Enhancing a Pairs Trading strategy using Financial Indicators with an application of Machine Learning

João Nuno Costa dos Santos
joaonuno98@tecnico.ulisboa.pt

Nuno Cavaco Gomes Horta
nuno.horta@tecnico.ulisboa.pt

***Abstract*—Trading is a popular market-neutral investment strategy used by investors worldwide. This strategy focuses on relative price, profiting both from increasing and decreasing prices, thus avoiding high market volatility. By carefully selecting the pairs and analysing their behaviour, the investors pursue market opportunities to sell a relatively overvalued security and simultaneously buying an undervalued one. These opportunities usually arise from a spontaneous divergence, and a profit is made from the eventual pair's price convergence. Due to the evolution of computing power and higher accessibility of data, over the last decades, more and more investigation has been made into new investment approaches.**

In this work, it's proposed an enhanced model of Pairs Trading through the use of Long Short-Term Memory Networks to forecast the behaviour of stocks based on its financial indicators. These forecasts aim to either entering earlier or later (than the reference that is the simple threshold-based model) a certain opportunity to increase its profit. Also, two other decision functions were added to make the overall enhanced model less vulnerable to abnormal market fluctuations.

During the test period, the proposed model, had a 54% increase in profit, when compared with the regular threshold-based model. However, this increase in performance is not due to the forecasting itself, but rather due to the decision functions that not only mitigate potential losses but also invest in new opportunities that the traditional model doesn't.

***Index Terms*—Pairs Trading, Stock Market, Neural Networks, Financial Indicators, Machine Learning**

I. Introduction

Pairs Trading is a popular market-neutral[1] investment strategy developed in the 1980s.The opportunities for investment in ”Pairs Trading” rely on the premise that if the stock prices of the securities in the pair have followed each other, then it should continue in the future. Accordingly, if there is a divergence, it should mean that it is an attractive opportunity to invest assuming the prices will converge afterwards. These opportunities are found through the monitorisation of the spread[2] of the pair. Whenever there is a spread anomaly, a market position is entered, then, after the prices converge, it is exited.

Figure 1 is an example of pairs trading being applied to a pair of stocks where its normalized spread is defined as:

$$S_t = \frac{(PFE_t/JNJ_t) - \tilde{x}_{200}}{\sigma_{200}} \tag{1}$$

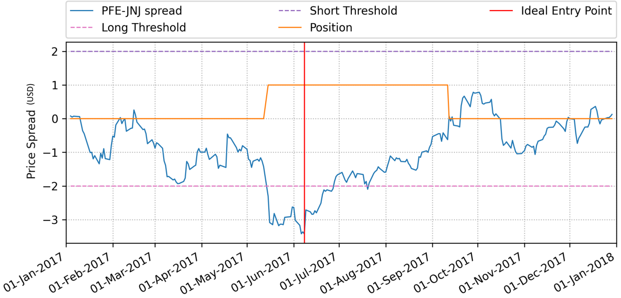

Fig. 1. Example of Pairs Trading applied to a pair of securities

during the year of 2017. It is worth noting that $\tilde{x}_{200}$ and σ_{200} represent the mean of the ratio and the standard deviation respectively, of the previous 200 days [3]. This value is arbitrary and it is used to prevent ever growing spreads that will incur in huge losses, a lower look-back period results in a more unstable spread.

Research in the field relies on purely statistical data to enhance this strategy. Even though Machine Learning applications have exponentially grown in the financial market, concerning Pairs Trading, there haven't been many improvements. This lack of research opens up a compelling opportunity to explore Intelligent Computation methods applied to the Pairs Trading strategy.

This work is divided in three main stages: the first one proposes an approach to find pairs for the investment strategy; the second focuses on creating forecasting models for the selected stocks combining financial indicators to achieve better performance; lastly, the third stage aims to use the information provided by the forecasting model to enhance the regular pairs trading investment strategy.

II. Background and Related Work

Each stage of this project is described in detail along with the most relevant related work.

A. Pairs Selection

Selecting pairs for this strategy comprises two main steps: (i) selecting all eligible securities for the portfolio and (ii) pairing them up together in the most promising way possible. Regarding the first step, there have been works using two different approaches. The first one is selecting specific industries, countries or another particular group [1], [2]. This method will result in more predictable pairs and will save a

[1] A market-neutral strategy seeks to profit both from increasing and decreasing prices in one or more markets, while trying to avoid market risk

[2] the spread is defined to be the ratio between the price of two securities

[3] this number will be referred to as ”look-back period”

lot of computing time. After selecting the group of eligible securities, the investor must define which ones to combine to form the most promising pairs. The most common procedures to select pairs involve the squared distance, cointegration and correlation.

The SSD measure stands for ”Sum of Squared Distances” of the price series of two stocks. However, since each security has a different price range, the most common way to mitigate these differences is to normalize the values. For each time series, the normalized price is determined by:

$$P_t' = \frac{P_t - \tilde{x}}{\sigma} \tag{2}$$

where P_t is the price of the asset at the moment t, $\tilde{x}$ is the average price of the asset throughout the time series, σ is its standard deviation and lastly P_t' is the normalized price. After the normalization the calculation of the SSD measure can be done, using the following equation:

$$SSD_{x,y} = \frac{1}{n}\sum_{t=1}^{n}(x_t' - y_t')^2 \tag{3}$$

For an optimal pair, the investor is looking to minimize this value since it would mean that both securities' price series have had similar behaviour in the past. A zero spread pair shall not be considered optimal as it would not provide trading chances.

Two series (x and y) are said to be cointegrated if the linear combination $x_t - \beta_2 y_t$ is stationary. Firstly and using the Engle-Granger two-step approach, the static regression shall be estimated by:

$$x_t = \mu + \beta_2 y_t + u_t \tag{4}$$

where μ and β_2 are constant values, and u_t is a residual term that must be stationary in order for the series x and y to be cointegrated. Here the Dickey-Fuller [3] test is performed to test the null hypothesis of no cointegration.

