Enhancing a Pairs Trading strategy with the application of Machine Learning

Simão Moraes Sarmento
simao.moraes.sarmento@tecnico.ulisboa.pt

Nuno Horta
nuno.horta@tecnico.ulisboa.pt

***Abstract*—Pairs Trading is one of the most valuable market-neutral strategies used by hedge funds. It is particularly interesting as it overcomes the arduous process of valuing securities by focusing on relative pricing. By buying a relatively undervalued security and selling a relatively overvalued one, a profit can be made upon the pair's price convergence. However, with the growing availability of data, it became increasingly harder to find rewarding pairs. In this work we address two problems: (i) how to find profitable pairs while constraining the search space and (ii) how to avoid long decline periods due to prolonged divergent pairs. To manage these difficulties, the application of promising Machine Learning techniques is investigated in detail. We propose the integration of an Unsupervised Learning algorithm, OPTICS, to handle problem (i). The results obtained demonstrate the suggested technique can outperform the common pairs' search methods, achieving an average portfolio Sharpe ratio of 3.79, in comparison to 3.58 and 2.59 obtained by standard approaches. For problem (ii), we introduce a forecasting-based trading model, capable of reducing the periods of portfolio decline by 75%. Yet, this comes at the expense of decreasing overall profitability. The proposed strategy is tested using an ARMA model, an LSTM and an LSTM Encoder-Decoder. This work's results are simulated during varying periods between January 2009 and December 2018, using 5-minutes price data from a group of 208 commodity-linked ETFs, and accounting for transaction costs.**

***Index Terms*—Pairs Trading, Market Neutral, Machine Learning, Deep Learning, Unsupervised Learning**

I. Introduction

Pairs Trading is a popular trading strategy widely used by hedge funds and investment banks. It is capable of obtaining profits irrespective of the market direction.

This is accomplished with a two-step procedure. First, a pair of assets whose prices have historically moved together is detected. Then, assuming the equilibrium relationship should persist in the future, the spread between the prices of the two assets is monitored and in case it deviates from its historical mean the investor shorts the overvalued asset and buys the undervalued one. Both positions are closed upon price convergence.

However, with the growing availability of data, it is becoming increasingly harder to find robust pairs. In this work, we address two problems in specific: (i) how to find profitable pairs while constraining the search space and (ii) how to avoid long decline periods due to prolonged divergent pairs.

The remainder of this document is organized as follows: in section II we introduce the main concepts of Pairs Trading while describing the associated research work. In section III we suggest a new pairs selection framework to address the first problem motivating this research work. In section IV we propose a new trading model in response to the second problem on the origin of this work. Next, in section V we design the simulation environment to test the proposed approaches, for which the results are presented in section VI.

II. Background and Related Work

Each stage composing a Pairs Trading strategy is described in detail along with the most relevant related work.

A. Pairs Selection

The pairs selection stage encompasses (i) finding the appropriate candidate pairs and (ii) selecting the most promising ones.

Starting with (i), the investor should select the securities of interest (e.g stocks, ETFs, etc) and search for possible combinations. In the literature, two methodologies are typically suggested for this stage: performing an exhaustive search for all possible combinations among the selected securities, or grouping them by sector, and constrain the combinations to pairs formed by securities within the same sector. While the former may find more unusual interesting pairs, the latter reduces the likelihood of finding spurious relations. For example, [1, 2] impose no restriction on the universe from which to select the pairs. Contrarily, some research work, as [3–5] arranges the securities on category groups and select pairs within the same group.

Concerning (ii), the investor must define what criteria should be used to select a pair. The most common approaches are the distance, correlation, and cointegration approaches.

The distance approach, suggested in [3], selects pairs which minimize the historic sum of squared distances between the two assets' price series. This method is widely used but according to [6] it is analytically sub optimal. If $p_{i,t}$ is a realization of the normalized price process $P_i = \underline{(P_{i,t})}_{t\in T}$ of an asset i, the average sum of squared distances $\overline{ssd}_{P_i,P_j}$ in the formation period[1] of a pair formed by assets i and j is given by

$$\overline{ssd}_{P_i,P_j} = \frac{1}{T}\sum_{t=1}^{T}\left(p_{i,t} - p_{j,t}\right)^2 . \tag{1}$$

Thus, an optimal pair would be one that minimizes Eq.(1). However, this implies a zero spread pair is considered optimal

[1] The formation period corresponds to the period in which securities are being analyzed to form potential pairs.

which logically may not be as it would not provide trading chances.

The application of Pearson correlation as a selection metric is analyzed in [7]. The authors examine its application on return series with the same data sample used in [3] and find that correlation shows better performance, with a reported monthly average of 1.70% raw returns, almost twice as high as the one obtained using the distance approach. Nevertheless, this criteria is not foolproof as two return level correlated securities might not share an equilibrium relationship, and divergence reversions cannot be explained theoretically.

At last, the cointegration approach entails selecting pairs for which the two constituents are found to be cointegrated. If two securities, Y_t and X_t are found to be cointegrated, then by definition, the series constructed as

$$S_t = Y_t - \beta X_t, \tag{2}$$

where β is the cointegration factor, must be stationary. Defining the spread series in this way is particularly convenient since under these conditions the spread is expected to be mean-reverting, meaning that every spread divergence is expected to be followed by convergence. Hence, this approach finds econometrically more sound equilibrium relationships. The most cited work in this field is [8], that proposes a set of heuristics for cointegration based strategies. Furthermore, [9] performs a comparison study between the cointegration approach and the distance approach and finds that the cointegration approach significantly outperforms the distance method.

B. Trading Models

The most common trading model follows from [3], and can be described as indicated below:

i Calculate the pair's spread ($S_t = Y_t - X_t$) mean, μ_s, and standard deviation, σ_s, during the formation period.

ii Define the model thresholds: the threshold that triggers a long position, α_L, the threshold that triggers a short position, α_S, and the exit threshold, α_{exit}, that defines the level at which a position should be exited.

iii Monitor the evolution of the spread, S_t, and control if any threshold is crossed.

iv In case α_L is crossed, go long the spread by buying Y and selling X. If α_S is triggered, short the spread by selling Y and buying X. Exit position when α_{exit} is crossed.

