Predicting cointegration using neural network and fundamental ratios in statistical arbitrage

Chan, Yun Kit Chan

yunkitchan004@gmail.com

Supervised by: Dr. David Rossiter

COMP 4971F – Independent Work

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

Spring 2023

Abstract

Since 1980, pair trading has become one of the most widely used methods for identifying statistical arbitrage opportunities in the stock market. Different methods have been proposed over time to take advantage of the strategy. There is, however, the conventional approach that does not gauge future cointegration relationships, and limited literature examines this flaw that may lead to bad trades. This work examines the performance of pairs trading strategies using neural network techniques applied to the oil and gas industry and detecting the effects of the fundamental ratios in pairs before making a trade decision. We will demonstrate the merits and effectiveness of the proposed approach by conducting an experiment on 82 stocks selected from the industry.

Table of Contents

1. Introduction

1.1 Background

Statistical arbitrage techniques became a well-liked strategy on Wall Street after Gatve et al. (1999) demonstrated that the methodology could produce a consistent positive return for 30 years. (1962-1997). Pairs trading is a quantitative trading strategy that forms a portfolio of two related assets and exploits inefficient financial markets in order to take advantage of possible deviations from the "equilibrium" state to profit. Based on the assumption that the pair has an equilibrium relationship and the price spread is likely to be converged, we can profit by going long on the comparatively undervalued asset, short on the comparatively overvalued asset, and unwind the position when converging.

Apart from applying in the US stock market, statistical arbitrage is also a practical methodology in other markets and assets class. Dai et al. (2011) explore the arbitrage opportunities between futures and their spot. Kanamura et al. (2009) observed the spreads between commodity ETFs and physical commodities; Breyer et al. (2016) analyzed the self-financing strategy in Brazilian and European, and achieved a positive return in different markets.

1.2 History of pair trading

Gerry Bamberger and Nunzio Tartaglia first introduced pair trading at Morgan Stanley in the 1980s, trading for two highly-correlated assets. With the growing availability of data, traders have proposed numerous approaches to make profit more stable to measure the relationship. Krauss et al. (2020) have summarized different approaches, including cointegration, distance, stochastic spread, and residual approaches. The approaches mainly differ in how pairs are selected during the formation period.

The cointegration approach is a mathematics method to explore the possibility of cointegration invented by Vidyamurthy (2004). The methodology has gained the most attention among the other techniques because of its superior profitability. The distance approach selects pairs that minimize the sum of squared differences between the two normalized assets' price series. For the stochastic approach, Elliott et al. (2005) model the mean reversion behavior of the spread of two assets. Tim Leung (2020) utilized Ornstein-Uhlenbeck (OU) process through maximum likelihood estimation to optimize the exit rule and intraday portfolio value.

Finding the appropriate pairs of assets is only the first challenge in pair trading. Constructing the open entry and exit rule becomes the following key to profit. The conventional approach is to open entry or exit when the spread reaches some pre-determined threshold. When the entry threshold is reached, the outperforming stock is sold short, and the underperforming stock is purchased. Then we exit the trades when the spread converges back to its mean. Alongside the rising machine learning adoption, sophisticated artificial intelligence and neural network comes into play in this process. Researchers validated the applicability of LSTM and RNN for identifying pair-trading opportunities and spread prediction (Flori et al., 2020; Karsi, 2019). Diego et al. (2019) recognized pairs using PCA and DBSCAN clustering methods. Krauss (2016) implemented deep neural networks (DNN), gradient-boosted-trees (GBT), random forests (RAF), and ensemble learning to predict the individual stock prices, then buy the top k stocks, and sell short the bottom k stocks in the rank.

1.3 Problem and hypotheses

Although the existing method to select asset pair seems efficient, the system has one crucial flaw when examined thoroughly. No matter which selecting approach is adopted, the price deviation and the cointegration relationship are the lagged observation of two companies' trading activities, earnings potential, and future viability. A stock pair's historical correlation barely indicates the future relationship between two assets. To put it another way, when traders realize that two assets lose the cointegration relationship, the strategy will lead to long decline periods because of the prolonged divergent pair.

In order to solve the problem above, this paper contributes to the literature by proposing a methodology to predict the cointegration relationship using the spread of financial statement variables of two companies in a pair in the oil and gas industry enhanced by the artificial neural network. The rationale for investigating one specific industry will be explained in the coming part of the research. The comparative fundamental analysis of two assets predicts two firms' intrinsic value change and reveals the earnings potential. The model will filter the selected pairs and eliminate the potential prolonged divergent pair.