After proving that there is a cointegration relationship, the ’p-value’[4] is obtained through regression surface approximation as explained in by MacKinnon in [4], [5]. Pairs with ’p-values’ under 0.5 are considered mean-reverting[5] stock pairs.

The Pearson correlation method defines the correlation between two price time series x_t and y_t by:

$$CORR_{X,Y} = \frac{\sum\limits_{t=1}^{n}(x_t - \tilde{x})(y_t - \tilde{y})}{\sqrt{\sum\limits_{t=1}^{n}(x_t - \tilde{x})^2}\sqrt{\sum\limits_{t=1}^{n}(y_t - \tilde{y})^2}} \tag{5}$$

where $\tilde{x}$ and $\tilde{y}$ are the average values of the time series x_t and y_t respectively. In [6] it is reported that the correlation measure is has a big impact on overall return and risk. In general stock pairs with higher correlation tend to be better candidates for pairs trading.

B. Stock Price Prediction

Recurrent Neural Networks are a variance of Neural Networks where the output at each step is fed into the next one (Figure 2) whereas in regular NN's all the inputs and outputs are independent of each other. This makes them suitable to tasks such as text and speech recognition [7], [8], [9], or stock price prediction [10], [11], [12].

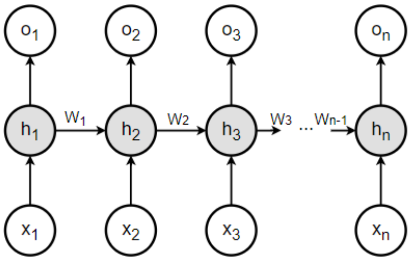

Fig. 2. Recurrent Neural Network Structure

Considering the RNN structure, if every hidden unit (h_x) is replaced by an LSTM cell and between every cell there is another connection called ”Cell State” the resulting structure is what is called Long Short-Term Memory Network. Each LSTM cell defines its internal state as a function of the current state and input, through a gating mechanism. In Figure 3 it is depicted a scheme of an LSTM cell.

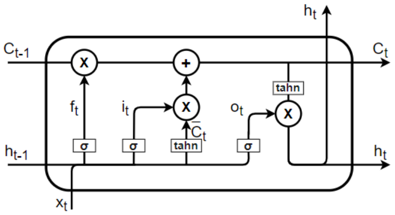

Fig. 3. Long Short-Term Memory cell

III. Proposed Pairs Selection Framework

During this stage, it will be explained how an investor may find pairs suitable for the implementation of Pairs Trading. This framework comprises three main steps, the range selection, the calculation of the three measures followed by its filtering, and lastly the historical profit calculation of the remaining pairs.

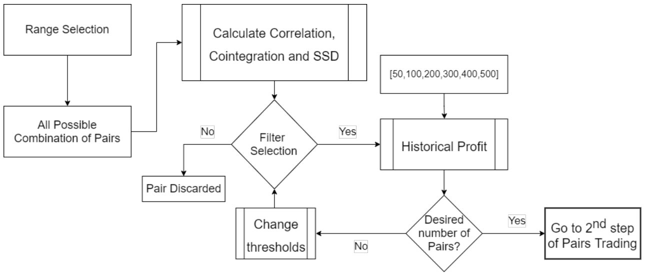

Fig. 4. Pairs selection process

[4] The p-value ranges from 0 to 1

[5] Mean reversion is a theory used in finance that suggests that asset prices and historical returns eventually will revert to the long-run mean or average level of the entire dataset

A. Range Selection

airs trading can be applied to any asset in the stock market like stocks, Exchange-traded funds, currencies or indexes. The eligible securities for pairs trading don't need to come from the same country or industry. However, in order to reduce the possible range of electable pairs, it is common that investors opt to restrict the pool of securities within a certain industry, core business or country.

There are a couple of indexes that measure the value of a section of a country's stock market via a weighted average of selected stocks and can be used as a pool of stocks. The three most common types of indexes are the Global, Regional and National indexes. For this particular work, only the stocks listed in the Nasdaq-100 index will be evaluated. The Nasdaq-100 is one of the most preeminent large-cap growth indexes, as it includes one hundred of the largest non-financial companies listed on the Nasdaq Stock Market, based on market capitalization.

B. Filter selection

To every possible pair resulting from the range selection, it is calculated both the cointegration, correlation, and SSD measures. It is worth noting that, 'p-values' closer to 0 means that there is a better cointegration between both stocks than the bigger 'p-values'. The correlation among two-time series also ranges from 0 to 1, however, the higher the value is, the better correlation exists among them. For a pair to be elected it has to have a 'p-value' below 'i' and a cointegration value above 'c', being 'i' and 'c' arbitrary thresholds.

Based on the filters selected by the investor, a different number of pairs will compliant and will be assessed in the following stage.

C. Historical Profit

After selecting a smaller group of pairs, the threshold-based trading model as the one explained in figure 1 will be simulated and each pair's historical profits shall be analysed. This should give an empirical confirmation of which pairs suit the most this investment strategy. Also, for each pair, the lookback period described in the equation 1 was ranged from 50 to 500. Lower values of the lookback window, allow for more market positions opened, however, it reduces the percentage of profitable transactions. Studying how each pair would have performed in the past for each look back period value, should be a good indicator of how they will behave in the future.

IV. Proposed Forecasting Model Creation

Following the overall architecture of this project, after selecting the pairs that will compose the portfolio, the next step is to create a forecasting model to help predict each pair's spread's behaviour. To achieve this goal, for every stock in the portfolio, a different combination of input features will be tested in a Neural Network. The models that offer the best performance will be used for each stock.

A. Training Model

As usual, the data from 2000 to 2018 will be separated into "training" and "test" data. For every stock within the portfolio, a new model for price prediction will be created, in order to maximize profit. This will also be an iterative process since some parameters can be changed to form new models. Once the desired performance is achieved, the model is saved.