The simplicity of this model is particularly appealing, motivating its frequent application in the field. Nonetheless, the entry points defined may not be optimal since no information concerning the spread subsequent direction is incorporated in the trading decision. Some efforts have emerged trying to propose more robust models. Techniques from different fields, such as stochastic control theory, statistical process modelling and Machine Learning have been studied. In particular, the results obtained by Machine Learning approaches have proved very promising. Dunis et al. [10, 11] explore the application of Artificial Neural Networks to forecast the spread change for two famous spreads. Thomaidis et al. [12] propose an experimental statistical arbitrage system based on Neural Network Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models for modeling the mispricing-correction mechanism between relative prices composing a pair. Huck [13], Huck [14] uses RNNs to generate a one-week ahead forecast, from which the predicted returns are calculated. Lastly, Krauss et al. [1] analyze the effectiveness of deep neural networks, gradient-boosted-trees and random forests in the context of statistical arbitrage using S&P 500 stocks. Apart from this, Machine Learning techniques still remain fairly unexplored in this field and the results obtained indicate this is a promising direction for future research.

III. Proposed Pairs Selection Framework

At this research stage we aim to explore how one investor may find promising pairs without exposing himself to the adversities of the common pairs searching techniques. On the one hand, if the investor limits its search to securities within the same sector he is less likely to find pairs not yet being traded in large volumes, leaving a small margin for profit. But on the other hand, if the investor does not impose any limitation on the search space, he might have to explore excessive combinations and possibly find spurious relations.

We intend to reach an equilibrium with the application of an Unsupervised Learning algorithm, on the expectation that it will infer meaningful clusters of assets from which to select the pairs.

A. Dimensionality reduction

The first step towards this direction consists in finding a compact representation for each asset, starting from its price series. The application of Principal Component Analysis (PCA) is proposed. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables, the principal components. Each component can be seen as representing a risk factor. We suggest the application of PCA in the normalized return series, defined as

$$R_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}}, \tag{3}$$

where $P_{i,t}$ is the price series of a asset i. Using the price series could result in the detection of spurious correlations due to underlying time trends. The number of principal components used defines the number of features for each asset representation. Considering that an Unsupervised Learning algorithm will be applied to these data, the number of features should not be large. High data dimensionality presents a dual problem. The first being that in the presence of more attributes, the likelihood of finding irrelevant features increases. Additionally, there is the problem of the curse of dimensionality, caused by the exponential increase in volume associated with adding extra dimensions to the space. According to [15], this effect starts to be severe for dimensions greater than 15. Taking this into consideration, the number of PCA dimensions is upper bounded at this value and is chosen empirically.

B. Unsupervised Learning clustering

Having constructed a compact representation for each asset, a clustering technique may be applied. To decide which algorithm is more appropriate, some problem-specific requisites are first defined:

– No need to specify the number of clusters in advance.
– No need to group all securities.
– Strict assignment that accounts for outliers.
– No assumptions regarding the clusters' shape.

The assignment should be strict, otherwise it would increase the number of combinations when looking for pairs, which is conflicting with the initial goal. Also, by making the number of clusters data-driven, we introduce as little bias as possible. In addition, outliers should not be incorporated in the clusters, and therefore grouping all assets should not be enforced. Finally, due to the nonexistence of prior information that indicates the clusters should be regularly shaped, the selected algorithm should not adopt this assumption.

Taking into consideration the previously described requirements, a density-based clustering algorithm seems an appropriate choice. It forms clusters with arbitrary shapes and thus no gaussianity assumptions need to be adopted. It is naturally robust to outliers as it does not group every point in the data set. Furthermore, it requires no specification of the number of clusters.

The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm is the most influential in this category. Briefly, DBSCAN detects clusters of points based on their density. To accomplish that, two parameters need to be defined: ε, which specifies how close points should be to each other to be considered "neighbors", and $minPts$, the minimum number of points to form a cluster. From these two parameters, in conjugation with some concepts that we omit here[2], clusters of neighboring points are formed. Points falling in regions with less than $minPts$ within a circle of radius ε are classified as outliers, hence not affecting the results. In spite of the advantages stated so far, DBSCAN still carries one drawback. The algorithm is appropriate under the assumption that clusters are evenly dense. However, if regions in space have different densities, a fixed ε may be well adapted to one given cluster density but it might be unrealistic for another, as depicted in Figure 1. It is evident that cluster A, B, and C could eventually be found using the same ε, but A_1 and A_2 would not be distinguished.

The OPTICS algorithm proposed in [17] addresses this problem. OPTICS is based on DBSCAN, with the introduction of some important concepts that enable a varying ε implementation. In this enhanced setting, the investor is only required to specify the parameter $minPts$, as the algorithm is capable of detecting the most appropriate ε' for each cluster[3]. Therefore, we propose using OPTICS not just to account for varying cluster densities but also to facilitate the investor's task.

[2]Interested readers may refer to [16].
[3]This description is very simplified. We suggest the interested readers refer to [17].

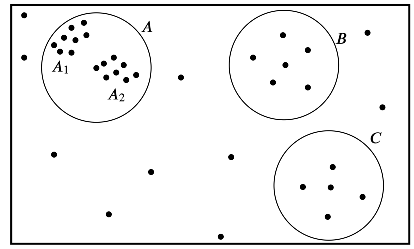


Fig. 1. Clusters with varying density. Adapted from: [17]

C. Pairs selection criteria

Having generated the clusters of assets, it is still necessary to define a set of rules for selecting the pairs to trade. It is critical that the pairs' equilibrium persists. To enforce this, we propose the unification of methods applied in separate research work. According to the proposed criteria, a pair is selected if it complies with the four conditions described next. First, a pair is only deemed eligible for trading if the two securities forming the pair are cointegrated. To test this condition, we propose the application of the Engle-Granger test due to its simplicity. To protect from the test reliance on the dependent variable, we propose that the test is run for both possible selections of the dependent variable, and that the combination generating the lowest t-statistic is selected. Secondly, to provide more confidence in the mean-reversion character of the spread, an extra validation step is suggested. We resort to the concept of Hurst exponent, H, which quantifies the relative tendency of a time series either to regress strongly to the mean or to follow a trend [18]. If H belongs to the range 0–0.5 it indicates that a time series is mean-reverting. Hence, we require that a pairs' spread Hurst exponent is less than 0.5. In third place, we intend to discard stationary pairs with unsuitable timings. A mean-reverting spread by itself does not necessarily generate profits. There must be coherence between the mean-reversion time and the trading period. The half-life of mean-reversion is an indicator of how long it takes for a time series to mean-revert [19]. Therefore, we propose filtering out pairs for which the half-life takes extreme values: less than one day or more than one year. Lastly, we suggest enforcing that every spread crosses its mean at least twelve times per year, enforcing one trade per month on average.

