takes the following value:

$$DF_\tau = \frac{\hat{\gamma}}{SE(\hat{\gamma})} \tag{3.29}$$

As this test is asymmetrical, we are only concerned with negative values of our test statistic (3.15). If the calculated test statistic is < than the critical value for the "classic" DF test, then the H_0 is rejected and no unit root is present. The more negative the value of the test, the stronger the rejection of the hypothesis that there is a unit root at the specific lelvel of confidence

3.4 Cointegration

A peculiar feature of most economic time series is the presence of a unit root $y_t \sim I(1)$ and $\Delta y_t \sim I(0)$. This kind of time series has very unpleasant characteristics to analyze. However, it is possible that between two processes $\{y_t\}_{t=-\infty}^{\infty}$ and $\{k_t\}_{t=-\infty}^{\infty}$ exists also a linear combination which is I(0). This is basically the concept of two cointegrated time series. In other words, saying that $\{y_t\}_{t=-\infty}^{\infty}$ and $\{k_t\}_{t=-\infty}^{\infty}$ are cointegrated means that they have a common stochastic trend and that the common trend can be eliminated by taking a specific difference of the series such that the resulting series is I(0). Note that two variables can be uncorrelated whilst they are cointegrated.



In the image we have:

$$a_t = \sum_{t=0}^{n} \varepsilon_t \qquad Where \quad \varepsilon_t \sim iid, (0, \sigma^2) \tag{3.30}$$

which is a Gaussian random walk, and we have:

$$b_t = a_t + \varepsilon_t \qquad Where \quad \varepsilon_t \sim iid, (0, \sigma^2) \tag{3.31}$$

Since their linear combination with a simple difference coefficient of -1 is Gaussian noise, we know for sure that these two variables are cointegrated, and that is because Gaussian noise is I(0).

It can be that two series of integration order $c > 1$ happen to be cointegrated, resulting in a linear combination of order f s.t. $c > f > 0$, but since it never happens in economics, it does not need further explanation.

Also, it is possible that a linear combination of a set of time series results in I(0). This result consist in a stationary portfolio composed by > 2 time series, and the same techniques applied in this research on the 2 time series case could be applied in a portfolio case.

3.4.1 Engle Granger Test

Testing for cointegration means testing for the existence of long-run equilibria among the elements of one or more time series. The Engle-Granger(1987) methodology is a residual-based testing method which only consider the case with at most one cointegrating vector, meaning two cointegrated variables x_t and y_t. It can be divided into three steps:

The first step consists in testing singularly x_t and y_t for their order of integration. The two must have the same order of integration in order to be cointegrated. For this purpose, the Augmented Dickey Fuller test can be used. If the null hypothesis of existance of a unir root is accepted for both x_t and y_t, than one may proceed to the second step.

In the second step is performed the estimation of the long run equilibrium relationship between the time series using OLS.

$$y_t = \mu + \beta x_t + \epsilon_t \tag{3.32}$$

Engle and Granger (1987) demonstrate that if two variables are cointegrated, then OLS estimator produce "super-consistent" estimate of the cointegrating vector, which means that the OLS estimate $\hat{\beta}$ converges to the true value β faster than it would if the series were stationary. The parameters are estimated from:

$$\hat{\beta} = \frac{\sum(x_t - \overline{x}_t)(y_t - \overline{y}_t)}{\sum(x_t - \overline{x}_t)^2} \qquad \hat{\mu} = \overline{y}_t - \hat{\beta}\overline{x}_t \tag{3.33}$$

Where $\overline{x}_t$ and $\overline{y}_t$ are the means of x_t and y_t. The estimated regression is given by:

$$\hat{y}_t = \hat{mu} + \hat{\beta} x_t \tag{3.34}$$

And the residuals:

$$\hat{\epsilon_t} = y_t - \hat{y}_t \tag{3.35}$$

The third step coinsist in a unit root test (e.g. the ADF test) in order to check whether the residuals I(0). If the residuals are I(0), then x_t and y_t are cointegrated, otherwise they are not.

3.5 Mean Reverting processes

3.5.1 Hurst Exponent

The Hurst index is called the "Dependency Index" or "Long Distance Dependency Index". It quantifies the relative tendency of the time series to drop sharply towards the average or cluster in one direction. It was first mentioned by Harold Edwin ia a paper published in 1951. Hurst was an hydrologist and he was looking for a way to model the levels of the river Nile. His studies turned out to be very useful in several other fields, such as finance.

The Hurst exponent H is defined in terms of the rescaled range of a time series:

$$E\left[\frac{R(n)}{S(n)}\right] = Cn^H \qquad as \quad n \to \infty \tag{3.36}$$

(3.36) is derived as for a time series $X = X_1, X_2, ..., X_n$ as follows:
The mean is calculated:

$$m = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.37}$$

The series is transformed in a mean-adjusted series:

$$Y_t = X_t - m \qquad fort = 1, 2, ..., n \tag{3.38}$$

It is calculate the cumulative deviate series Z:

$$Z_t = \sum_{i=1}^{t} Y_i \qquad fort = 1, 2, ..., n \tag{3.39}$$

For Z, is calculate a range series R:

$$R_t = max(Z_1, Z_2, ..., Z_t) - min(Z_1, Z_2, ..., Z_t) \qquad fort = 1, 2, ..., n \tag{3.40}$$

A standard deviation series S:

$$S_t = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(X_i - \mu)^2} \qquad fort = 1, 2, ..., n \tag{3.41}$$

At this point the rescaled range can be calculated:

$$(R/S)_t = \frac{R_t}{S_t} \qquad fort = 1, 2, ..., n \tag{3.42}$$

Once we obtain the rescaled range, one can estimate the H. exponent by fitting the power-law (3.36) to the data. This is be obtained by taking the logarithm of both sides and running a linear regression: the angular coefficient of the linear model will be the Hurst exponent.

Once the Hurst Exponent is calculated, its values can be interpreted as follows

- **H < 0.5:** is a negatively dependent series,
- **H = 0.5:** is a weakly dependent process.
- **H > 0.5 :** denotes a series with strong positive correlation

(For any H in (0,1) the process is mean reverting)

By the way, ok you can use this R/S for the thesis, but for the future you should augment it by a correction for the autocorrelation).

any H in (0,1) gives a stationary process; however, only H=0.5 is weak dependent; H in (0.5,1) has strong the dependence, so it is not weak dependent).

(By the way, if you call H the Hurst exponent and d our order of integration, then an I(0) process has d=0, an I(1) process has d=1; in general it holds that H=d+1/2).

3.5.2 Ornstein–Uhlenbeck Model

Ornstein Uhlenbeck process is a mean-reverting process that can be used to model mean reverting time series processes [8]. By following the work of [13] such a process can be described by the following stochasting differential equation:

$$dX_t = \alpha(\mu - Xt)dt + \sigma dW_t \tag{3.43}$$

Where

$$\alpha > 0; \qquad W_t \quad is \quad Wiener \quad process$$

It can be turned explicit as follows:

$$X_t = e^{-\alpha t}X_0 + \mu(1 - e^{-\alpha t}) + \int_0^t \sigma e^{\alpha(s-t)}dW_s \tag{3.44}$$

Then:

$$EX_t = e^{-\alpha t}X_0 + \mu(1 - e^{-\alpha t}) = \mu + o(1) \qquad as \quad t \to \infty \tag{3.45}$$

$$VarX_t = \frac{\sigma^2}{2\sigma}(1 - e^{-2\alpha t}) = \frac{\sigma^2}{2\sigma} \qquad as \quad t \to \infty \tag{3.46}$$

The coefficient α is called "speed of mean reversion" and $t_{\frac{1}{2}}$ is called "half-life of mean reversion" and it can be view as the average time that will take the process to back half-way to the average value. It can be calculated with the following:

$$t_{\frac{1}{2}} = \frac{log2}{\alpha} \tag{3.47}$$

The value of the equation (3.47) is used by [3] in order to determine whether a financial time series is adequate for the application of mean reverting strategies.

3.6 Time Varying parameter estimation

The idea behind this section is that the phenomena that we are observing gradually change over time, therefore the parameters used by our models should change in order to adapt to the most recent version of the phenomena. At this regard, the Kalman filters algorithm is used in a methodology called flexible least square in order to estimate through time the parameters of a linear regression.

3.6.1 Kalman filters

Kalman filter is a linear algorithm that updated the expected value of a hidden variable based on the latest value of an observable variable. The assumption is that the observable variable is a linear function of the hidden variable, with noise. Also, it assumes that the hidden variable at time t is a linear function of itself at t-1, with noise. The algorithm is composed by two steps: a prediction step and a correction step. The Kalman filter considers the errors to be gaussian distributions with 0 mean, thus the predictions for the new state given the measurement and the state model will also be gaussian distributions, but with different mean and variance. The algorithm has a weighting principle in the sense that it updates the state putting more "weight" to the distribution that has smaller variance.