B. Feature Preparation

Financial indicators are often used to support investors choices. Also, the price evolution of the respective stock's pair will be used as an input feature to train each model. All stocks were tested with all possible combinations of input features on two different train/test data divisions (80%-20% and 90%-10%). Also, different look-back windows[6] were tested (5, 10, 15 and 20 days).

All these parameters will be combined and all possibilities will be tested to find the best performing model for each one of the twelve stocks.

C. Five day forecasting

The two process explained before will be performed once for each stock, and consecutively, twelve models with different combinations of input features and look-back windows will be saved. These are the models that achieved the best performance during the test phase. Afterwards, for each stock, four new models are going to be created with the same combination of parameters, to predict the next four days. At the end of this stage, for each stock, there are five models that aim to predict each one of the following five days.

V. Trading and Portfolio Management Models

A. Problem Statement

The Pairs trading strategy profits from the mean reversion of the spread. However, the threshold-based model described in figure 1 defines the market entry point whenever the threshold is crossed. This will lead to periods of high uncertainty as the pair continues to diverge as shown in Figure 5.

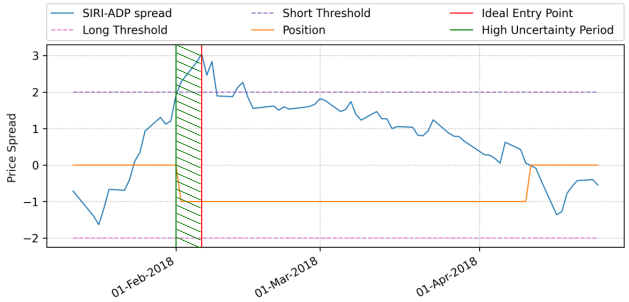

Fig. 5. Example of a high uncertainty period

The proposed trading model aims to increase the accuracy of the optimal entry point.

[6] The look-back window represents how many days worth of information are fed into the model

B. Trading Model

The proposed model will be triggered as the spread reaches near one of the two thresholds (short and long). From this point on, any time would be good to open a position, however, the ultimate goal is to enter on the ”Ideal Entry Point” explained in Figure 5. Using the predicted prices obtained from IV, the algorithm will try and plot the evolution of the spread for the following couple of days. While the forecast for the following days keeps deviating the spread from the mean, the market position will remain on hold. As soon as the spread starts to revert back, the algorithm will open a position (long or short).

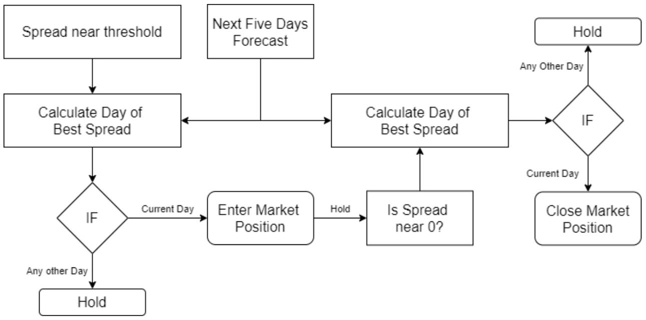

Fig. 6. Trading model decision flowchart

C. Stop Loss-Function

Remembering the fundamentals of Pairs Trading, profit is made from the convergence of the Spread to its historical average, hence why the normalized spread is used. However, if a short position is opened, the profit will be made if the regular spread decreases (and vice-versa for a long position). Whenever a position is opened for a very long period of time, there is a huge risk that for some odd reason, the pair is no longer correlated and its spread will no longer revert to its historical mean. This will result in huge losses for the investor and having a stop loss function is a fundamental tool to avoid catastrophic results. To create a stop-loss function, there are two fundamental values that need to be looked at:

- Position ”age”: How many trading days has the position been opened for:
- Spread’s deviation: How much has the spread increased or decreased

VI. Implementation

A. Pairs Selection

Following the process described in Figure 4, and only using information from 2000 to 2018 the first step to do is to select the range of stocks. Out of the 100 stocks listed in the Nasdaq-100 index, we started by filtering out all those that didn’t have at least 5000 trading days of information (leaving us with 69 stocks). With the remaining ones, for every possible pair it was calculated the cointegration (as explained by MacKinnon in [4], [5]), correlation (5) and the SSD (3). With this information, Table I was created, indicating how many of these pairs remain when the three thresholds are changed.

Out of the 2346[7] possible combinations, In Table I the amount of pairs that would comply with all the selected filters is demonstrated.

TABLE I
Variation of the number of Pairs based on the selected filters

p-value	Corr. Value	SSD Measure (s)			
		0.08	0.07	0.06	0.05
0.02	0.95	51	39	29	13
0.03	0.96	61	**47**	34	16
0.04	0.96	72	54	38	19
0.05	0.96	77	56	38	19

For all 47 pairs that were selected in the previous section, the simple threshold-based model was run during the formation period (2000-2018) with different values for ”look-back days”. Reducing the look-back period would increase the number of market openings, however, it would also reduce the profit of each transaction. Having it in mind, and after looking at the historical profits, the top pairs that had a more consistent profit with a satisfactory number of market positions per year were selected. The final six selected pairs are:

- ADBE-MSFT
- ADP-INTU
- AMGN-CMCSA
- HAS-PAYX
- IDXX-MCHP
- LRCX-MAR

B. Forecasting Model

The goal of this section is to create a forecasting model for each selected stock presented above. As explained in II-B, LSTM networks are well-suited to classifying, processing and making predictions based on time series data, hence why they were chosen for this project. Different combinations of input features were tested on the LSTM and its accuracy was accessed during the ”test period” (2000 to 2018).