D. Framework diagram

The three building blocks of the proposed framework have been described. Figure 2 illustrates how they connect. As we may observe, the initial state should comprise the price series for all the possible pairs' constituents. We assume this information is available to the investor. Then, by reducing the data dimensionality, each security may be described not just by its price series but also by the compact representation emerging from the application of PCA in the return series (State 1). Using this simplified representation, the OPTICS algorithm is capable of organizing the securities into clusters (State 2). Finally, we may search for pair combinations within the clusters and select those that verify the rules imposed.

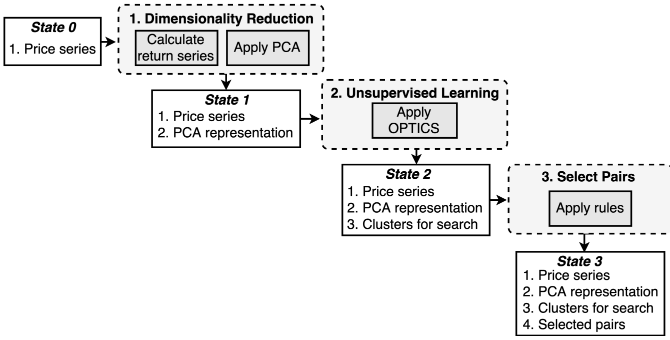


Fig. 2. Pairs selection diagram.

IV. Proposed Trading Model

We proceed to address the second problem this work intends to explore: handling the long decline periods due to prolonged divergent pairs.

A. Trading Model

A potential alternative to continuously monitor the spread and track deviations consists of modeling the spread directly. This way, a prediction can be made regarding how the spread will vary in the future and a position is only entered if the predicted conditions are favourable. By taking advantage of a time-series forecasting algorithm to predict the spread at the next time instant, we may calculate the expected spread percentage change at time $t+1$ as

$$\Delta_{t+1} = \frac{S^*_{t+1} - S_t}{S_t} \times 100, \tag{4}$$

where S and S^* correspond to the real and the predicted spread, respectively. When the absolute value of the predicted change is larger than a predefined threshold, a position may be entered, on the expectation that the spread will suffer an abrupt movement from which the investor can benefit from. Assuming the investor is not holding a position yet, the next position, P_{t+1}, may be described according to

$$P_{t+1}: \begin{cases} \text{if } \Delta_{t+1} \geq \alpha_L, & \text{Go long} \\ \text{if } \Delta_{t+1} \leq \alpha_S, & \text{Go short}\ . \\ \text{otherwise}, & \text{Wait} \end{cases} \tag{5}$$

Once a position is entered, it is maintained while the predicted spread direction does not change. When it switches, the position is exited. This strategy defines the basis of the proposed trading model. It is still to be described how the thresholds (α_L, α_S) should be calculated. A possible approach consists of framing an optimization problem, and try to find the profit-maximizing values. However, this approach is rejected due to its risk of data-snooping and unnecessary added complexity. We propose a simpler, non-iterative, data-driven approach. We start by obtaining $f(x)$, the spread percentage change distribution during the formation period, given that the spread percentage change at time t is defined as

$$x_t = \frac{S_{t+1} - S_t}{S_t} \times 100.$$

From $f(x)$, the set of negative percentage changes, $f^-(x)$, and positive percentage changes, $f^+(x)$, are considered separately. Since the proposed model targets abrupt changes but also requires that they occur frequently enough, looking for the extreme quantiles seems an adequate solution. Therefore, we recommend using the top decile and quintile from $f^+(x)$ as candidates for defining α_L and the bottom ones, from $f^-(x)$, for defining α_S. The quintile-based and decile-based thresholds are both tested in the validation set and the most optimistic combination is adopted. Formally,

$$\begin{aligned} \{\alpha_S, \alpha_L\} &= \underset{q}{\operatorname{argmax}}\ R^{\text{val}}(q), \\ q \in \Big[&\{Q_{f^-(x)}(0.20),\, Q_{f^+(x)}(0.80)\} \\ &\{Q_{f^-(x)}(0.10),\, Q_{f^+(x)}(0.90)\} \Big] \end{aligned} \tag{6}$$

where R^{val} is the return obtained in the validation period.

To summarize, the model construction follows the diagram illustrated in Figure 3. For each pair, the investor starts by training the forecasting algorithms to predict the spread. Furthermore, the decile-based and quintile-based thresholds are collected to integrate the trading model. Having fitted the forecasting algorithms and obtained the two combinations for the thresholds (State 1), the model is applied on the validation set. From the validation performance, the best threshold combination is selected (State 2). At this point the model is ready to be applied on unseen data.

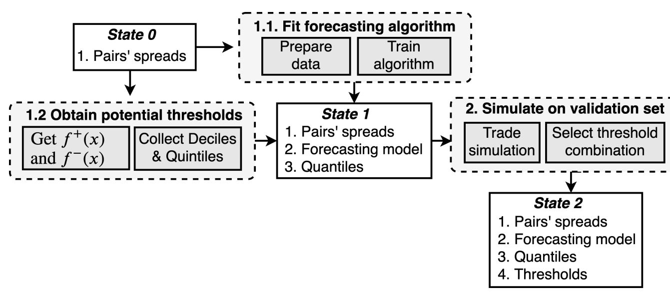


Fig. 3. Proposed model construction diagram.

An application example is illustrated in Figure 4. For the sake of illustration, the forecasting has perfect accuracy, meaning the positions can be set in optimal conditions.

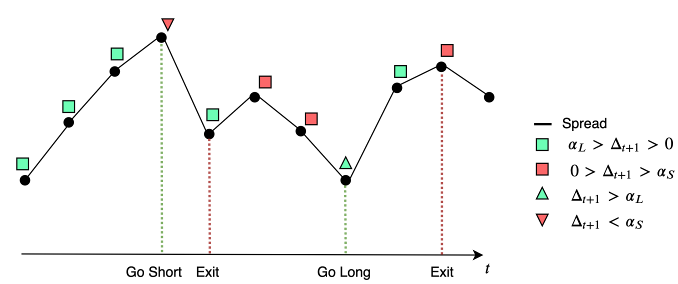


Fig. 4. Example of the proposed forecasting-based strategy.

B. Applied forecasting algorithms

Forecasting algorithms commonly applied in the literature can be divided into two major classes: parametric and non-parametric models. The former assumes that the underlying process can be described using a small number of parameters. The latter makes no structural assumptions about the underlying structure of the process. We propose the application of a benchmark parametric approach, the autoregressive–moving-average (ARMA), and two non-parametric models, the Long Short-Term Memory (LSTM) and the LSTM Encoder-Decoder. This will allow inferring to what extent the strategy profitability depends on the complexity of the time-series forecasting algorithm. The justification for choices adopted are described next.