In our case we are interested in the estimation of the linear regression parameters between two prices that we think cointegrated, in order to do that we use the state space model proposed by [5]. The regression coefficient at time $t+1$ is modeled as a noisy version of the previous coefficient at time t . First, we introduce a random vector ω_t with zero mean and some covariance matrix V_ω, so that

$$\beta_{t+1} = \beta_t + \omega_t \tag{3.48}$$

We also introduce another random variable ε_t having zero mean and some variance V_ε so that:

$$t_t = x_t'\beta_t + \varepsilon_t \tag{3.49}$$

Equations (3.48) and (3.49) jointly considered result in a linear state-space model, for which it is assumed that the innovations series ε and ω_t are mutually and individually uncorrelated

In order to run the Kalman Filter algorithm by using the defined state space modeol, we need to encode the variables in the following way:

- Vector of observable variables

- Vector of hidden variables
- Matrix of state-transition model
- Matrix of observation model

the observed variable is one of our price series, and the hidden variables are the parameters β of the linear regression. The observed and hidden variables are related by the familiar spread equation. The other price series provides our observation model. Finally, the state-transition model which should describe the evolution of the hidden variables from one state to the other will be defined by the equation (3.48).

Once the variables are all defined, the algorithm must also be initiated with initial values for the covariances of the measurement and state equations, those values measures how fast the estimated parameters can change at each iteration.

3.7 Machine Learning for time series forecasting

The term machine learning refers to the automated detection of meaningful patterns in data [10]. Machine learning is a very active field in the research topic and its applications range is very broad. Inside its applications there are also several time series forecasting cases [7]

3.7.1 Neural networks

An artificial neural network is a general purpose model whose internal structure is composed by interconnected and autonomous processing units called artificial neuron, from now on simply "neurons", each of which is placed inside a node. Each neuron implements a parametric mathematical function looking as follows:

$$y = \Gamma(\sum_{i=1}^{n} \omega_i x_i - b) = \Gamma(\sum_{i=0}^{n} \omega_i x_i) \tag{3.50}$$

Where Γ is called activation function, also called transfer function. x_i is the input of the neuron, which based on the position of the neuron in the network, can be either the raw feature of a data point or an output of another neuron. Each node in a neural network is grouped in layers, and the node of a layer are connected to the nodes of the previous or subsequent layer through edges. Every edge has its ω_i parameter that is called "weight", that is the second component of the dot product serving as input of the activation function.

A visual representation of a neuron is fig 3.1

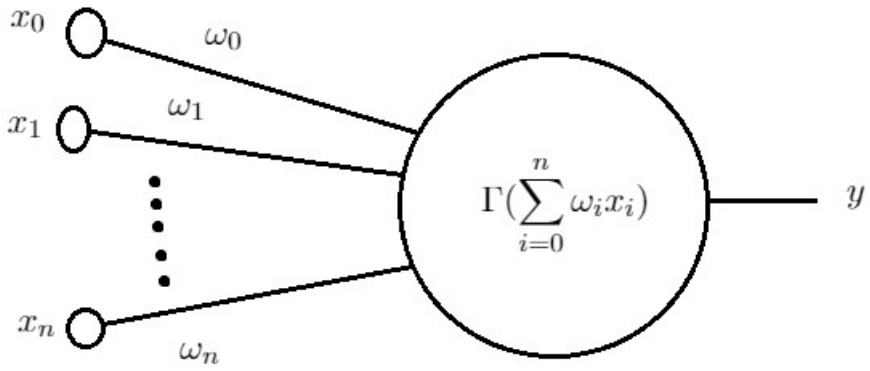


Figure 3.1: Neuron

Based on the scope of the model and the position of the neuron in the network, different activation function can be employed. The main characteristic of these functions is that they are strictly monotone increasing, which means that the derivative is nonzero everywhere. This is important since the training phase -discussed later- leverage a gradient based optimization process.

All nodes have an activation function, nevertheless the activation function used in the output layer determine whether the neural network model is a regression model or a classification model, a further explanation about this matter can be found in the sections 3.7.2 and 3.7.3

A very common actvation function is the Hyperbolic tangent

$$\Gamma(p) = tahn(p) = \frac{e^p - e^{-p}}{e^p + e^{-p}} = \frac{2}{1 + e^{-2p}} - 1 \tag{3.51}$$

A visual representation of this function is shown in fig 3.2

A series of nodes stacked upon each others create a layer, every node of one layer is connected to all the nodes of the previous layer through an edge, which carries the weight parameter ω. A neural network is composed by layers, that can be classified in three categories:

- **Input Layer:**: It has as many nodes as the dimensionality of the feature space
- **Hidden Layers:**: Are the layers that lays between the input layer and the output layer. The number of such layers and the number of nodes inside a layer are hyperparameters to be determined when the network is designed

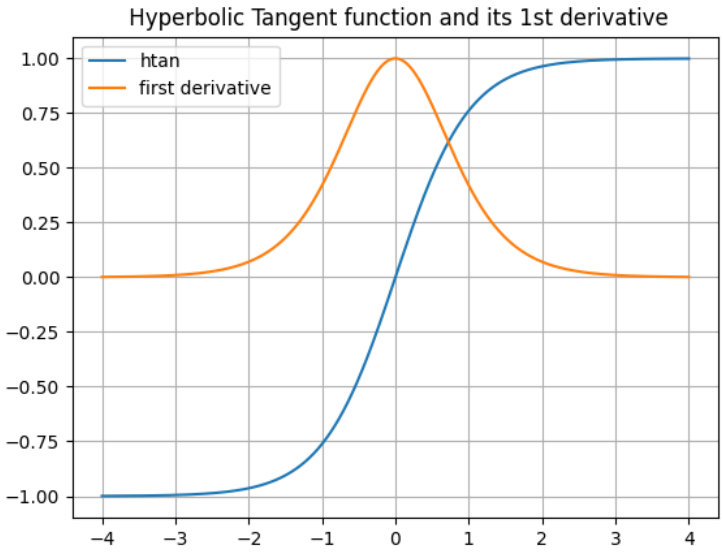
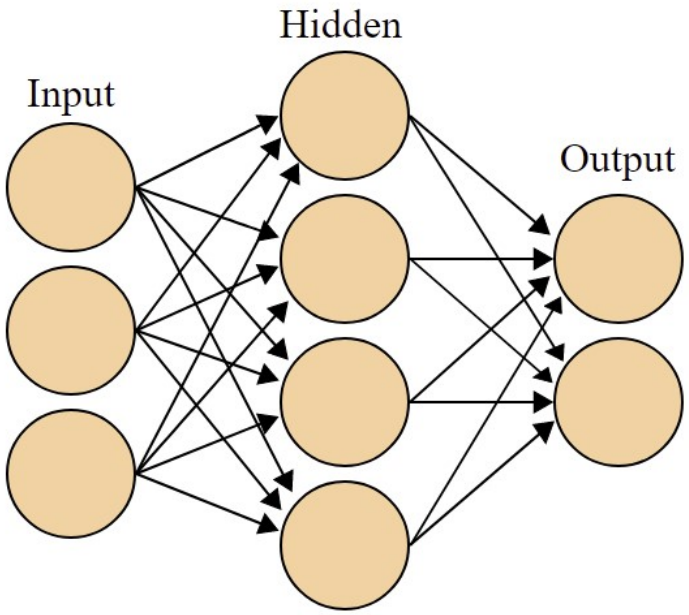


Figure 3.2: Hyperbolic tangent

- **Output Layer:**: It is the last layer of the network, its output represents the prediction $\hat{y}$. The activation function is chosen based on the task.



Training Phase

As it happen for all supervised machine learning models, the process/phenomena that one need to model has to be encoded in a feature-label dataset, and it has to be divided in two parts: a training set and a test set. The train set is the portion of data used by the learning algorithm to tune the model in order to try learning the underlying data generating process, whereas the test set is the portion of data in which an assessment is made on the capability of the model to correctly forecast the labels. In the case in which the learning algorithm presents also some hyperparameters - as it happens for neural networks- the train set is sub-divided in train set and validation set, in order to tune the hyperparameters without looking at the test set.

Neural networks are parametric models, meaning that the number of internal of the model that we want to find is fixed. These values are estimated trough the learning algorithm execution over the train data. Such parameters in a neural network are the weights, and the training phase goal is to find the values of such parameters which minimize the loss function.

Neural networks algorithms has also a series of hyperparameters that must be set in the proper way. These hyperparameters do not belong to the model but belong to the learning algorithm which trains the model. They must be set before the train start and there are techniques to tune such values in the proper manner. Such hyperparameters are the learning rate, the number of epochs and the batch size. The goal of machine learning is to find a model perfoming a function which correctly maps feature datapoints into labels datapoints, also known as predictor

How does an machine learning algorithm produces predictors? What a learning algorithm usually does is iterate over the train set trying to find the values of the model's parameters which minimize a given loss function in the prediction of the labels. A good predictor has been found if the average loss function computed over the train set (train error) can be succesfully minimized, and the average loss function computed over the test set (test error) returns a value in the whereabouts of the train error.