To create the LSTM network, the ”keras” library was used. The network was created with two LSTM layers of 50 neurons each, and then with a 25 neurons ”Dense” layer. Then, the model is compiled with the ”adam” optimizer and using the ”mean squared error” as a loss function, that indicates how well the model is predicting.

As explained earlier, it was only used data from 2000 to 2018. The first 90% of trading days were used to train each model and the last 10% to evaluate its performance. During the test period, the predicted closing price value was compared to the real closing value through the use of the Root mean Squared Error Function.

$$RMSE = \sqrt{\frac{sum(predicted\_value - real\_value)^2}{\#\ trading\ days}} \tag{6}$$

Since each prediction is made in USD, this value was then divided by the average closing price of the test period so that the evaluation of the performance of each model is made as a percentage.

[7] $_{69}C_2 = \frac{69!}{2!(69-2)!} = \frac{69!}{2! \times 67!} = 2346$

$$RMSE\% = \frac{RMSE}{average\ closing\ price} \tag{7}$$

This adjustment allows us to better compare how each combination of input features behaves across the different stocks.

To train and test models, there is the need to collect and process data. As explained in picture 7, with the values collected in step 1, the input features are calculated. Afterwards, they are normalized to avoid big differences in the scale of the data. Lastly, for each model, different combinations of input features may be chosen.

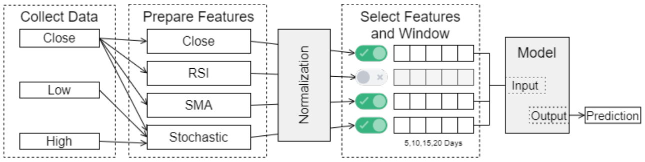

Fig. 7. Feature Preparation Process

To collect data, an API called Tiingo2was used to load the information of each stock into a 'pythonpickle file'. As the stock market only opens on work days, during our test period (from 2000 to 2018)there are more than 4000 days worth of data. In each day, for each stock it is loaded its "High" [8], "Low" [9] and Adjusted Close [10].

As input features, three financial indicators (SMA, Stochastic and RSI), will be used. Additionally, the closing price of both the stock itself and the respective stock pair will be used to create the models for each one of the stocks that make the portfolio.

Regarding the Simple Moving Average, it is usually used by investors a "long" SMA (10 day SMA), and a "short" SMA (10 day SMA) to confirm market trends. To use this information as an input feature, the SMA-10 was subtracted by the SMA-50.

The Stochastic Oscillator is calculated using the following the equation:

$$\%K_t = \frac{P_t - L_{14}}{H_{14} - L_{14}} \times 100 \tag{8}$$

where H_{14} and L_{14} are respectively the highest and lowest prices of the past 14 trading days. The value of %D (the one saved for each day) can be calculated through a Simple Moving Average of the past 3 days of the value of %K.

The RSI is calculated using the following two equations,

$$RSI_t = 100 - \frac{100}{1 + R_{14}} \tag{9}$$

$$R_{14} = \frac{(Prev\ Avg\ Gain \times 13) + Current\ Gain}{-((Prev\ Avg\ Loss \times 13) + Current\ Loss)} \tag{10}$$

where for the calculation of the R_{14} the "Previous Averages" are calculated with the previous 13 trading days. An RSI value over 70 indicates that a security is "Overbought" and may be on the verge of a momentum shift. An RSI value under 30 is considered "Oversold".

This calculation process is made right after the data collection and three new arguments (RSI, SMA and Stochastic) are added to each day's information. This information is updated to the python pickle file to avoid repeating this process every time.

To predict the following day, the model is trained using a combination of input features with its respective values for the previous days (further mentioned as "Feature Window"). Besides testing the accuracy of the model with different feature combinations, it was also tested different feature windows of 5, 10, 15 and 20 days.

After testing every combination of input features, with all four possible feature windows, on a single epoch[11]. Having these models locked for each stock, the next step was to increase the number of epochs to try to achieve better accuracy. As such, each model was trained with 1, 2, 3, 4 and 5 epochs and the best-performing ones were the ones demonstrated in table II.

TABLE II
BEST PERFORMING MODELS FOR EACH STOCK

Stocks	Input Combo	Window	Epochs	RMSE %
ADBE	[Close,SMA]	20	3	0.56%
MSFT	[Close]	5	3	0.44%
ADP	[Close,SMA]	10	3	0.54%
INTU	[Close,Stoch]	15	3	0.04%
AMGN	[Close]	20	3	0.34%
CMCSA	[Close,RSI]	15	3	0.42%
HAS	[Close,Stoch,SMA]	5	3	0.13%
PAYX	[Close,RSI,SMA]	5	3	0.48%
IDXX	[Close,RSI]	20	5	0.03%
MCHP	[Close,Stoch,SMA]	20	4	0.16%
LRCX	[Close,Stoch]	15	5	0.36%
MAR	[Close,Stoch,SMA]	5	3	0.28%

In this project we will use the same input features and input window on all models of the same stock. To achieve this, during the testing phase, the model had as an expected output, not the following day but the day after, training it to forecast how much the closing price of that stock would cost after two trading days. The same process was applied to train and test models that would predict the third, fourth and fifth following days.

VII. Proposed Trading Model

This last step of the project was built using Microsoft's tool Power BI due to its capabilities of displaying data in a very clear and simple way. This section will be first demonstrated how the regular threshold model was created in Power BI. This model will be compared side by side with the proposed model during the test period of 2019 and 2020.