Although financial time series are very complex in nature [20], the ones under analysis are by construction stationary, as they correspond to the linear combination of cointegrated price series. Thus it is fair to ask if an ARMA model may succeed at forecasting this series. This model describes a stationary stochastic process as the composition of two polynomials, the autoregression AR(p) and the moving average MA(q), as

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}, \tag{7}$$

where p and q represent the order of the polynomials.

Nevertheless, there is an underlying motivation for applying more complex models, such as Artificial Neural Networks (ANN). First, ANNs have been an object of attention in many different fields, which makes its application in this context an interesting case study. Furthermore, ANN-based models have shown very promising results in predicting financial time series data in general [21]. From the vast amount of existing ANN configurations, the LSTM architecture is deemed appropriate due to its capabilities of learning non-linear representations of the data while memorizing long sequences. LSTMs assume the existence of a sequential dependency among inputs, and previous states might affect the decision of the neural network at a different point in time.

Furthermore, from a trading perspective, it might be particularly beneficial to collect information regarding the prediction not just of the next instant of a time-series but also of later time steps. An LSTM Encoder-Decoder architecture is naturally fitted to such scenarios. This architecture is comprised by two LSTMs, one for encoding the input sequence into a fixed-length vector, the encoder, and a second for decoding the fixed-length vector and outputting the predicted sequence, the decoder, as illustrated in Figure 5.

In this multi-step forecasting scenario, the trading rules are adapted by simply calculating the prediction change N times in advance. Likewise, the thresholds α_L and α_S should be calculated with respect to the distribution of the percentage change between $x(t+N)$ and $x(t)$.

Given the limited computation resources, the neural network models' tuning is constrained to a set of most relevant variables: data sequence length, the number of hidden layers and the nodes in each hidden layer. Early-stopping and dropout are applied as regularization techniques.

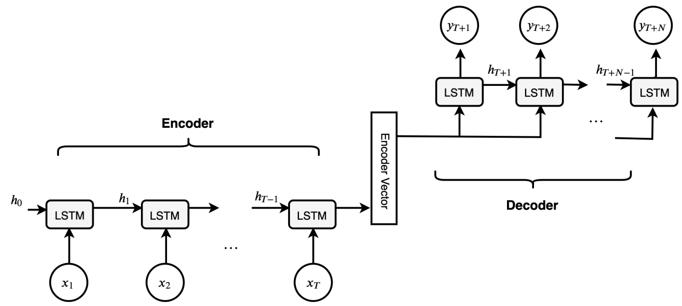


Fig. 5. LSTM Encoder-Decoder.

V. Research design

The research design contemplates two stages, corresponding to each problem being addressed.

A. Research Stage 1 - Pairs selection

First, we intend to examine how the three different pairs' search techniques (unrestricted, grouping by category and unsupervised learning) compare to each other. For this purpose, the three methodologies are implemented. The proposed pairs selection rules are also constructed and applied on top of each search technique. As for the trading setup, since the focus lies on comparing the search techniques relative to each other, we do not concern about meticulously optimizing the trading conditions. Therefore, we apply the standard threshold-based model proposed in [3], with the parameters specified in Table I. The spread's standard deviation, σ_s, and mean, μ_s, are calculated with w.r.t to the entire formation period.

TABLE I
Threshold-based model parameters.

Parameters	**Values**
Long Treshold	$\mu_s - 2\sigma_s$
Short Threshold	$\mu_s + 2\sigma_s$
Exit Threshold	μ_s

To test the performance of the selected pairs, we implement three different test portfolios resembling probable trading scenarios. Portfolio 1 considers all the pairs identified in the formation period. Portfolio 2 takes advantage of the feedback collected from running the strategy in the validation set by selecting only the pairs that had a positive return. Lastly, Portfolio 3 corresponds to the situation in which the investor is limited to invest in a fixed number of k pairs. In such case, we suggest selecting the top-k pairs according to the return obtained in the validation set. We consider $k = 10$, as it stands in between the choices of [3], which uses $k = 5$ and $k = 20$.

B. Research Stage 2 - Trading Model

At this stage, we aim to compare the robustness provided by the standard threshold-based model with the proposed forecasting-based model, simulated using an ARMA, an LSTM and an LSTM Encoder-Decoder with an output length of two. We propose to first evaluate the forecasting performance of each algorithm. As benchmark, a naive baseline is considered, which simply outputs $Y_{t+1} = Y_t$. Then, to evaluate the trading strategy itself, we suggest using the pairs search technique which proved more appealing according to the results obtained in the previous research stage. As for the test portfolio, we consider using Portfolio 2.

C. Dataset

Trading ETFs is considered adequate since they are easy to trade, as they trade like stocks, and because their dynamics are expected to change much slower than that of a single stock. Adding to that, research in the field [7, 22] obtained more robust mean-reverting time series by using a linear combination of stocks to form each component of the spread. We presume using ETFs may be a proxy to accomplish that more efficiently.

This work fixates a subset of ETFs which track single commodities, commodity-linked indexes or companies focused on exploring a commodity. This reduces the number of possible pairs, making the strategy computationally more efficient and leaving space for careful analysis of the selected pairs.

A total of 208 commodity-linked ETFs are available for trading in January 2019, for which five categories may be identified based on the ETFs composition (Agriculture, Broad Market, Energy, Industrial Metals and Precious Metals). This information is collected from [23].

We considered price series data with 5-min frequency. The motivation for using intraday data is three-fold. First, with finer granularity the entry and exit points can be defined with more precision, providing opportunities for higher profit margins. Secondly, we may detect intraday mean-reversion patterns that could not be found otherwise. At last, it provides more data samples, allowing to train complex forecasting models with less risk of overfitting.

The periods considered for simulating each research stage are illustrated in Figure 6. There are essentially two possible configurations: (i) the 3-year-long formation periods, and (ii) the 9-year-long formation period. In both cases, the second-to-last year is used for validating the performance, before running the strategy on the test set. We define a 1-year-long trading period, based on the findings of [4], that claim the profitability can be increased if the initial 6-month trading period proposed in [3] is extended to 1 year.

Configuration (i) is adopted when using the threshold-based trading model (described in Table I). A formation period of three years seems appropriate. Although this period is slightly longer than what is commonly found in the literature[4], we decide to proceed on the basis that a longer period may identify more robust pairs. Configuration (ii) is used for simulating the forecasting-based trading model, thus providing more formation data to fit the forecasting algorithms. In this case, the first 8 years are used for training, as indicated in Figure 6.

For the first research stage, we propose using three different periods to have more statistical evidence on the results obtained. In the second research stage, this is not conceivable due to the computational burden of training the forecasting algorithms. Hence, we consider one period using configuration (i), and a second period using configuration (ii), to evaluate how the standard model could have performed in the same test period.