The backpropagation is the algorithms works in sych a way that comunicates backwards on the network starting from teh output node how a single training datapoint would like to nudges the weights, in order to have the most rapid decrease of the loss function. Backpropagation algorithm is carried out by knowing the label of the specified example in the training set.

A gradient descent step, would mean to perform backpropagation for all the training set and average the desired changes that you get. This would be very slow, especially for large datasets; what stochastic gradient descent does instead is to separate the training set in the above mentioned batches, which are chunks of training data, and perform the gradient

descent step for each of these batches, with the result of speeding up the training at a much lower computational cost. The magnitude of the changes in the weights performed at every step is the aforementioned learning rate.

By using Stochastic gradient descent instead of regular gradient descent, the algorithm takes the weight values toward local minima instead of the true global minimum, besides that, the results are empirically proven to be good.

One last hyperparameter is the number of epochs: that is the number of times that the learning algorithm goes through the entire train set, allowing all the train datapoints to be taken into consideration for the weights update

Different loss functions are used based on whether the model is built for a regression task or classification task. When it is the classification case, a common choice is the cross-entropy loss, or log loss. This function measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label.

When the classes are 2:

$$L = -(y \log(p) + (1 - y) \log(1 - p)) \tag{3.52}$$

Where y is the true label (either 0 or 1) and p is the predicted probability of belonging to one of the classes.

When the classes are M $>$ 2:

$$L = -\sum_{c=1}^{M} y_{0,c} \log(p_{0,c}) \tag{3.53}$$

Where M is the number of classes, y is a binary indicator (0 or 1) if class label c is the correct classification for observation o; p is the predicted probability observation o is of class c.

If the task is regression, a common choice for the loss function is the mean squared error:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} (y - \hat{y}_i)^2 \tag{3.54}$$

Where y is the true label, and $\hat{y}$ is the prediction of the model.

3.7.2 NNs for regression

If the machine learning problem is designed as a regression problem, the model will take as input a set of data points called features whose dimension is d: X^d, and the labels will consist of a real number y. If this is the case, the output layer should be composed by only one node and the activation function should be a simple linear function called linear activation:

$$y = x \tag{3.55}$$

3.7.3 NNs for classification

If the machine learning problem is designed as a classification problem, the model will take as input a set of data points called features whose dimension is d: X^d, and the labels will consist class. In a multiclassification problem, classes can be encoded as arrays in which every position represents a class, and the belonging to a class is determined with 1, whereas not belonging to a class is represented with 0.

If the problem is built as a multiclassification problem with N classes, the output layer should be composed by only one node and the activation function should be a softmax activation function:

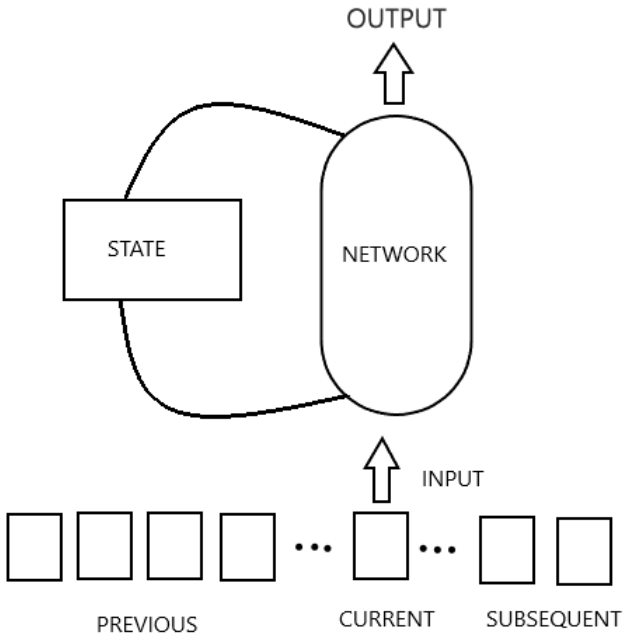
$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{N} e^{z_N}} \qquad \forall j = 1, ..., N. \tag{3.56}$$

The output will be an array whose values can be interpreted as the probability of a class membership.

3.7.4 Recurrent neural networks

Neural networks described so far are said feedforward, that is a model that transforms an input into an output through mathematical functions with no memory of the previous input. Every prediction is separated and don't influence the others.

With recurrent neural networks was introduced the concept of "state" that can be seen as a piece of information that the retains from an input and take it into account in the next forecast. This concept allows the model to take into account not only the input, but also the context in which the input occurs. This characteristic turns out to be very useful in time series datasets.



3.7.5 Vanishing Gradient

Recurrent neural networks suffer from an issue called vanishing (and exploding) gradient, this is due to the fact that the backpropagation algorithm used to train neural networks is designed in a way that it tweaks all the parameters ω of the networks for every example in the train set as the following:

$$\omega_{new} = \omega_{old} - LearningRate \times \nabla L(\omega_{old}) \tag{3.57}$$

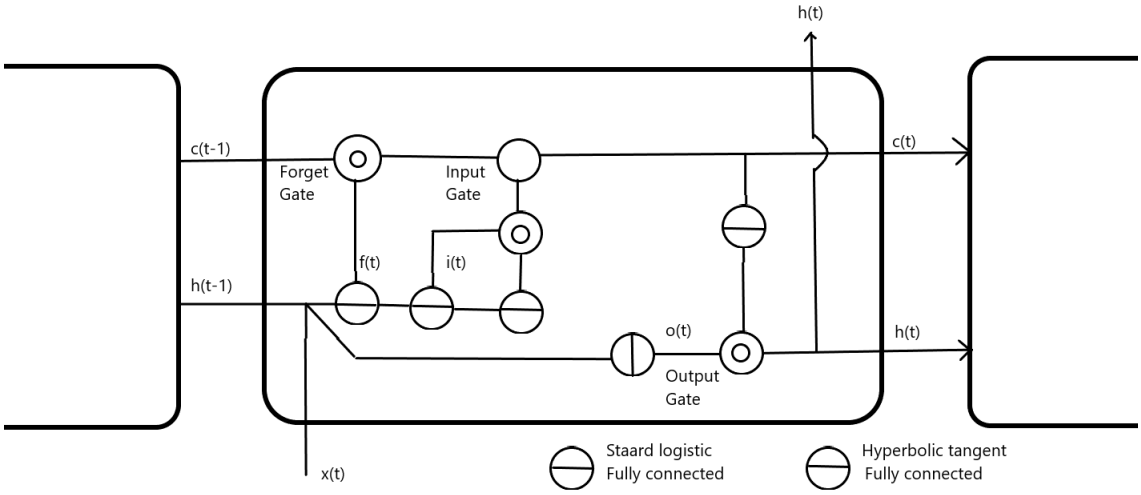
Where $\nabla L(\omega_{old})$ is the gradient of the loss function wrt the weights for a specific node.

The training phase in RNNs behave slightly differently compared to feed forward neural networks because the hidden layer of one observation is used to train the hidden layer of the next observation. Such a design make them subject to vanishing gradient issue because the cost function computed at a deep layer of the network will be used to change the weights of nodes at more shallow layers, and since the operation used to tweak the weights is multiplication, it turns out that the gradient calculated in a step that is deep in the neural network will be multiplied back through the weights earlier in the network, and this can cause the gradient to "vanish" through the network. Specifically, if the multiplying factor is small this phenomena is called vanishing gradient, causing that the gradient calculated at deep stage of the network to have a too small impact on the weight in the shallower

layer; on the other hand if it is big it is call exploding gradient, it will cause a too big impact on the weight in the shallower layer.

3.7.6 Long Short Term Memory

A version of recurrent neural network which does not present the vanishing gradient issue is the Long Short Term Memory (LSTM) networks, that have within the network a cell with the following structure:



Where:

- **x(t)**: is the input data
- **c(t-1)**: is the context of the previous instant (input)
- **h(t-1)**: is the output of the previous instant (input)
- **f(t)**: vector controlling the forget port
- **i(t)**: vector controlling the input port
- **o(t)**: vector controlling the output port

And there will be two outputs c(t) and h(t), respectively the context and output, that will be calculated respectively:

$$c_t = f_t \times C_{t-1} + i_t \times tahn(W_c \bullet [h_{t-1}, x_t] + b_C) \tag{3.58}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o \tag{3.59}$$

$$h_t = o_t \times tahn(C_t) \tag{3.60}$$

The internal structure of the LSTM cell reenact the context at each step, allowing for the decision of which information to let pass unchanged and which to modify based on the current context and current input. This feature allow the "long term" memory feature of the model [11]. Furthermore it interrupts the multiplicative effect of the gradient during the training phase, solving the issue of vanishing gradient.