It is important to understand that in Power BI, all data is stored in tables. Similarly to Excel tables, these are composed of rows and columns and can have formulas to calculate values

[8] highest value reached on the day (in USD)

[9] lowest value reached on the day (in USD)

[10] an amends to a stock's closing price to reflect that stock's value after accounting for any corporate actions

[11] An epoch is a term used in machine learning and indicates the number of passes of the entire training dataset the machine learning algorithm has completed

TABLE III
ACCURACY OF MODELS ON EACH FORECASTING DAY

Stocks	Day 1	Day 2	Day 3	Day 4	Day 5
	RMSE %				
ADBE	0.56%	0.17%	0.40%	0.03%	0.23%
MSFT	0.44%	0.50%	0.04%	0.21%	0.13%
ADP	0.54%	0.16%	0.33%	0.03%	0.21%
INTU	0.04%	0.09%	0.27%	0.17%	0.46%
AMGN	0.34%	0.13%	0.02%	0.25%	0.32%
CMCSA	0.42%	0.00%	0.44%	0.06%	0.12%
HAS	0.13%	0.21%	0.24%	0.09%	0.30%
PAYX	0.48%	0.20%	0.33%	0.33%	0.16%
IDXX	0.03%	0.57%	0.02%	0.49%	0.45%
MCHP	0.16%	0.33%	0.16%	0.15%	0.47%
LRCX	0.36%	0.08%	0.16%	0.19%	0.18%
MAR	0.28%	0.18%	0.08%	0.26%	0.02%
Average	**0.31%**	**0.22%**	**0.21%**	**0.19%**	**0.26%**

based on other cells and tables. One big difference from Excel is that a column can only have one formula, written in DAX (Data Analysis Expressions), that will define all its cells.

A. *Data Structure*

Before writing DAX code and creating models, some tables need to be imported from other sources. Firstly, the Tiingo API was used to import the adjusted close prices of the twelve stocks that compose the portfolio from the year of 2018[12] to 2020, this table is called "Real Values" (Table IV).

TABLE IV
STRUCTURE OF TABLE "REAL VALUES"

Real Values				
Date	High	Low	Adj Close	Stock
1/2/2018	177.8	175.26	177.7	ADBE
⋮	⋮	⋮	⋮	⋮

Then the table "5Days Forecasting" (Table V) was generated through a python script, using the models created earlier were used to forecast the following five days of every day of the test period.

TABLE V
STRUCTURE OF TABLE "5DAYS FORECASTING"

5Days Forecasting			
Date	Predictions	Stock	Predicted Day
27/4/2018	158.0	AMGN	1
27/4/2018	157.7	AMGN	2
⋮	⋮	⋮	⋮

Finally, a small excel table called "Pairs Table" (Table VI) was imported with the name of the stocks of the portfolio and the name of the pair[13] they belong.

The last table used on the project is called "Models" (Table VII) and to its base structure, will be added columns for each step of the creation of the models.

TABLE VI
STRUCTURE OF TABLE "PAIRS TABLE"

Pairs Table	
Pair Name	Stock
ADBE-MSFT	ADBE
⋮	⋮

TABLE VII
BASE STRUCTURE OF TABLE "MODELS"

Models					
Date	Pair	Spread	Average	Std Dev	N. Spread
2/01/2019	ADP-INTU	2.477	2.768	0.343	-0.845
⋮	⋮	⋮	⋮	⋮	⋮

There is a many-to-many relationship between the 'Date' columns in "Real Values", "5Days Forecasting" and "Norm Spreads". Also, there is a one-to-many relationship between the 'Stock" column in the "Pairs Table" with "5Days Forecasting" and "Real Values".

B. *Threshold Based Model*

To create the simple threshold-based model, four columns were added to the "Models" table, one for each threshold ('Long Threshold' with a constant value of '-2' and a 'Short Threshold' with a constant value of '2'). The third auxiliary column is an index that indicates the number of each row (after being sorted by pair and afterwards by date).

After developing the logic needed in Power BI, the "Models" table had more than 3000 rows and 21 columns. It is impossible to understand how the model performed or to get any insightful detail. Power BI, besides all the data processing capabilities that it offers, is an excellent tool to get clean and custom made visualisations of our data. In figure 8 it is represented the threshold-based model's performance for all six pairs in the selected portfolio. For each pair, it is presented the normalized spread, both long and short thresholds and in a purple dashed line the market position[14] taken throughout the period of 2019 and 2020.

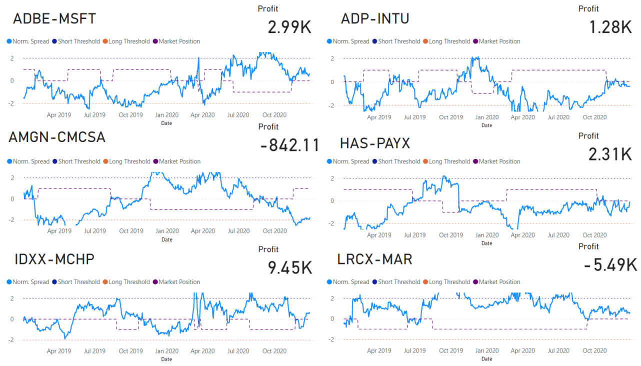

Fig. 8. Application of the Threshold based model in Power BI

[12] 2018 data is only used to calculate all the indicators that need previous days to calculate its values

[13] the name of the pair is always the concatenation of the two stocks, however, this step makes it easier to create some graphs and filters in Power BI

[14] the market position line can only take three values: 0 (no position held), 1 (long position) and -1 (short position)

C. Enhanced Model

The proposed model will use some of the data that has already been processed in the calculation of the threshold-based model. Following the architecture depicted in figure 6, both decisions that need to be made (open and close a market position) have as a key indicator, the day of best spread.

Using the information from the table ”5Days Forecasting”, it is possible to calculate the spread of each one of the following 5 days. In this formula it is being used the spread and not the normalized spread, given that ultimately it is the spread’s variance that indicates the profitability of a transaction. Two simple DAX functions were created to return the day of maximum and minimum spread. These two values will be of huge importance when selecting the best day to open/close a position.

There are five main steps when deciding when to open or close a certain market position. The first one is the stop-loss function that if the spread is bigger (in absolute value) than 2 (value of both thresholds), with a divergence of the spread of over 75% and the position has been opened for over 100 days, will automatically close avoiding further losses.