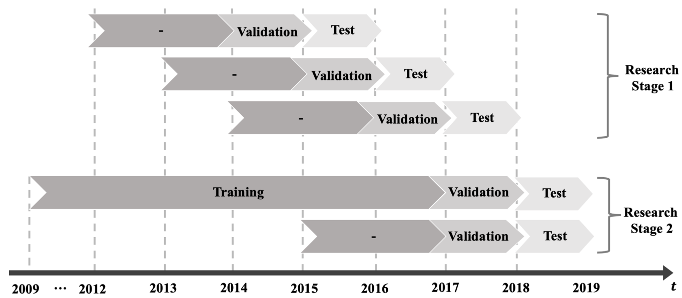


Fig. 6. Trading periods.

As preprocessing steps, we start by removing all ETFs with missing values throughout the period being considered. Then, we remove ETFs that do not verify the minimum liquidity requisites to ensure the considered transaction costs associated with bid-ask spread are realistic[5]. The minimum liquidity requisites follow the criterion adopted in [3, 25], which discards all ETFs not traded during at least one day.

D. Trading simulation

Concerning the portfolio construction, we impose that all pairs are equally weighted in the portfolio, so that the returns can be obtained dividing the performance by the number of pairs, with no need to concern about relative weights.

An aspect that follows concerns the capital allocation within each pair. We consider that the capital resulting from the short position is immediately applied in the long position. This type of leverage is adopted by most hedge funds. On this basis, we construct a framework which ensures that every trade is set with just \$1. This is accomplished by imposing that

$$\max(\text{asset}_1, \text{asset}_2) = \$1,$$

where asset_1 and asset_2 represent the capital invested in each pair's constituent, as illustrated in Figure 7. Although the gross exposure is higher, a \$1 dollar initial investment is always sufficient. As trading progresses, we consider that all the capital earned by a pair in the trading period is reinvested in the next trade.

[4][3, 4, 24] use a 1-year-long formation period. [5] makes use of a 3-month formation period.

[5]Trading illiquid ETFs would result in higher bid-ask spreads which could dramatically impact the profit margins.

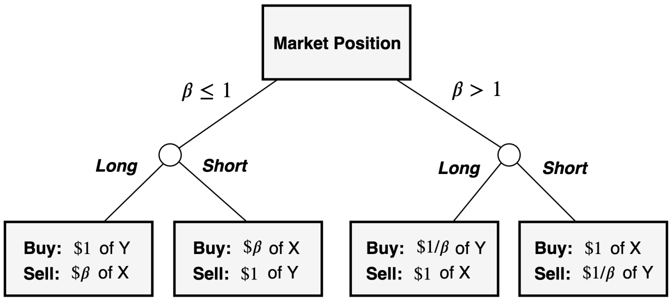


Fig. 7. Market position definition.

All the results presented in this work account for transaction costs. The transaction costs considered are based on estimates from [25], in which the authors perform an in-depth study on the impact of transaction costs in Pairs Trading. The costs comprise three components: commissions (8 bps), market impact (20 bps) and short-selling constraints (1% per annum). Besides, commission and market impact costs are adapted to account for both assets in the pair.

As a trading system can not act instantaneously, there might be a small deviation in the entry price inherent to the delay of entering a position. To account for this factor and make sure the strategy is viable in practice, we assume a conservative one period (5-min) delay for entering a position.

This work does not comprehend an implementation of a stop-loss system, under any circumstances. This means a position is only exited if the pair converges or the trading period ends.

E. Evaluation metrics

Regarding the trading evaluation, we propose analyzing the strategy Return on Investment (ROI), Sharpe Ratio (SR) and the portfolio Maximum Drawdown (MDD).

The ROI is calculated as the net profit divided by the initial capital, which we enforced to be \$1.

The portfolio SR is calculated as

$$SR_{\text{year}} = \frac{R^{\text{port}} - R_f}{\sigma_{\text{port}}} \times \text{annualization factor}, \tag{8}$$

where R^{port} represents the daily portfolio returns and R_f the risk-free rate[6]. The portfolio volatility, σ_{port}, is calculated as

$$\sigma_{\text{port}} = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_i \operatorname{cov}(i,j) \omega_j}, \tag{9}$$

where w_i is the relative weight of asset i in the portfolio. The annualization factor is set according to the methodology proposed by Lo [27] (Table 2 in [27]), to prevent imprecise approximations.

[6]The average the 3-Month treasury bill rate, taken from [26], during the corresponding test period and converted to a daily basis for consistency with the formula.

F. Implementation environment

All the code developed in this work is built from scratch using Python. Some libraries are particularly useful. First, *sci-kit learn* proves helpful in the implementation of PCA and the OPTICS algorithm. Second, *statsmodels* provides an already implemented version of the ADF test, useful for testing cointegration. Last, we make use of *Keras* to build the proposed neural networks. The code is publicly available in [28].

Concerning the running environment, most of the simulation code is run on a local CPU, except for the training of the LSTM models. They involve a huge amount of matrix multiplications which result in long processing times. These operations are massively sped up by taking advantage of the parallelization capabilities of a GPU.

VI. Results

The results obtained at each research stage are presented next.

A. Analysis of the pairs selection framework

We start by presenting some relevant statistics in Table II concerning the number of pairs found for the three different pairs search techniques being compared at this stage.

TABLE II
SELECTED PAIRS USING DIFFERENT SEARCH METHODS.

	Formation Period	2012-2015	2013-2016	2014-2017
Clustering Type	**No Clustering**			
	Number of clusters	1	1	1
	Possible combinations	4465	5460	6670
	Pairs selected	101	247	150
	Category			
	Number of clusters	5	5	5
	Possible combinations	2612	3318	4190
	Pairs selected	59	51	51
	OPTICS			
	Number of clusters	9	13	12
	Possible combinations	185	140	129
	Pairs selected	39	40	18

As expected, when no restrictions are imposed in the search space, a larger set of ETFs emerges and consequently more pairs are selected. Contrarily, when grouping ETFs in five partitions (according to the categories described in section V-C) there is a reduction in the number of possible pair combinations. This is not more evident due to the underlying unbalance across the categories considered. Because energy linked ETFs represent close to half of all ETFs, the combinations within this sector are still vast. Lastly, the number of possible pair combinations when using OPTICS is remarkably lower. Although the number of clusters is higher than when grouping by category, their smaller size results in fewer combinations. We proceed to analyze in more detail the results obtained with this algorithm.