3.8 Forecast Evaluations

The most intuitive function that can be used to assess the forecast of a model is the zero-one loss function

$$L(y, \hat{y}) = \begin{cases} 0 & if \quad y = \hat{y}; \\ 1 & if \quad y \neq \hat{y} \end{cases} \tag{3.61}$$

By applying this function to all forecasts of the model, it results in a count of how many labels the model got right, from that value can be calculated the accuracy metrics, which is the percentage of right predictions out of the overall number of predictions. This loss function is typically used for the evaluation of binary classification models. For regression models the typical evaluation metrics regard how far the forecasts are from the true label, like for example the Mean Squared Error; however, in order to make the interpretation of the results more understandable, and based on the fact that the regression model will be used as a buy/sell signal, the function (3.61) and the accuracy metrics will be used also in the regression case. Note that the evaluation will be constructed as follows:

$$L(y, \hat{y}) = \begin{cases} 0 & if \quad y \times \hat{y} > 0; \\ 1 & if \quad y \times \hat{y} < 0 \end{cases} \tag{3.62}$$

i.e. if the sign of the label is the same of the prediction, the loss is zero; otherwise the loss is 1.

3.9 Economic evaluation

In this section it will be discussed the methods of assessment of the experiments carried out in this research. Whereas the machine learning models have their own metrics of assessment, it only express a measure of the goodness of their predictions. Since in this research we are using such models as a building block in a more complex trading framework, we need some metrics to evaluate the goodness of the different trading strategies tested on the dataset, within such artificial trading framework.

Such metrics are well known and used a lot in the finance and trading industry, but it must be stated that the calculation of such metric - when it is performed on a simulated

artificial framework - heavily depends on the assumptions made when the framework was built. It is not wise then to compare the metrics of an artificial framework such as the one presented in this research, with the same metrics calculated on another framework - nor the metrics of the performance of an actual investment in the financial markets.

The only purpose of the calculation of the performance metrics is to compare among them the different approaches within the same framework

3.9.1 sharpe ratio

The index of Sharpe, also known as Sharpe Ratio, is a metric that measure the performance of an investment strategy taking into account the return and adjusting for the risk. It is defined as follows

$$S_a = \frac{E[R_a - R_b]}{\sigma_a} \tag{3.63}$$

Where R_a is the return of the investment that one want to assess; R_b is the "risk-free" return such as a treasury security, and it represents a benchmark. $R_a - R_b$ is then considered the expected value of the excess of the investment return over a risk-free investment. σ_a is the standard deviation of the excess of investment return and it is the component that measure the risk of the investment.

The value of the SR can also be annualized:

$$(yearly) \quad S_a = \frac{E[R_a - R_b]}{\sigma_a} \times (annualization \quad factor) \tag{3.64}$$

It is also very popular in the literature a version of the SR that only takes into account the "bad" volatility called "Shortino Ratio" [2] , meaning that only measure the volatility for the return periods that are negative. This is a very controversial way of measuring the performance since it is not considered to be a good method to classify the volatility as "good" or "bad", the only thing that volatility express is risk - that is nor good or bad, is just risk. Since i agree with those critics i will just use the classic SR to assess the strategies.

3.9.2 maximum drawdown

Maximum drawdown is a measure of the the maximum observed loss in percentage terms from a peak to a trough of a portfolio, before a new peak is attained. Maximum drawdown is an indicator of downside risk over a specified time period.

$$MDD = \frac{LowestTrhough - Peak}{Peak} \tag{3.65}$$

3.9.3 reliability and meaning of a backtest

As mentioned in the beginning of this section, it is important to make a note about the meaning of a backtest. A backtest is a simulated investment strategy carried out on past data. The performance of a backtest are nothing else than a simulation, one should not think that the performance of a backtest are indicative of performances in the future, the only thing that a backtest can tell is what is likely to work compared to what is less likely.

Chapter 4

Testing Framework

In this section will be proposed a testing framework that will be used to evaluate a variety of trading strategies. Such a framework is necessary because in order to evaluate a trading strategy it is necessary to have a simulation environment. With the simulation environment it will be possible to simulate investment results indeed. The proposed trading strategies will be discussed in this chapter and are listed in the table 4.1 below.

Table 4.1: Table of presented strategies.

Strategy number	Description
1	Classic Threshold Approach
2	Classic Threshold Approach with time-varying parameters
3	Bollinger Bands Strategy
4	Bollinger Bands Strategy with NN classification entry signal
5	Bollinger Bands Strategy with NN regression entry and exit signal

Each of the strategies will be tested on the same pairs. The pair selection criteria is indeed the same for all strategies. This research will show how different approaches in trading and modelling the spread performs, therefore the pair selection criteria is only one and it is the same for all the strategies. The pair selection criteria will be also presented in this chapter.

4.1 Testing Framework Construction

4.1.1 The purpose of a testing framework

The only way to test an investment strategy without putting capital at risk is to build a testing framework and simulate operations on past data. Actually there is another way, and that is called "paper trading" and refers to the practice of operating a trading strategy on financial markets by using dummy capital. Paper trading provide a simulation very close to

real trading performance and it gives insights about an investment strategy that are much more reliable with respect to a simulation on past data, however it is time consuming.

Let's take a close look at why the results of a test on past data should not be indicative of actual trading performances:

- The main threat is due to the fact that test can be run multiple times with different settings, resulting in over-fitting to the available sample
- Every trading operation has an impact on the market and this phenomena is called slippage. This can be estimated but it remains only an estimation
- Past prices provides only the history of the price at which an asset was traded. No information about the order book is available. If an asset was illiquid[1] at a specific point in time it could not be known.

So, if the test performances are not indicative of future results in a real trading environment, what a testing framework is good for? It is good because it is the only thing that we can use to test an investment strategy. What one can do in order to make its results as reliable as possible is to state at the beginning of the research a list of assumptions and try to simulate as close as possible the real trading environment. One can get very sophisticated with the estimation of every single part of the test , for example by retrieving order book data to simulate real slippage values, or by using tick data to perform the tests.

However sophisticate the test can be, it will remain a test on past data. That being said, i would look at a testing framework as a way to tell the investor which approach is more likely to work in the future with respect to another approach.

The assumptions

The testing framewok designed for the research have the following assumptions:

- One hour closing prices are used, all trades are opened or close at round hours
- no stop loss is used
- Slippage, trading commissions and bid/ask spread are taken into account with a comprehensive $0,1\%$ fee on every trade
- every trade is perfectly filled and there are no liquidity issues

[1] Illiquid refers to the state of an asset that cannot easily and readily be sold or exchanged for cash without a substantial loss in value.

Trading a Pair

The strategy at the basis of the presented approaches is called Pairs Trading, this is because instead of trading single financial assets, it focuses on synthetic[2] assets created by a linear combination of two assets, called spread. This strategy is indeed also called "spread trading".

The rationale of the strategy is based on finding cointegrated assets, which by definition have a linear combination that provide a stationary series, For example let's assume that we've found two cointegrated assets X and Y, and we find the parameters of their linear combination β and α using a linear regression

$$Y = \hat{\beta}X + \hat{\alpha} + \varepsilon \tag{4.1}$$

Where ε the resulting errors in the regression. It can be seen as the resulting white noise. The idea behind pairs trading is to rewrite the equation as

$$\varepsilon = Y - \hat{\beta}X - \hat{\alpha} \tag{4.2}$$

Where β is called hedge ratio and indicates how many units of asset X one needs to short for every unit asset Y bought, when having a long position on the spread. Or, how many units of asset X one needs to buy for every unit asset Y sold short, when having a short position on the spread. From the equation (4.2), which is a sort of notation abuse that represent the spread equation, it is clear why theoretically pairs trading is a convenient place to be for a trader: the synthetic asset that is traded can be seen as the white noise resulting from the linear regression of two cointegrated pairs, which means something I(0), iid and with zero mean.

Even though theoretically pairs trading is the perfect strategy, when applied to the real world it is very uncommon to find equation (4.2) holding true. The term ε is very unlikely to have the characteristics of white noise in the out of sample data[3]. This means that we will never trade something which is I(0). On the other hands it turns out that in order to apply a trading strategy we don't necessarily need a I(0) asset, but it is enough an asset with mean reverting properties.

In order to perform a pairs trading investment strategy it is essential in the first place to select a pair - or better - a portfolio of pairs.