A second decision is made whenever the ’Norm. Spread’ reaches near the threshold. For example, if it reaches near the ’Long Threshold’ (using -1.95 as an example) if the ’minSpread_ID’ has a value of 0, it means that the forecasting is predicting that the spread will revert to zero, indicating that a long position should be opened. A similar process is used to open short positions. In both cases, a margin of 0.15 was added on both thresholds to detect new opportunities.

As the ’Norm. Spread’ reverses to zero and crosses the ”riskMargin” variable, and using the same logic regarding the ’minSpread_ID’ and ’maxSpread_ID’, the position may be closed. This clause aims to close the position whenever the ’Norm. Spread’ is the closest to zero, even though it may never cross it.

In the eventuality that the ’Norm. Spread’ crosses the zero line, the position will be closed. In the same manner, if there isn’t a position opened when the ’Norm. Spread’ reverts back to zero and crosses any of the long or short thresholds, the position is opened to avoid losing out on an opportunity. This last case can happen if the forecasting model believes that the ’Spread’ will keep diverging when in fact it converges.

To all these information, it is added a few columns for profit calculation and afterwards, and similarly with the threshold-based model, for this new one, was created a dashboard (figure 9) to track the model’s behaviour throughout the test period. The profit is shown in green indicates that it is better than the profit achieved in the previous model. In the following section it will be made a direct comparison between all transactions to understand better where the new enhanced model outperformed the threshold-based one.

VIII. Results

A. Pairs Selection

Regarding the pair’s selection, the aim of the used process described in VI-A was to select a short number of pairs that would allow the implementation of a pairs trading strategy. It is worth mentioning that during the test period, a worldwide pandemic crisis (Covid-19) started and it had a huge influence on the behaviour of the stock market. Due to this, the results will be evaluated in two periods of time: ”Pre-Covid” (from Jan-2019 to Feb-2020, all-inclusive) and ”Covid” (From Mar-2020 to Dec-2020 all-inclusive).

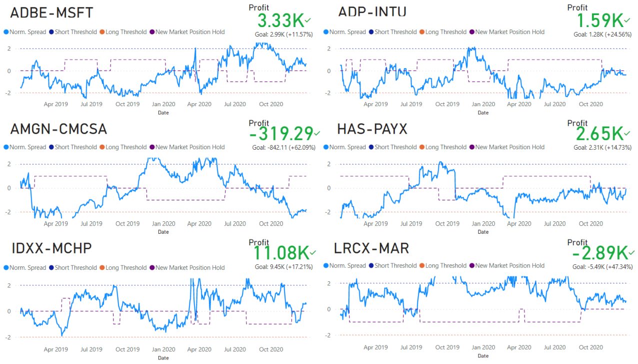

Fig. 9. Application of the Enhanced Model in Power BI

The following tables will compare the Standard Deviation of the spread with the Simple Threshold-Based model.

TABLE VIII
PRE-COVID STANDARD DEVIATION OF SPREAD DURING TEST PERIOD

Pair Name	Standard Deviation	STBM Profit (USD)
ADBE-MSFT	0.14	717.73
ADP-INTU	0.02	2120
AMGN-CMCSA	**0.48**	**-30.55**
HAS-PAYX	0.12	1740
IDXX-MCHP	0.29	1910
LRCX-MAR	0.09	1470

TABLE IX
COVID STANDARD DEVIATION OF SPREAD DURING TEST PERIOD

Pair Name	Standard Deviation	STBM Profit (USD)
ADBE-MSFT	0.11	2990
ADP-INTU	0.07	1280
AMGN-CMCSA	**0.68**	**-842**
HAS-PAYX	0.15	2310
IDXX-MCHP	0.42	9450
LRCX-MAR	**0.93**	**-5490**

In table VIII it is clear to see that in general, all pairs behaved as expected resulting in profit when applying the simple model. During the pandemic situation, the standard deviation generally increased as is expected, and in two of the pairs, this instability resulted in big losses for the investor.

B. Model Behaviour

As briefly demonstrated in Figure 9 the overall results of the enhanced model were great. However, during this section, a deeper evaluation of the model’s performance will be done.

The first result to be evaluated is the percentage of Portfolio Decline Days, which means the number of days when the return on investment was negative.

As expected, during the Covid period, the percentage of portfolio decline days increased when comparing to the Pre-Covid period. Also, and even though the improvement wasn’t

TABLE X
PORTFOLIO DECLINE DAYS WHEN USING THE STBM

Time Period	Simple Threshold Based Model # of Opened Position Days	 # of Decline Days	 % of Decline Days
Pre-Covid	923	503	54%
Covid	840	553	65%
TOTAL	1763	1056	**60%**

TABLE XI
PORTFOLIO DECLINE DAYS WHEN USING THE ENHANCED MODEL

Time Period	Enhanced Model # of Opened Position Days	 # of Decline Days	 % of Decline Days
Pre-Covid	937	527	56%
Covid	772	449	58%
TOTAL	1709	976	**57%**

that big, with the enhanced model it was possible to decline a bit the amount of portfolio decline days.

Resuming Figures 8 and 9 in Table XII it is clear to see that the profitability increased in all 6 pairs when applying the enhanced model.

TABLE XII
PROFIT COMPARISON BETWEEN BOTH MODELS

Pair Name	Simple Threshold Based Model	Enhanced Model
ADBE-MSFT	2990 $	3330 $
ADP-INTU	1280 $	1590 $
AMGN-CMCSA	-842 $	-319 $
HAS-PAYX	2310 $	2650 $
IDXX-MCHP	9450 $	11080 $
LRCX-MAR	-5490 $	-2890 $

To understand better this increase in profitability, we'll have a closer look at the circumstances that led to better transactions. After reviewing all transactions made in all six pairs, these were divided into 3 main categories.

The main difference between the behaviour of both models was the amount of opened positions. As stated in table XIII, the enhanced model manages to ”find” new opportunities to enter the market that in the end increases the profitability by a lot.