The results concerning the OPTICS application are obtained using five principal components to describe the data. We empirically verified that up to the 15-dimensions boundary (motivated in section III-A) the results are not significantly affected. We adopt 5 dimensions since we find more adequate to settle the ETFs' representation in a lower dimension provided that there is no evidence favoring higher dimensions.

To validate the clusters formed and get an insight into their composition we examine the results obtained in the period of Jan 2014 to Dec 2017[7]. To represent the clusters in a 2-D setting, the data must be reduced from 5 dimensions. We consider the application of t-SNE [29] for this purpose. Figure 8 illustrates the clusters formed. The ETFs not clustered are represented by the smaller circles, which were not labeled to facilitate the visualization.

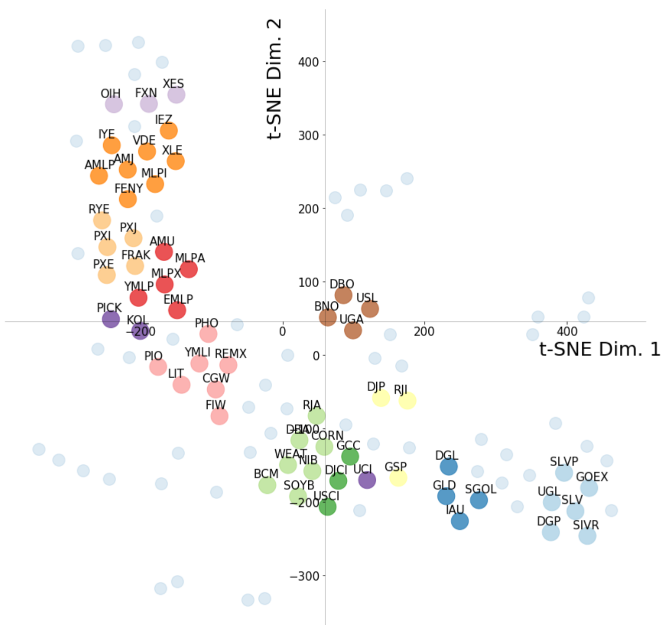
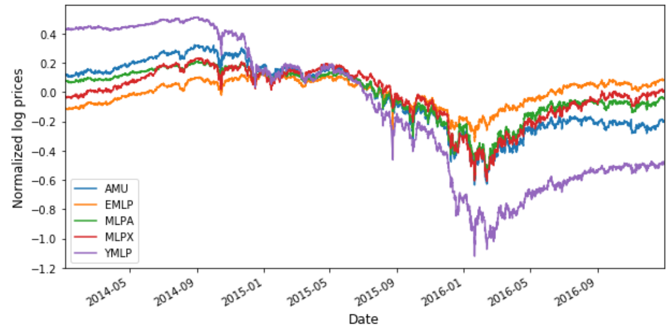


Fig. 8. Application of t-SNE to the clusters generated by OPTICS.

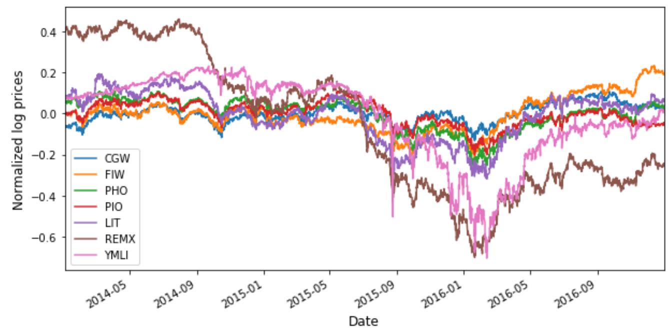
In order to evaluate the integrity of the clusters, we propose analyzing the composing price series. Therefore, we select two clusters and represent the logarithm of the price series[8] of each ETF. Figure 9(a) illustrates a cluster in which the ETFs identified do not just belong to the same category but are also part of the same segment, the Equity US: MLPs. This evinces that the OPTICS approach is capable of detecting a specific segment just from the time series data. Figure 9(b) demonstrates the OPTICS clustering capabilities extend beyond selecting ETFs within the same segment, as we may observe ETFs from distinct categories, such as Agriculture (CGW, FIW, PHO, and PIO), Industrial Metals (LIT and REMX) and Energy (YMLI). There is a visible relation among the identified price series, even though they do not all belong to the same category.

[7]This period is chosen arbitrarily because an extensive analysis covering all periods does not fit this report.

[8]The price series illustrated result from subtracting the mean of the original price series, to facilitate the visualization.



(a) Normalized prices in Cluster 1.



(b) Normalized prices in Cluster 2.

Fig. 9. Price series composition of some clusters.

We confirm the generated clusters display a tendency to group subsets of ETFs from the same category while not impeding clusters containing ETFs from different ones.

With respect to the trading performance, Table III unveils the test results obtained with each clustering type using the three different portfolios introduced in section V-A. To aggregate the information in a more concise way, the average over all years and portfolios is described on the rightmost column. Note also that the three evaluation metrics described in section V-E are accentuated, to differentiate from the remaining, less critical, descriptive statistics. We can confirm the profitability in all the environments tested, which corroborates the idea that the pairs selection rules are robust.

Comparing the different clustering techniques, if an investor is focused on obtaining the highest ROI, regardless of the incurred risk, performing no clustering is particularly appealing.

But when risk is taken into the equation, the OPTICS based strategy proves more auspicious. It is capable of generating the highest average portfolio Sharpe ratio of 3.79, in comparison with 3.58 obtained when performing no clustering or 2.59 when grouping by category. Also, it shows more consistency w.r.t the portion of profitable pairs in the portfolio, with an average of 86% profitable pairs, against 80% when grouping by category and 79% when performing no clustering at all. At last, it achieves more steady portfolio drawdowns, with the lowest average MDD. It is capable of mantaining the MDD values within an acceptable range even when the other two techniques display considerable deviations, as in 2017.

TABLE III
TRADING PERFORMANCE FOR EACH PAIRS SEARCH TECHNIQUE.