4.1.2 Pair selection and time windows

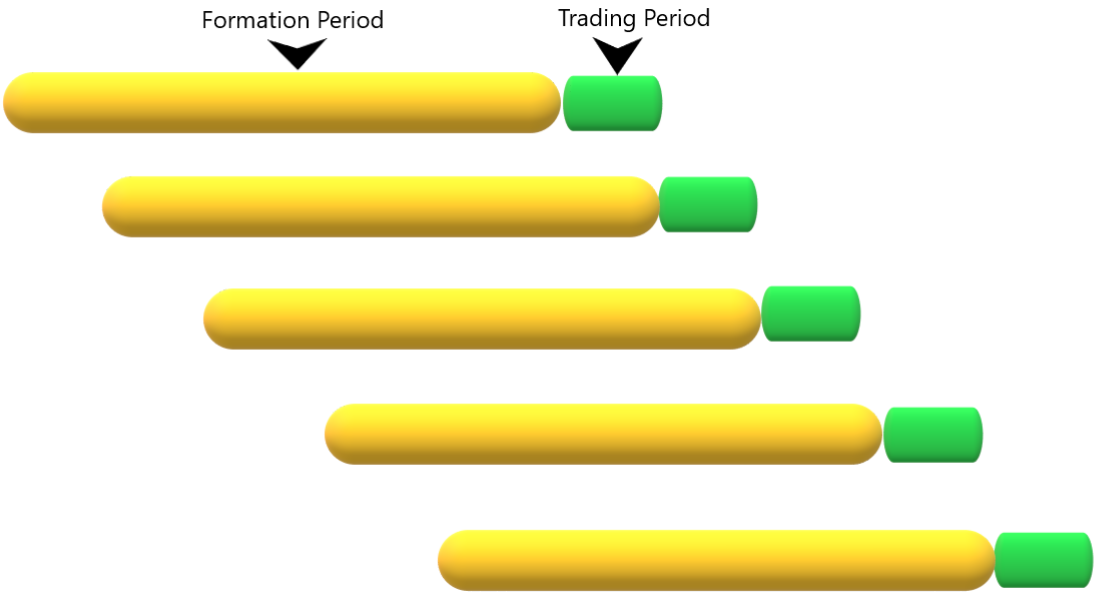
As the literature suggest [12], [9] amongst different methods of pair selection, the best one is the cointegration method, which consist in using using a statistical test to find pairs

[2] A position in which one takes various positions to create the same effect as holding a certain asset or other investment vehicle.

[3] Data not used to train the model

that are likely to be cointegrated and choose the pairs amongst them. In this research all other methods were ignored.

The available dataset covers two years of data. And a lot of assets in the dataset are missing at the beginning of the period. For this reason the trading framework was designed by following the idea of [9] with two dynamic time windows of pair selection period and trading period. In this way what remains constant is the methodology in which the portfolio of traded pairs is selected, whereas the whose basket of available assets can grow larger.



The periods for both the windows are fixed at:

- **Formation Period:** 3 months
- **Trading Period:** 15 days

For every formation period, the Engle Granger test was performed on all possible pair basket with a critical value of 1%. On all pairs that passed the test, the spread equation (4.2) was calculated. Amongst all available spreads, a choice had to be made. The options for a portfolio selection criteria that were tried out were:

- select the x spreads having the returns of its component more correlated

- select the x spreads having the prices of its component more correlated
- select the x spreads having the least Half-life
- select the x spreads having the least Hurst Exponent

Amongst all the selection method, the Hurst Exponent method turned out to work best, and therefore was selected as portfolio selection criteria. No particular portfolio optimization techniques were used. Instead the following simple rule was adopted: for each pair selection period, select the 10 pairs with the least Hurst Exponent amongst all pairs that resulted cointegrated by EG with a cv of 1%

This pair selection method is the one and only used in the research. All the different approaches that will be presented will differ for their techniques used in the trading period, whereas the portfolio of tradable pairs is the same in all approaches.

4.2 Classic threshold aproach - time invariant parameters

In this approach we assume the equation (4.2) to be true, meaning that we expect the spread to be perfectly stationary with zero mean. The approach taken is therefore a simple threshold based methodology: an upper threshold and a lower threshold are defined at a certain distance from the mean of the spread (which is zero). Whenever the spread crosses the upper thresholds a short position on the spread is opened, and the position is closed either when the spread cross back to zero or when the trading period ends.

The tresholds are defined by a measure of the sample standard deviation of the spread in the formation period. In particular, the upper one is defined as: $2 \times \hat{\sigma}$ where $\hat{\sigma}$ is the sample standard deviation of the spread in the formation period; and the lower one is defined as: $-2 \times \hat{\sigma}$ where $\hat{\sigma}$ is the sample standard deviation of the spread in the formation perid.

4.3 Classic threshold approach - time varying parameters

4.3.1 Kalman Filters

In this Approach, we assume that the equation (4.2) does not hold for out of sample data and we adopt the Kalman filter methodology explained in section 3.6.1 to update the parameters $\hat{\alpha}$ and $\hat{\beta}$ of the equation (4.2) at every new data point. The entries and the exits of this strategy are the same of the first strategy, the only difference is that the spread values are now calculated by using time varying parameters.

The following figures shows the spread between BCH and ASD calculated in the classical way, and the spread with the estimation of kalman filters. The period is from 31 January 2020 up until Friday 15 May 2020.

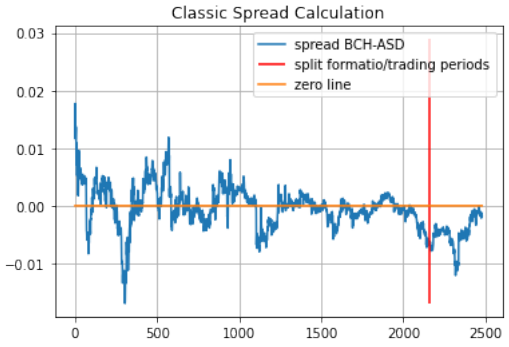


Figure 4.1: Spread Classic

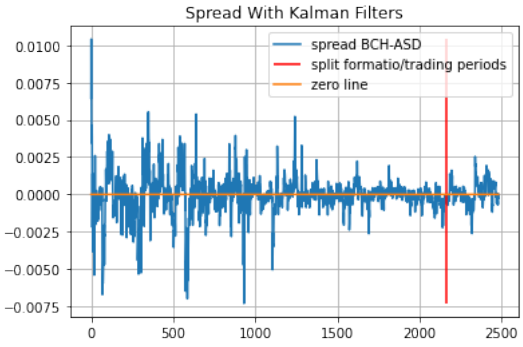


Figure 4.2: Spread With Time Varying Params

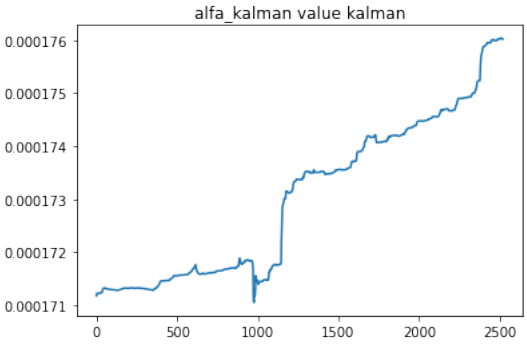


Figure 4.3: α parameter estimation

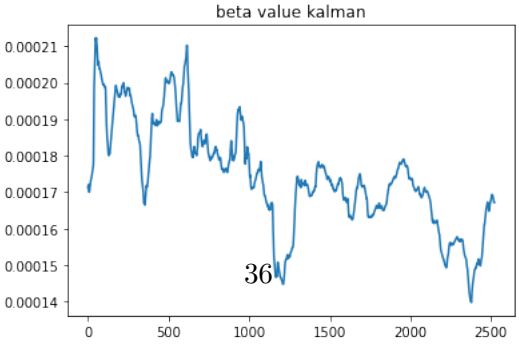


Figure 4.4: β parameter estimation

It is clear to see that the spread with time varying parameters shown in figure 4.2 looks much more interesting because it reverts more frequently on the mean. But it is also important to be aware that the spread calculated in this way is a mere indicator and it does not represents anymore a synthetic asset like the classic spread shown in fig 4.1. In the classic spread, if you buy the spread low and close the position at a higher spread price, you will turn a profit. This is because the parameters are fixed and the spread movements are the exact sum of its component movements. In the case of time varying parameters though this is not true, and that is because the price of the spread is not changing only because of the assets, but also because of the new parameters values. That is why in the testing framework only prices of the single assets were used to calculate profit and losses, not spread prices.

Figures 4.3 and 4.4 shows the estimates of the spread parameters of the spread in fig 4.2 .

This approach was presented in [3], where the author applies this methodology for trading the spread between iShares MSCI Australia ETF (EWA) and iShares MSCI Canada ETF (EWC).

4.4 Moving Averages

Also in this Approach, we assume that the equation (4.2) does not hold for out of sample data, but as a difference from the approach of the Kalman Filters, the parameters of the equation (4.2) will not be changed. Instead, we will modify the strategy in a way that it is adequate for an underlying asset which is not stationary, but has mean reverting properties.

Such strategy leverage moving averages and don't require the spread to revert to the value of zero. A moving average of a predetermined look-back shifted upwards by 2 sample standard deviations calculated on the same look-back period will be the entry point for a short position on the spread; the exit will be at the moving average without shift. A moving average of a predetermined look-back shifted downward by 2 sample standard deviations calculated on the same look-back period will be the entry point for a long position on the spread; the exit will be at the moving average without shift.

This use of three moving averages constitutes the technical indicator very popular in finance called Bollinger Bands. The look-back period for the test was chosen arbitrarily and was set at 50 periods.

Also in this case the approach was presented in [3], where the author applies this methodology for trading the spread between an two ETF: GLD and USO. A similar strategy was also applied in the crypto market by [9]

4.5 Adding NN forecasting - regression

4.5.1 How the ML problem is designed

By providing us with a forecast of the future price, neural networks can provide us with more information to use in the trading strategy. In the regression case, the objective is to forecast the future spread price by looking at the past realizations of the spread. If it possible to assess the forecasting goodness of the model as good, we will be able to use those forecasts at our advantage in trading.