1.4 Paper structure

The rest of this paper is organized as follows. The technical background and related work are stated in Section 2, and the materials and data are discussed in Section 3. The methodology is described in Section 4, and Section 5 concludes the back-testing results. Section 6 provides our conclusions to this study.

2. Technical Background

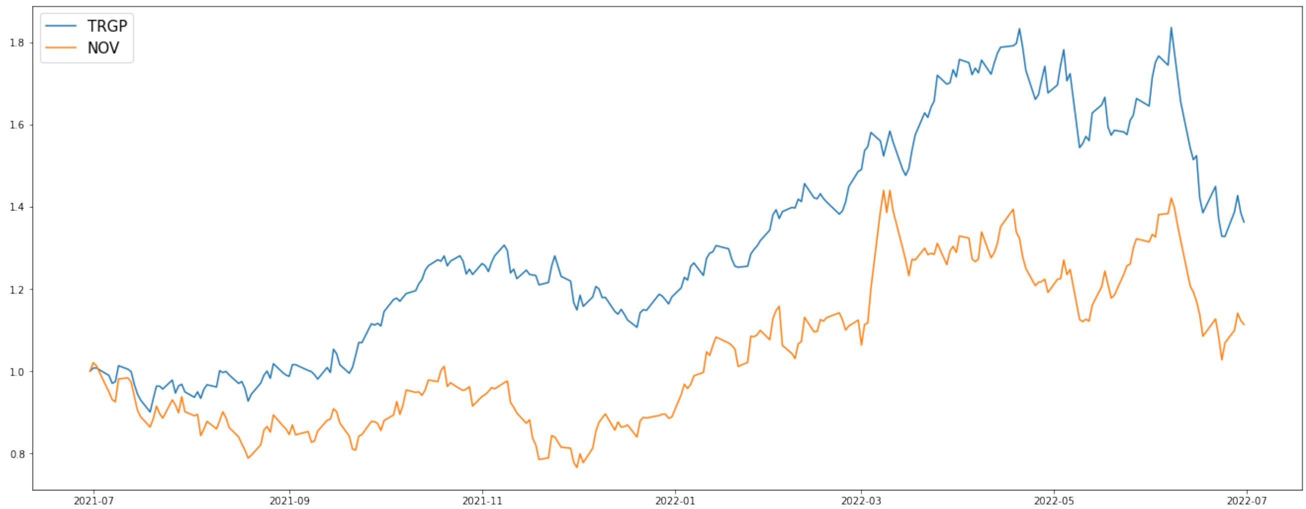
2.1 Cointegration

Cointegration describes the long-run relationship between two non-stationary but integrated at the same order time series, so we expect that any deviations from this long-run relationship are non-stationary (Bui et al., 2021). If two assets, Y_t and X_t, are cointegrated, then the series constructed as

$$S_t = Y_t - \beta X_t,$$

Where β is the cointegration factor, it must be stationary (Sarmento et al., 2020).

The results of cointegration tests reveal situations in which two non-stationary time series are integrated in such a way that they are unlikely to deviate from equilibrium over the long term (CFI Team, 2022). To put it differently, two assets' prices will drift in such a way that they do not drift away from each other. This concept is essential and efficient in this paper. The spread is expected to be mean-reverting under the circumstances, which indicates that every spread divergence is likely to be followed by convergence. On top of that, this approach is highly preferable because cointegration-based pairs trading outperforms the distance approach in terms of profitability, alpha stability, and non-convergence risk (Huck et al., 2015). Graph 1 below demonstrates the one-year return of two companies in a cointegrated pair (1% significance) - Targa Resources (TRGP) and NOV Inc (NOV) from 21Q3 to 22Q2. One of the potential reasons is trend of Book Value, Earning per share and Free Cash Flow is similar.



Graph 1: Return of Targa Resources (TRGP) and NOV Inc (NOV) from 21Q3 to 22Q2

2.2 Artificial Neural Network

2.2.1 Origin

Artificial Neural Network (ANN), introduced by McCulloch and Pitts in 1943, is a modeling technique inspired by biological neural networks in the human brain. The model efficiently uses a computational connectivity approach and learns from observations of a group of phenomena. It is made up of a group of interconnected units exchanging information with each other in order to produce an output. (Yildiz, 2010)

2.2.2 Structure of ANN

ANN consists of three primary node layers: input layer, hidden layer(s), and output layer. Each node connects to the other and has an associated weight and threshold. Any node whose output is above the defined threshold value is activated and sends data to the next layer of the network (IBM Cloud Education, 2020). Otherwise, no data is transmitted to the next layer in the network. The neural network passing data from one layer to the next is called feedforward network. Backpropagation is also a common algorithm to train feedforward neural networks. Unlike feedforward network, backpropagation moves in the opposite direction from output to input.