TABLE XIII
NUMBER OF POSITIONS OPENED COMPARISON BETWEEN BOTH MODELS

Pair Name	Simple Threshold Based Model	Enhanced Model
ADBE-MSFT	5	6
ADP-INTU	4	5
AMGN-CMCSA	2	2
HAS-PAYX	2	3
IDXX-MCHP	4	5
LRCX-MAR	2	3

One particular example occurs in the ADBE-MSFT pair. In figure 10 it is depicted how the normalized spread behaved, and it is interesting to analyse that there was a downwards spike (around the 3^{rd} of August) where the normalized spread almost reaches 0. This would have been a great time to close this position, however, the simple model didn't do it. Due to the margins added to the enhanced model it managed to analyse the predictions for the following days and decided to close the position.

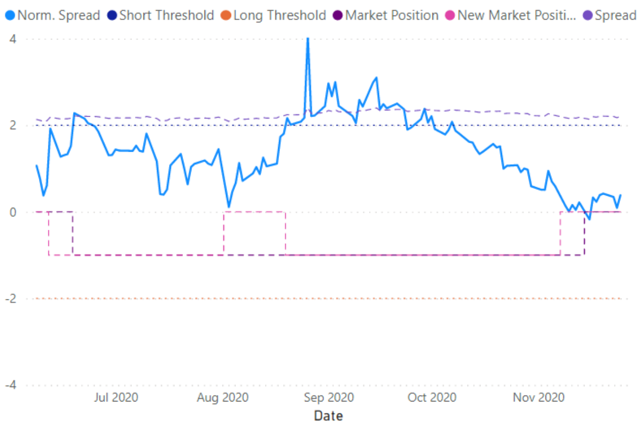

Fig. 10. New market position opened in ADBE-MSFT pair

In this particular case, the predictions for the days following the 3^{rd} of August indicated that there were no following days with a lower spread, which means that that day would be the best one to close the position. Also, and a couple of days later, the normalized spread had an upwards peak that led to another entering opportunity, allowing the enhanced model to have two different market positions while the simple model had only one. During this period, the enhanced model had more than twice as much profit as the simple one.

As stated in Tables IX and XII, and further confirmed in figure V-C, the pair LRCX-MAR's spread diverged massively in the second half of the test period. This is something that the investor could not predict and for the trading algorithm, those are just market entry opportunities. It is for cases like this, that the stop-loss function was created.

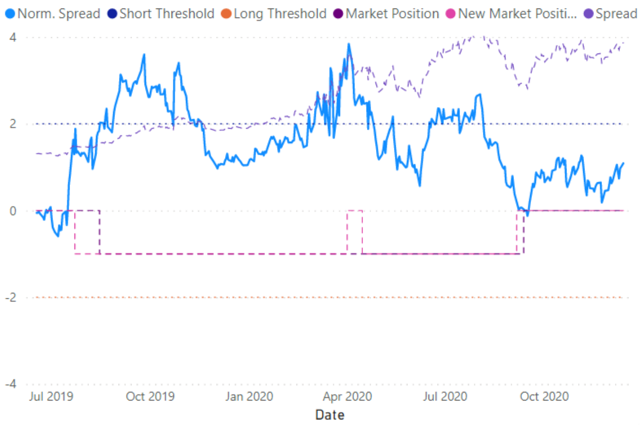

Fig. 11. Stop Loss function being activated in LRCX-MAR pair

On the 2^{nd} of April, the spread has had a 75% growth when comparing to the position opening day. Also, more than 100 days have passed since the opening which triggers the stop loss functions into closing it. Some days later, since the normalized spread is still above the short threshold, a new position is reopened. The simple fact that a huge market position was split, prevented the investor from losing almost 7k $ and would have lost 4k $ instead.

Apart from the new opportunities and the stop-loss function being activated, the two models may differ from one another in the entry and closing days of the same transaction. As explained in V-B the enhanced model, will use the predictions made by the forecasting model and try to calculate the best days to open or close a transaction.

In figure 12 it is depicted an example of a transaction where each model had its entry and closing day for that same transaction.

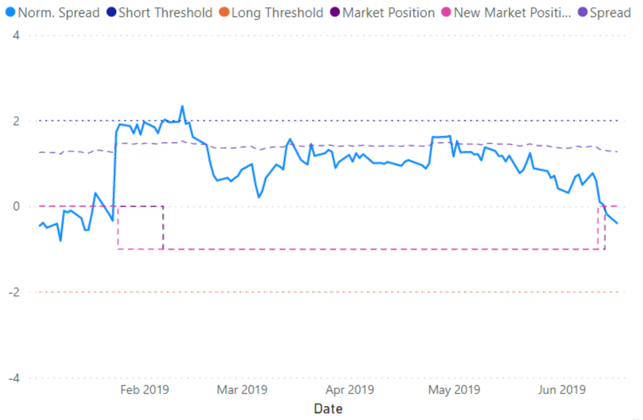

Fig. 12. Different Entry and Closing days for the same opportunity in LRCX-MAR pair

In table XIV it is shown the result of the forecasting model (two columns on the right) as well as the actions taken by each model. On the 25th of January, as the Norm. Spread got closer to the short threshold, the whole process was activated and the min and max spread days were calculated. Given that according to the forecasting model the 25th was the day of maximum spread, the enhanced model opened a short position. The closing process was similar and also resulted in the enhanced model taking action a couple of days earlier than the simple model.

TABLE XIV
KEY ACTIONS PERFORMED BY BOTH MODELS ON THE SAME OPORTUNITY

Date	Basic Model	New Model	Max Spread	Min Spread
25-01-19		Open	0	3
07-02-19	Open		0	3
12-06-19		Close	4	0
14-06-19	Close		4	0

Even though in both cases, the transaction was profitable, the fact that the spread kept growing to reach the threshold, means that the predicted max spread day was not right. The same happened with the closing day, meaning that the simple model had a better performance than the enhanced one. For this particular transaction, the enhanced model only achieved 80% of the profit achieved by the simple model.