The machine learning problem is set up in such a way that the forecast will be one hour in the future (one-step forecast) and indeed are used returns of the spread as features and labels. The construction of the spread is the most simple one, it is the spread obtained with the equation (4.1), where the parameters are calculated during the formation period. The available time series spread data to the machine learning problem goes from the first available data point of the formation period, up until the last data point of the trading period.

Data preparation

The data preparation process is the same for every spread series, and it is the following: it start with making the spread positive by adding such a number that makes it positive, the logarithm of the results is taken and the percentage returns of such logarithms are taken. In order to normalize those values, the sample mean and standard deviation is taken from the train set and the whole series is normalized and the series z is obtained:

$$z = \frac{\mathbf{x} - \hat{\mu}}{\hat{\sigma}} \tag{4.3}$$

Once all the spread period composed by the formation period is processed in the z values, they must be encoded in labels and features. Features are composed by a vector containing a sequence of 10 values of z:

$$X_t = [z_{t-9}, z_{t-8}, ..., z_t] \tag{4.4}$$

Labels are composed by the next value of z:

$$Y_t = z_{t+1} \tag{4.5}$$

The split of the data

The data are split in a way that it is not possible to fall into look-ahead bias: the data in the formation period is used for both train set and validation set with a ratio of $\frac{2}{3}$ on the train set. The data in the trading period are not touched before the model has stopped training.

The model

The model is this case is composed by 5 LSTM layers, whose input shape is the same of the feature data dimensionality $d = 10$. The activation function for all the hidden layers is the Relu Function:

$$y = max(0, x) \tag{4.6}$$

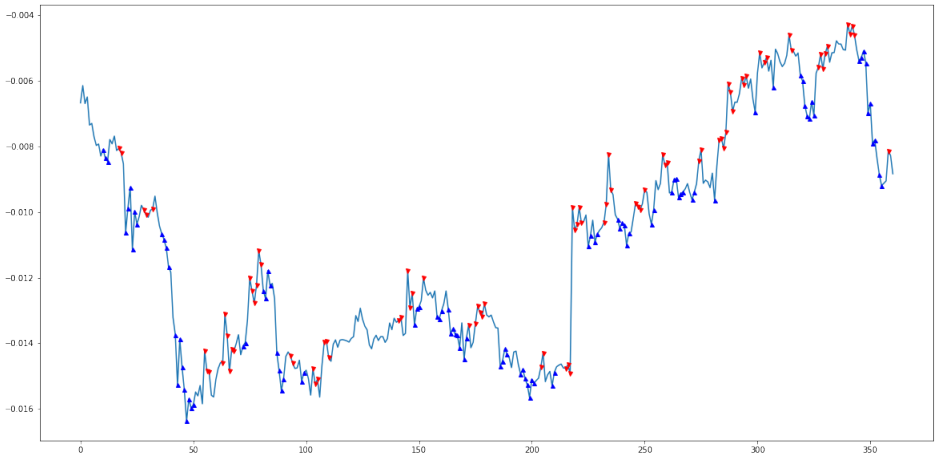
After those layers there is one layer composed by a single node for regression purpose. The loss function used for training and evaluation of the models is the mean squared error:

$$L(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(y - \hat{y}_i)^2 \tag{4.7}$$

Lastly, the number of epochs for the training period was set to 10.

4.5.2 Usage

Once the model is built and we are able to have some forecasts, we need to find a useful way to leverage the information provided by the model forecasts. It was indeed decided to use the model's forecasts as a further trading signal that must be added in an existing trading strategy. The base trading strategy was the one based on the Bollinger Bands. When the base trading strategy suggested a trading signal, the trade was taken only if the direction of the model forecast agreed with the direction of the trade. On top of that, not every forecast was considered as a signal, but only the ones which exceeded a certain threshold. Such a threshold is equal to the average value of the predictions in magnitude in the validation set. Forecasts that exceeded the threshold have proven to be more accurate. (see results)



The image above shows the signals provided by the model taken back on the spread at the time in which the prediction was made. Red arrows represents sell signals, blue arrows represents buy signals, the underlying time series is a spread. Note that the meaning of

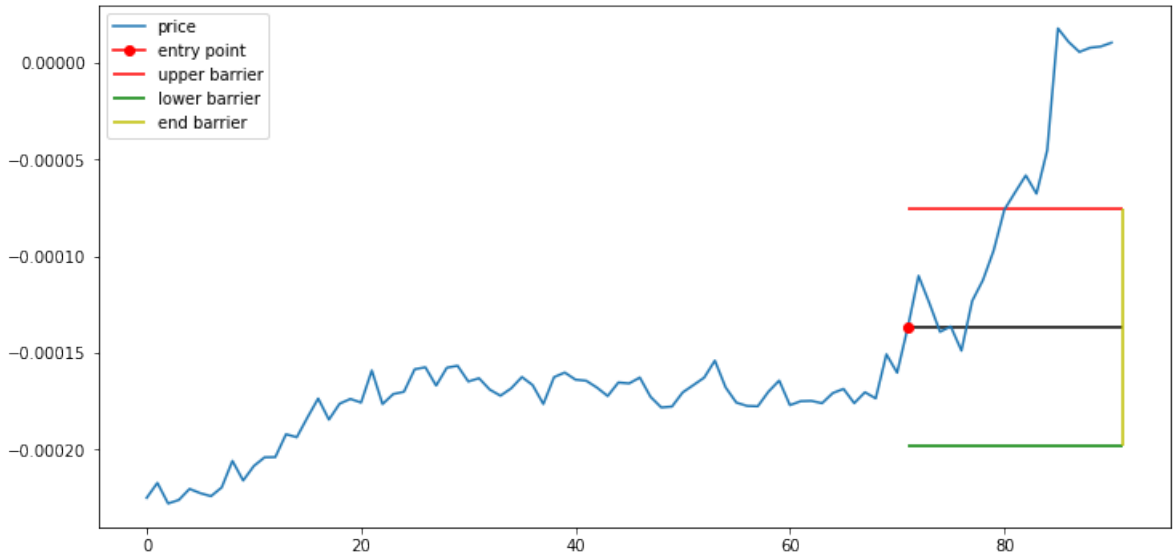
adding the model predictions in the strategy is that of making more precise entries: we want don't want to enter blindly in the market when the spread price is above/below the Bollinger bands, but we also want our model to think that the next data point will be in the right direction. Also, if we are in a position and the model detects a signal in the opposite direction, we close the position. The idea behind the exit is that we want to minimize the adverse excursion of our trades: we want to close a position if the forecast of the model is against our current position. Notice that in this particular strategy (5) we will have 2 exit rules: cross of the moving average and signal in the adverse direction of the current position.

4.6 Adding NN forecasting - classification

4.6.1 How the ML problem is designed

In the case of classification, we want our model to make predictions further in time, not just one period ahead. we will use the triple barriers method to label the data and we will decompose the problem into buy side and sell side.

The triple barriers method is a labeling method propsed by [4] that allows you to label financial data discriminating the situations of bullish trend from the situations of bearish trend from the situations of lateralization. This method is very convenient because it does not take into account a precise period in the future nor a fixed amount of price upward or downward, in fact the barriers are set in a multiplicative measure of volatility. In the research, the classification problem has been formulated in a binary way: in the buy case the label has been assigned equal to 1 if the price has reached first the upper barrier; zero instead in the case in which the price has reached first the vertical or lower barrier. The opposite has been done for the case sell.



In the case in the image, the label assigned for the buy model would be 1; the label assigned for the sell model would be 0.

Data preparation: labels

Since there are two models designed for two different purposes, two labelling processes had to be performed. To maintain a balance between labels, the distance of the high barrier in the buy case was set to a distance value equal to half of the low barrier point. The inverse was done in the sell case. This is because all cases in which the price touches the vertical barrier, this label will be equal to zero, and in order to have balanced forecasts we tried to generate labels that were half 1 and half 0.

10 periods were considered as the distance of the vertical barrier from the labeling point, still 10 periods was the look back period for the calculation of the standard deviation used to determine the upper and lower barriers. Specifically they were set at one and two standard deviations in the buy case and the opposite in the sell case

Data preparation: features

Instead of using the raw price as a feature for the model, we chose to use a very common and used technical analysis indicator, which measures the oversold and overbought situation of the price: the Relative Strenght Index. In particular 4 versions of the RSI were used, each with a different look back period. these periods are 14, 10, 5, 2. The value of the RSI is between 0 and 100, it was therefore sufficient to divide all the values by 100 to be able to normalize the features in the appropriate way for the model

The split of the data

The split of the data has been done in the same way of the regression case: the data in the formation period is used for both train set and validation set with a ratio of $\frac{2}{3}$ on the train set. The data in the trading period are not touched before the model has stopped training.