Test Period	**2015**			**2016**			**2017**			**AVG.**
Test Portfolio	1	2	3	1	2	3	1	2	3	-
No clustering										
SR	**3.53**	**4.12**	**3.32**	**3.96**	**4.51**	**3.56**	**4.08**	**4.05**	**1.11**	***3.58***
ROI	**10.4%**	**12.4%**	**17.4%**	**24.8%**	**26.3%**	**26.0%**	**11.9%**	**12.4%**	**11.5%**	***17.0%***
MDD	**1.42%**	**0.97%**	**2.59%**	**2.05%**	**1.98%**	**2.65%**	**1.33%**	**1.38%**	**9.28%**	***2.63%***
# pairs	101	77	10	247	223	10	150	141	10	*108*
% of profitable pairs	70%	80%	70%	86%	86%	90%	69%	70%	90%	*79%*
# total trades	229	173	17	411	361	15	212	195	14	*181*
# profitable trades	180	147	15	369	329	15	172	162	12	*156*
# unprofitable trades	49	26	2	42	32	0	40	33	2	*25*
Category										
SR	**1.56**	**2.39**	**3.75**	**3.48**	**3.82**	**3.09**	**2.17**	**2.14**	**0.89**	***2.59***
ROI	**5.52%**	**9.38%**	**17.8%**	**13.6%**	**13.9%**	**20.4%**	**7.86%**	**8.42%**	**8.31%**	***11.7%***
MDD	**1.77%**	**1.82%**	**2.09%**	**2.06%**	**2.26%**	**4.56%**	**2.47%**	**2.67%**	**8.91%**	***3.18%***
# pairs	59	40	10	51	44	10	51	47	10	*36*
% of profitable pairs	64%	85%	90%	86%	86%	90%	65%	64%	90%	*80%*
# total trades	154	108	39	107	83	20	64	54	13	*71*
# profitable trades	112	89	36	92	73	19	49	43	12	*58*
# unprofitable trades	42	19	3	15	10	1	15	11	1	*13*
OPTICS										
SR	**4.05**	**3.84**	**5.08**	**4.72**	**4.79**	**3.80**	**2.75**	**2.83**	**2.27**	***3.79***
ROI	**12.5%**	**13.5%**	**23.5%**	**10.5%**	**11.9%**	**15.2%**	**7.36%**	**8.38%**	**9.98%**	***12.5%***
MDD	**1.37%**	**1.66%**	**1.30%**	**0.80%**	**0.83%**	**1.46%**	**1.21%**	**1.35%**	**2.35%**	***1.37%***
# pairs	39	34	10	40	35	10	18	16	10	*24*
% of profitable pairs	82%	82%	100%	80%	83%	90%	78%	81%	100%	*86%*
# total trades	161	147	68	87	78	30	24	22	17	*70*
# profitable trades	140	128	67	72	66	27	21	20	17	*62*
# unprofitable trades	21	19	1	15	12	3	3	2	0	*8*

B. Evaluation of the forecasting-based model

We start by selecting pairs using the OPTICS clustering, due to its demonstrated ability. On these conditions, we find 5 pairs during the formation period of Jan 2009 to Dec 2017 and 19 pairs during Jan 2015 to Dec 2017 (periods defined in Figure 6). Not surprisingly, the number of pairs found for the former period is greatly reduced as the active cointegrated ETFs throughout this interval are more scarce. But since training the Deep Learning forecasting models is computationally very expensive, having fewer pairs is actually convenient. The corresponding five spreads are illustrated in Figure 10. The spreads look indeed stationary. There is an evident difference in their volatility, which further supports the importance of enforcing data-driven trading thresholds.

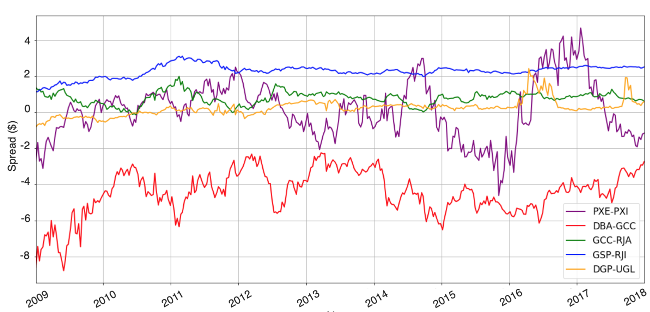


Fig. 10. Pairs identified in Jan 2009-Dec 2017.

Each spread in Figure 10 is fitted by the forecasting algorithms. The forecasting score is obtained by averaging the mean-square error (MSE) over the five spreads.

A total of 31 forecasting model architectures are implemented in this work to find the one with the most trading potential, meaning 155 models are trained (31 architectures $\times$ 5 spreads). We experiment increasingly complex configurations until signs of overfitting are evident. The forecasting performance obtained for the best configurations is described in Table IV.

TABLE IV
FORECASTING RESULTS COMPARISON.

Model	**Naïve**		**ARMA**	**LSTM**	**LSTM Encoder-Decoder**	
Configuration	- - -		*p* 8 *q* 3 -	*# inputs* 24 *# hidden layers* 1 *# hidden nodes* 50	*# inputs* 24 *# encoder nodes* 15 *# decoder nodes* 15	
Time Step	(t+1)	(t+2)	(t+1)	(t+1)	(t+1)	(t+2)
Parameters	-		11	10 651	3 016	
Validation						
MSE (E-03)	1.87	3.34	1.51	1.69	1.71	2.05
RMSE (E-02)	3.69	4.94	3.00	3.28	3.32	3.60
MAE (E-02)	1.50	2.24	1.78	2.04	2.13	2.45
Test						
MSE (E-03)	2.60	4.47	2.26	3.35	4.31	9.03
RMSE (E-02)	3.89	5.14	3.34	4.30	5.21	7.50
MAE (E-02)	1.68	2.61	1.96	3.08	3.94	5.91

We may verify that all the implemented models are capable of outperforming the naive implementation during the validation period. Curiously, we note that the LSTM-based models do not manage to surpass the ARMA model, at least w.r.t to the chosen metrics. Also, the results obtained in the test set indicate signs of overfitting besides the efforts taken in that regard, as the LSTM-based models are no longer superior to the naive performance. The incapability of the LSTM-based models to outperform the simpler approaches is in accordance with the findings of [30], which asserts that time-series problems found in the literature are often conceptually simpler than many tasks already solved by LSTMs and more often than not they all relevant information about the next event is conveyed by a few recent events. We suspect this is the case in this work. At last, we analyze the performance obtained by the integration of the previous algorithms in the proposed trading model scheme. Based on the validation records, the quintile-based thresholds are used with ARMA and the decile-based with the LSTMs. The test results in these conditions are illustrated in Table V.

TABLE V
TRADING RESULTS COMPARISON USING A 8-YEAR-LONG FORMATION PERIOD.