In the end, the enhanced model outperformed the simple threshold-based one, however, that is due to the new opportunities that it found, and also due to the stop-loss function that prevented transactions lasting for too long and the spread from diverging a lot. The transactions that both models participated in, generally speaking, the enhanced model didn't perform better than the simple one. That lead to the first and one of the main conclusions of this project: not only it is hard to build an accurate forecasting model to predict the market fluctuations, but it is also even harder to have a trustworthy forecast of the best entry/closing days because it requires a division of two forecasted values that increases a lot the error they may have.

Using the values presented in tables XII and XIII, table XV was created. These values indicate that the profit per transaction also increased from one model to the other, and since the initial investment for each transaction is 10000 $, it is safe to say, that on average, the profit will be around 6.4% per transaction.

TABLE XV
PROFIT PER TRANSACTION COMPARISON BETWEEN THE TWO MODELS

	Basic Model	New Model
Total Profit (USD)	9998	15441
Number of Transactions	19	24
Profit per Transaction (USD)	526	643

The major conclusion about the performance of the enhanced model is that it can indeed outperform the simple one. However, this outperformance comes from the finding of new trading opportunities or exiting some positions to avoid major losses. Even though these two segments may be influenced by the forecasting model, it is very minor when comparing the influence that it has whenever both models take on the same opportunity. The forecasting model is the only one responsible for the difference of both models, it can't out-perform the simple model due to the error that is propagated from the prediction of both stocks to the forecasted spread[15]. In the end, both the new opportunities have a huge positive impact on the overall profit and whenever the enhanced model only relies on the forecasting model to make decisions it ends up losing when compared to the simple threshold model.

IX. Conclusions and Future Work

A. Conclusions

The main goal of this project was to enhance an investment strategy using this forecasting model. Even though the enhanced model had a better performance than the simple one, the influence of the forecasting method was close to none. To predict market fluctuations, the model should be much more complex and there will always be an error. This error can increase a lot when divided by another stock prediction that also has an error associated with it. Due to this error propagation, whenever a decision relied solely on the forecasting results, its performance would be generally worse than the regular threshold model.

The simple threshold model has two main weaknesses, the first one is that it only enters on positions that cross the thresholds, wasting great profitable opportunities. The second weakness that can incur huge losses, is the fact that the simple model can hold on to positions for a long time. Tackling these two problems can increase profitability in such a way that reduces the impact of any eventual gains from entering earlier

[15] The forecasted values of the closing price of each stock naturally have an error associated, however, since the forecasted spread is the division of the two values, this error gets even bigger

or later, based on the forecasting model. This approach had a significant boost in performance, increasing the profitability of the model by more than 54%.

B. Future Work

As a follow-up for this project, there are some different approaches that could be worth exploring.

On the pairs selection phase, there are some white-spaces, namely trying to mix and match individual stocks with indexes or ETF's (Exchange Trade Funds).

The forecasting model by training it to predict the spread directly using financial indicators of both stocks in the pair at the same time. The goal with this approach would be to decrease the error propagation and having a more trustworthy tool to reduce portfolio decline days, as well as, outperforming the standard model by entering/closing closer to the ideal entry/close point. One other approach would be to use sentiment analyses to predict the stock's price behaviour.

Regarding the trading model, the stop-loss function should be improved by keeping track of the spread's behaviour and in a worst-case scenario, leaving the pair to avoid reopening another position due to the high risk that it may have for the investor.

X. References

[1] J. P. Broussard and M. Vaihekoski, "Profitability of pairs trading strategy in an illiquid market withmultiple share classes,"Journal of International Financial Markets, Institutions and Money, vol. 22,pp. 1188–1201, 2012

[2] M. C. Blázquez, C. D. la Ordem De la Cruz, and C.P.Román, "Pairs trading techniques An empiricalcontrast, "European Research on Management and Business Economics, vol. 24, pp. 160–167, 92018.

[3] W. A. F. David A. Dickey, "Distribution of the estimators for autoregressive time series with a unitroot," 1979.

[4] J. G. Mackinnon, "Critical values for cointegration tests." [Online]. Available: http://www.econ.queensu.ca/faculty/mackinnon/

[5] J. Mackinnon, "Approximate asymptotic distribution functions for unit root and cointegration tests,"1992.

[6] Jieren Wang, C. Rostoker, and A. Wagner, "A high performance pair trading application," pp. 1–8,2009.

[7] H. C. Wang, W. C. Hsiao, and S. H. Chang, "Automatic paper writing based on a rnn and thetextrank algorithm,"Applied Soft Computing Journal, vol. 97, 12 2020

[8] M. Dhyani and R. Kumar, "An intelligent chatbot using deep learning with bidirectional rnn andattention model,"Materials Today: Proceedings, 6 2020.

[9] W.-J. Wang, Y.-F. Liao, and S.-H. Chen, "Rnn-based prosodic modeling for mandarin speech andits application to speech-to-text conversion." [Online]. Available: www.elsevier.com/locate/specom

[10] Q. Wang, W. Xu, X. Huang, and K. Yang, "Enhancing intraday stock price manipulation detectionby leveraging recurrent neural networks with ensemble learning,"Neurocomputing, vol. 347, pp.46–58, 6 2019

[11] Z. Hajiabotorabi, A. Kazemi, F. F. Samavati, and F. M. M. Ghaini, "Improving dwt-rnn model viab-spline wavelet multiresolution to forecast a high-frequency time series,"Expert Systems with Ap-plications, vol. 138, 12 2019.

[12] A. S. Saud and S. Shakya, "Analysis of look back period for stock price prediction with rnn variants:A case study on banking sector of nepse," vol. 167. Elsevier B.V., 2020, pp. 788–798