The model

The model includes an input layer composed of 4 nodes, two hidden layers composed by 128 nodes each and whose activation function is still the Relu activation function. It ends with an output layer composed by one node whose activation function is the softmax function. The loss function chosen for the model was the sparse categorical crossentropy function, which is a small modification of the function (3.42)

4.6.2 Usage

The use of the model In this classification case is very similar to the use case of the regression model. instead of having a signal when the forecast exceeds a predetermined threshold, we will have a signal when the two models do not contradict each other. in fact i recall that for each spread we have two different models: one that indicates opportunities to buy and one that indicates opportunities to sell. We will have a buy signal when the buy model gives a positive signal and the sell model does not give a sell signal. Vice versa for the sell case.

This type of labelling technique was introduced by [7].

Chapter 5

Results

5.1 Pair selection result:

The number of pairs for which the null of no cointegration was refused with critical value of 1% has grown dramatically during the testing periods of the research, as it can be appreciated by looking at figure 5.1 that shows the total number of likely cointegrated pairs per trading period.

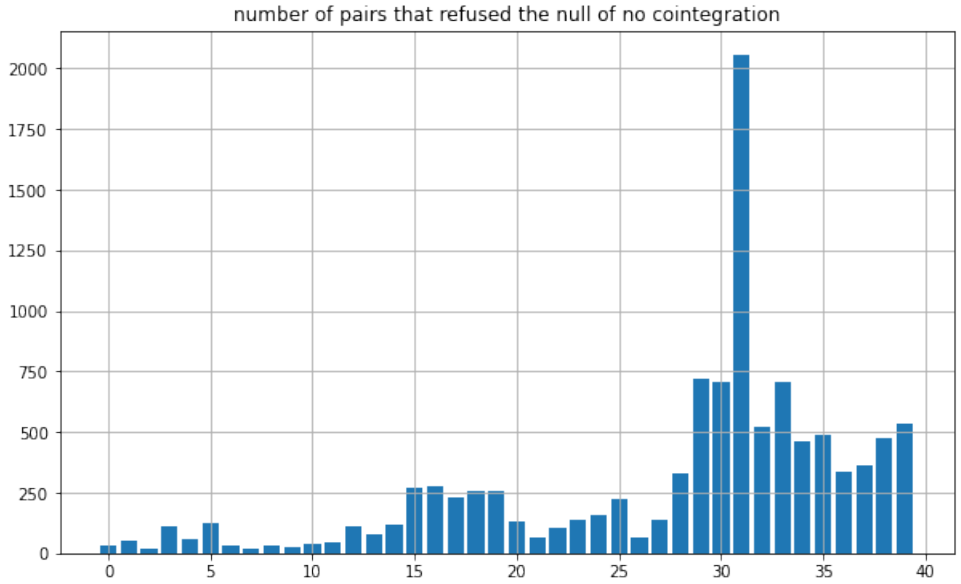


Figure 5.1: Number of likely cointegrated pairs w.p. 1%, per period

An important note must be made: these results regards the results of the EG test run with critical value of 1% on all possible combinations of assets available. This means that every pair of assets was tested with two possible combinations, in that way every asset was tested both as explanatory variable and as a independent variable. In the pair selection,

in the case in which amongst the 10 pairs with the least Hurst's exponent lied two pairs composed by the same assets, only one of the two were selected.

5.2 Machine learning models evaluation:

It is a good idea before applying the predictions generated by the models in the trading strategies, to assess the forecasts themselves in order to learn whether the models are any good.

5.2.1 Regression

In the regression case the evaluation of the signals is considered in the following way: if the forecast has the same sign of the actual label, the prediction is considered good, otherwise it is considered bad. This is actually the "zero-one" loss for classification problems. This accuracy metric will tell us in a understandable way whether the model is able to predict the direction of the next data point, it can be interpreted as the percentage of a forecast to be in the right direction.

The so defined accuracy metrics were calculated for models and for all periods in the test set, also it was calculated the same accuracy metrics for the forecasts included in the signals: meaning the forecasts that exceeded the predefined threshold.

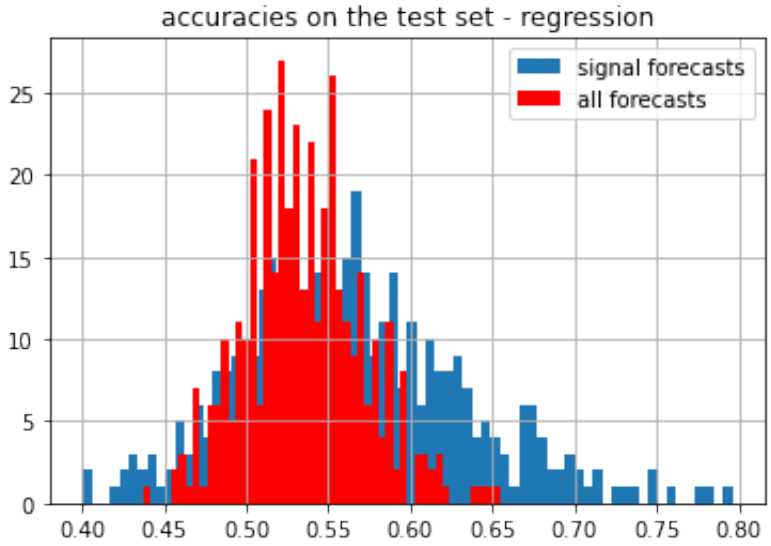


Figure 5.2: Accuracy metrics of regression model

As we can see from fig 5.2 the distribution of the accuracy metrics of the forecast which exceeded in magnitude a predefined threshold has a bigger variance w.r.t. the distribution

of the accuracy metrics of all forecasts, but it is also skewed at right, this means the average model has a better accuracy, even though some models will have a very bad accuracy metric (down to 40% accuracy). We can tell that the regression model can provide us with useful information.

5.2.2 Classification

In the regression case the evaluation of the signals is considered in the following way: if the forecast has the same label of the actual label, the prediction is considered good, otherwise it is considered bad. And this is the true "zero-one" loss function applied in its environment: a binary classification problem

The accuracy metrics for both models long and short were calculated for all periods and are grouped in the fig 5.3

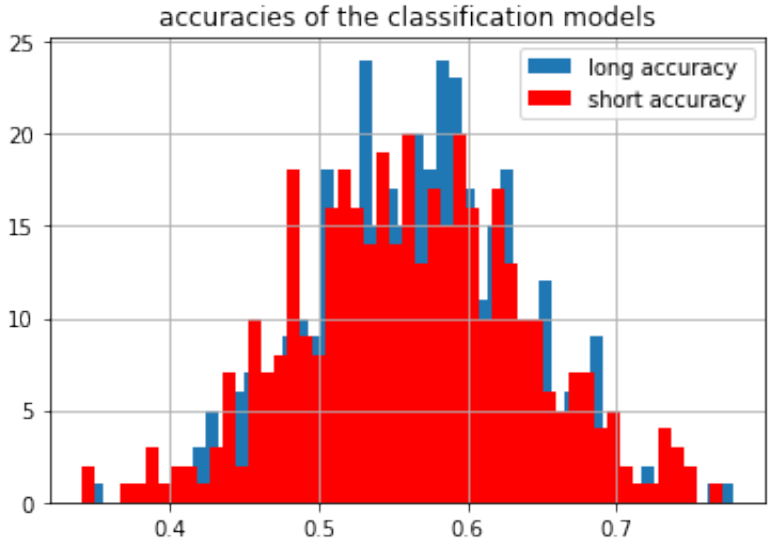


Figure 5.3: Accuracy metrics of classification models

As shown in the figure, the mean of both long and short distributions are above 50% value, this means that also in this case the models can provide us with useful information to use in the trading environment.

5.3 Strategies Recap

In the end we have discussed 5 different investment strategies. Each one of them use the same pairs selection methodology. They differ in the approach taken in calculating and/or trading the spread values. Lets recap the aproaches below:

- 1) Simple treshold aproach

- 2) Simple treshold aproach with time varying parameters estimated through Kalman Filters.
- 3) Bollinger Bands strategy
- 4) Bollinger Bands strategy with NN signal (classification model)
- 5) Bollinger Bands strategy with NN signal and exit (regression model)

The trading simulation was carried out by placing 100000 $ as initial capital, with 1000 $ for every position size and a comprehensive fee of 0.1% for every trade: note that this means that the total fee for a complete trade on the spread results in 0.4% total fee. This is because for every trade on the spread we have two trades on the assets; and in order to open and close a position we need two transactions, each of which require a trading fee.

5.4 Trading simulations results

The results of the five different approaches are grouped in the following table:

Strategy	ANNUALIZED SR	MAX DD	NUMBER OF TRADES
1	-0.84	17.0 %	378
2	-0.38	9.9 %	1330
3	1.03	11.1 %	3320
4	1.22	9.5 %	2776
5	1.50	6.2 %	3520

The Sharpe Ratio metric was calculated by considering cash as a risk free asset with a interest rate of 0%. The SR value is considered to be good only if it is above the value of 1; it is considered optimal if higher than 3; if it is below 0 it means that the investment strategy is expected to produce negative returns. Instead, regarding the Max DD metric, the lower it is, the better. The number of trades tells how active is the trading strategy: how many trades actually took place

As we can see, the best strategy according to both the evaluation metrics is the fourth approach, which obtain a Sharpe ratio of 1.5 and a maximum draw down of 6.2 % of the initial capital. The approaches 1 and 2 have shown poor results, but notice that for both metrics the time varying strategy outperforms the classical strategy. The bollinger band strategy has proven to be a good approach, as we can see by the SR of the strategy 3. Finally, the applications of machine learning filters to the strategy 3 show significant improvements, as the metrics for strategies 4 and 5 shows.