Trading Model	**Standard**	**ARMA based Model**	**LSTM based Model**	**LSTM Encoder Decoder based Model**
SR	1.85	1.22	0.50	0.98
ROI	6.27%	5.57%	2.93%	4.17%
MDD	1.43%	0.73%	0.47%	1.19%
# days of portfolio decline	**87**	**11**	**2**	**21**
# trades (positive-negative)	149 (89-60)	34 (22-12)	8 (6-2)	17 (14-3)
# profitable pairs	3	3	2	2

The results indicate that if robustness is evaluated by the number of days the portfolio value does not decline (accentuated in Table V), then the proposed trading model does provide an improvement. The forecasting-based models display a total of 2 (LSTM), 11 (ARMA) and 22 (LSTM Encoder-Decoder) days of portfolio decline, in comparison with 87 days obtained when using the standard model. This finding suggests the forecasting-based model is capable of defining more precise entry points, and hence reduce the number of unprofitable days. However, that comes at the expense of a reduction in both portfolio SR and ROI, questioning the benefits provided by the proposed model after all. We suspect the long required formation period is also responsible for this profitability decline. Therefore we proceed to analyze the standard trading model in the 3-year-long period.

TABLE VI
TRADING RESULTS FOR STANDARD TRADING MODEL USING A 3-YEAR-LONG FORMATION PERIOD.

Trading Model	Standard
SR	**3.41**
ROI	**11.3%**
MDD	1.12%
# days of portfolio decline	89
# trades (positive-negative)	30 (26-4)
# profitable pairs	13

By comparison, the performance in the 10-year-long period seems greatly affected by the long required duration, suggesting the less satisfactory returns emerge not simply from the trading model itself, but also due to the underlying time settings. Following this line of reasoning, if the forecasting-based models' performance increases in the same proportion as the standard trading model when reducing the formation period, the results obtained could be much more satisfactory.

VII. CONCLUSIONS

We explored how Pairs Trading could be enhanced with the integration of Machine Learning. First, we proposed a new approach to search for pairs based on the application of the OPTICS algorithm followed by a robust pairs selection criteria. The strategy achieved better risk-adjusted returns when using this method. Secondly, we introduced a forecasting-based model aiming to reduce decline periods associated with untimely market positions and prolonged divergent pairs. We demonstrated the proposed model is capable of reducing the average decline period in more than 75% although that comes at the expense of declining profitability. In addition, this work also contributes with empirical evidence of the suitability of ETFs traded in a 5-minutes setting in the context of Pairs Trading.

REFERENCES

[1] C. Krauss, X. A. Do, and N. Huck, "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500," *European Journal of Operational Research*, vol. 259, no. 2, pp. 689–702, 2017.

[2] J. Caldeira and G. V. Moura, "Selection of a portfolio of pairs based on cointegration: A statistical arbitrage strategy," *Available at SSRN 2196391*, 2013.

[3] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst, "Pairs trading: Performance of a relative-value arbitrage rule," *The Review of Financial Studies*, vol. 19, no. 3, pp. 797–827, 2006.

[4] B. Do and R. Faff, "Does simple pairs trading still work?" *Financial Analysts Journal*, vol. 66, no. 4, pp. 83–95, 2010. [Online]. Available: https://doi.org/10.2469/faj.v66.n4.1

[5] C. L. Dunis, G. Giorgioni, J. Laws, and J. Rudy, "Statistical arbitrage and high-frequency data with an application to eurostoxx 50 equities," *Liverpool Business School, Working paper*, 2010.

[6] C. Krauss, "Statistical arbitrage pairs trading strategies: Review and outlook," *Journal of Economic Surveys*, vol. 31, no. 2, pp. 513–545, 2017.

[7] H. Chen, S. Chen, Z. Chen, and F. Li, "Empirical investigation of an equity pairs trading strategy," *Management Science*, 2017.

[8] G. Vidyamurthy, *Pairs Trading: quantitative methods and analysis*. John Wiley & Sons, 2004, vol. 217.

[9] N. Huck and K. Afawubo, "Pairs trading and selection methods: is cointegration superior?" *Applied Economics*, vol. 47, no. 6, pp. 599–613, 2015.

[10] C. L. Dunis, J. Laws, and B. Evans, "Modelling and trading the gasoline crack spread: A non-linear story," *Derivatives Use, Trading & Regulation*, vol. 12, no. 1-2, pp. 126–145, 2006.

[11] C. L. Dunis, J. Laws, P. W. Middleton, and A. Karathanasopoulos, "Trading and hedging the corn/ethanol crush spread using time-varying leverage and nonlinear models," *The European Journal of Finance*, vol. 21, no. 4, pp. 352–375, 2015.

[12] N. S. Thomaidis, N. Kondakis, and G. Dounias, "An intelligent statistical arbitrage trading system," in *SETN*, 2006.

[13] N. Huck, "Pairs selection and outranking: An application to the s&p 100 index," *European Journal of Operational Research*, vol. 196, no. 2, pp. 819–825, 2009.

[14] N. Huck, "Pairs trading and outranking: The multi-step-ahead forecasting case," *European Journal of Operational Research*, vol. 207, no. 3, pp. 1702–1716, 2010.

[15] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*. Springer, 2006, pp. 25–71.

[16] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise."

[17] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.

[18] T. Kleinow, "Testing continuous time models in financial markets," 2002.

[19] E. Chan, *Algorithmic trading: winning strategies and their rationale*. John Wiley & Sons, 2013, vol. 625.

[20] Y.-W. Si and J. Yin, "Obst-based segmentation approach to financial time series," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2581–2596, 2013.

[21] R. C. Cavalcante, R. C. Brasileiro, V. L. Souza, J. P. Nobrega, and A. L. Oliveira, "Computational intelligence and financial markets: A survey and future directions," *Expert Systems with Applications*, vol. 55, pp. 194–211, 2016.

[22] M. Perlin, "M of a kind: A multivariate approach at pairs trading," 2007.

[23] "Find the Right ETF - Tools, Ratings, News," https://www.etf.com/, accessed: 2019-06-30.

[24] H. Rad, R. K. Y. Low, and R. Faff, "The profitability of pairs trading strategies: distance, cointegration and copula methods," *Quantitative Finance*, vol. 16, no. 10, pp. 1541–1558, 2016. [Online]. Available: https://doi.org/10.1080/14697688.2016.1164337

[25] B. Do and R. Faff, "Are pairs trading profits robust to trading costs?" *Journal of Financial Research*, vol. 35, no. 2, pp. 261–287, 2012.

[26] "3-Month Treasury Bill: Secondary Market Rate," https://fred.stlouisfed.org/series/TB3MS, accessed: 2019-07-11.

[27] A. W. Lo, "The statistics of sharpe ratios," *Financial analysts journal*, vol. 58, no. 4, pp. 36–52, 2002.

[28] S. Moraes Sarmento, "Github repository: Pairs trading," https://github.com/simaomsarmento/PairsTrading, 2019.

[29] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[30] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying lstm to time series predictable through time-window approaches," in *Neural Nets WIRN Vietri-01*. Springer, 2002, pp. 193–200.