The daily equity curves[1] of the five simulations are plotted in the fig 5.4

[1] An equity curve is a graphical representation of the change in the value of the initial capital over a time period.

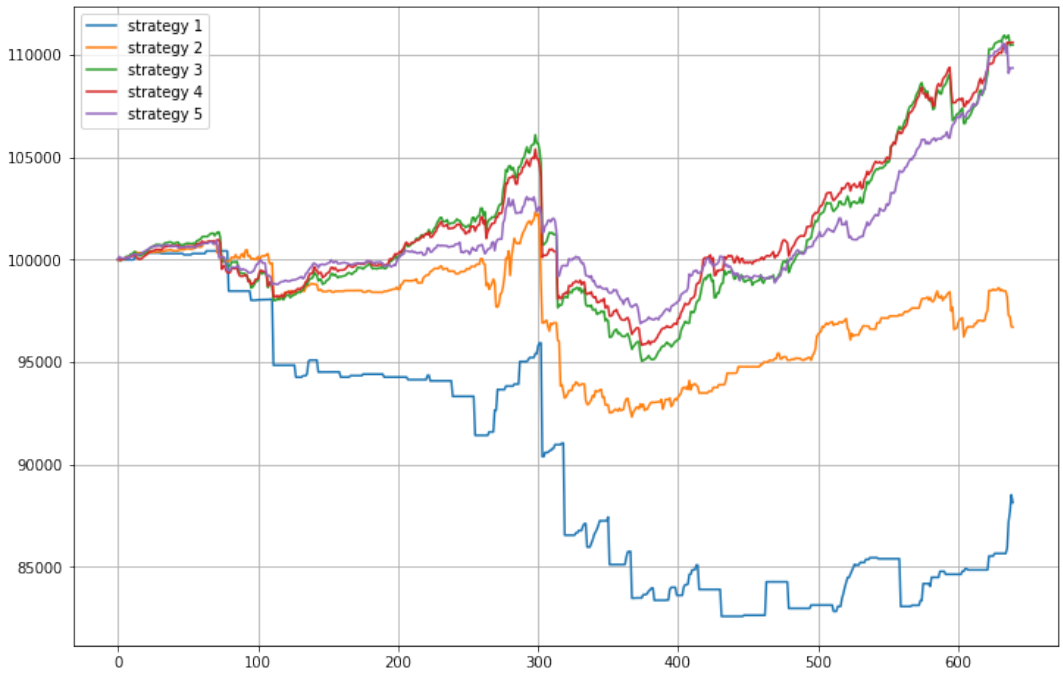


Figure 5.4: Equity Curve of the 5 strategies

As it is clearly shown by the graph 5.4, all five strategies are very correlated, this fact is normal and expected since the pairs traded by each strategy are the same and also the logic of the trading strategies is in all cases a mean reverting logic. Specifically, we see strategies 3,4,5 with very similar results in the equity line, and this is because strategies 4 and 5 are nothing else than the base bollinger band strategy - strategy 3 - with the application of a further signal provided by the NN models. In the regression case, the NN signal not only provides a further entry indication, but also provide an exit indication, this is why the total trades in strategy 5 exceeds total trades in the base strategy 3. The results of those strategies can be mistaken as equal by looking at the profit at the end of the period, but the risk profile is very different. According to the results, the best approach is the one taken by strategy 5; this strategy incorporates an additional exit rule that closes a position in the moment in which the price forecast is against the trader position, this design should make this strategy better risk wise. The results confirm indeed that the risk profile of the strategy 5 is the best one.

It is easy to spot a very bad period for all the strategies that goes from the 300th day up until the 400th day of trading. This period coincides with the end of 2020 - beginning of 2021, period in which the crypto market was in full bull run and a lot markets entered in a phase called FOMO[2]. In such a market environment, the application of a mean reverting

[2]"Fear of missing out" is a term used in financial market when the crowd aggressively buy an asset resulting in big spikes upwards in the price

strategy is definitely something that an investor should avoid. The fact that the best performances of this type of strategy are when the markets are not going euphorically to the upside is a very good thing, and this is because in such a market situation is where the profit of the classic buy and hold strategy are made. This means that a combination of those kind of strategy with a trend following strategy, like for example the naive buy and hold method, should provide an investor with more constant performances.

The results also shows that the naive approach of the strategy 1 has no chance to work, at least in the time scale considered in the research. Although such naive approach has proven to work on larger time on others assets like stocks ETF. [3]

Chapter 6

Conclusion

6.1 Comment and key findings

The results of the research have shown some important findings to be aware of when applying the pairs trading strategy on the cryptocurrency market. The first finding is that this type of strategy should not be applied in scenarios where the market enviroment is in a state of euphoria. Luckily, the strategy implemented for the research is a very short term strategy, wich could be classified as a intra-day[1] strategy; this means that the investor should recognize in time the status of the market environment and make his own decisions accordingly. The second finding is that the strategy implied to trade the spread can make or brake your results: the theoretical threshold approach has proven to perform very badly - at least on the time frames used for the research. The use of a simple bollinger band strategy applied on the spread has proven to be a good strategy. The third finding regards the successful applications of machine learning forecasting problems applied to financial data: both the regression model and the two classification models has proven to be able to learn from the data and provide the investor with useful information. The fourth and last finding is that the usage of the information provided by the machine learning models in a trading environment can dramatically improve the performance of a strategy, not from the net profit point of view, but from the risk-adjusted profit point of view: a metric expressed indeed with the Sharpe Ratio.

6.2 Future work

This research have shown some important aspects and methodologies in the argument of pairs trading investment strategy, nevertheless some aspects of such a strategy could have been developed much more. This choice was due to the fact that by changing one single building block in the testing pipeline, the results to be compared would have multiplied, making the entire research a mess. With that being said, there are some topics in which future work could be find place:

[1] A trading strategy whose positions are opened and closed within 24 hours

The research presented a very simple strategy for pair selection: this was chosen because it was a methods that was built on something that the literature have shown to work good [12]: only co-integrated pairs ware indeed selected; also it guaranteed the same number of assets for every period of trading; finally the selection based on the Hurst exponent value had a meaningful intuition, and has proven to work best. Besides that, it is also true that one could have performed studies and tests by applying correlation amongst spread when choosing the pairs, or some of the topics inside the portfolio selection theory: both good points on which one could develop this research on.

Another hint for possibility of future work could be the application of some methodologies in order to detect the market conditions, and determine whether such conditions are in favour for a specific investment strategy. This could be another application of machine learning to the field of trading and investments.

One last possibility for future work could be the study of other kinds of applications of the predictive power showed in the research by the machine learning models to the markets. In the research, such a predictive power was used as a filter for an underlying trading strategy. It could be that more adequate applications of the models exist.

Bibliography

[1] Binance Academy. Futures contracts. Available at `https://academy.binance.com/en/articles/what-are-forward-and-futures-contracts`.

[2] C.R. Bacon. *Practical Risk-Adjusted Performance Measurement*. The Wiley Finance Series. Wiley, 2012.

[3] E. Chan. *Algorithmic Trading: Winning Strategies and Their Rationale*. Wiley Trading. Wiley, 2013.

[4] M.L. de Prado. *Advances in Financial Machine Learning*. Wiley, 2018.

[5] Theodoros Tsagaris Giovanni Montana, Kostas Triantafyllopoulos. Flexible least squares for temporal data mining and statistical arbitrage. 2009.

[6] investopedia. Systematic risk. Available at `https://www.investopedia.com/terms/s/systematicrisk.asp`.

[7] J. Klaas. *Machine Learning for Finance: Principles and practice for financial insiders*. Packt Publishing, 2019.

[8] T. Leung and X.I.N. LI. *Optimal Mean Reversion Trading: Mathematical Analysis and Practical Applications*. Modern trends in financial engineering. World Scientific Publishing Company Pte Limited, 2015.

[9] Irina Kortchemski Masood Tadi. Evaluation of dynamic cointegration-based pairs trading strategy in the cryptocurrency market. 2021.

[10] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014.

[11] L. Troiano, A. Bhandari, P. Kriplani, and E.M. Villa. *Hands-On Deep Learning for Finance: Implement Deep Learning Techniques and Algorithms to Create Powerful Trading Strategies*. Packt Publishing, 2020.

[12] R.W.J. van der Have. Pairs trading using machine learning: An empirical study. 2017.

[13] volodja. Ornstein – uhlenbeck process. Available at `https://mathtopics.wordpress.com/2013/01/07/ornstein-uhlenbeck-process/`.

[14] Wikipedia. Cryptocurrency. Available at `https://en.wikipedia.org/wiki/Cryptocurrency`.