By activating functions, input layer nodes transmit information to hidden nodes, which either activate or do nothing. Any node whose output exceeds the defined threshold value is activated and provides data to the network's uppermost layer. Otherwise, no data is transmitted to the network's next tier. The visual below shows the process of data transmission. Neural networks rely on training data and need to be trained with a large number of data due to the complex structure, so ANN cannot deal with rare cases with insufficient data size.

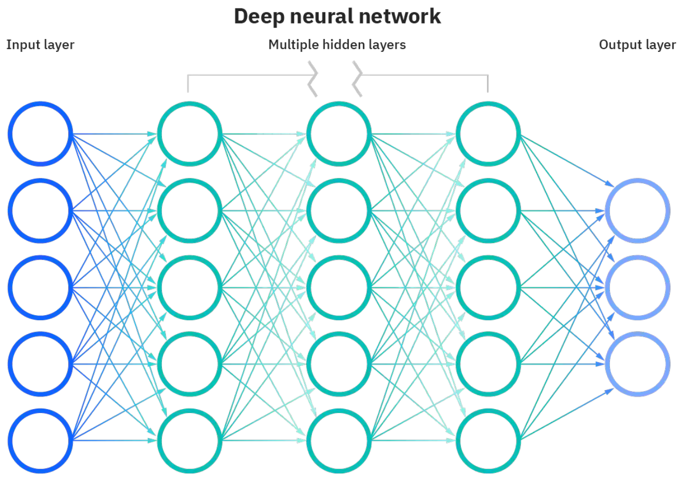
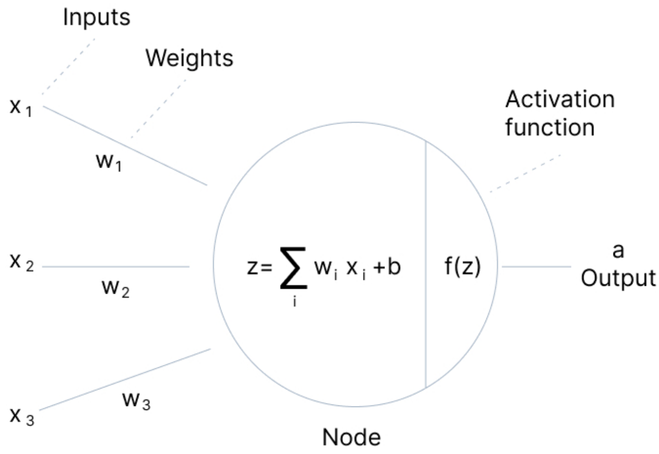


Image source: IBM

2.2.3 Learning Method

The summation function and transfer function are the two fundamental functions of each neuron's information processing system. The summation function receives the weighted sum of all inputs that reach the neuron, and we can think of each node as its linear regression model. The transfer function determines the activation level of the neuron and the relationship between stimulation level and output. One purpose of the transfer function is to limit the output, add non-linearity and decide whether a neuron should be activated. Common non-linear neural network activation functions include the Logistic activation function (Sigmoid), Hyperbolic Tangent (Tanh), Rectified Linear Unit (ReLU), and Softmax (Baheti, 2022). The final goal of ANN is to minimize the cost function by learning from the errors to adjust the weights and biases. By adjusting the weights and bias through gradient descent, the model would determine the direction to take to reduce errors and eventually converge at the local minimum. Graph 2 visualizes how data is transmitted and processed into output in a neuron.



Graph 2: Visualization of neural network learning in one node.

Source: https://www.v7labs.com/blog/neural-networks-activation-functions#h1

3. Materials and Method

3.1 Financial data to use

All price and fundamental data were obtained from Financial Modeling Prep. Financial Modeling Prep[1] delivers business and markets news and data, and it became the data source of this research. The dataset contains 136 stock companies (82 after stock screening) listed in the oil and gas industry from June 2012 to July 2022. For the stock prices, data frequency is daily, with 3,280 fundamental data observations for each stock and 231,486 price data in total in the whole dataset. Regarding fundamental financial data, Book Value (BV), Free Cashflow (CFC), and Earning Per Share (EPS) will be collected quarterly. As a result, 40 fundamental data observations (10 years * 4 quarters) and 2,823 price observations per stock will be collected.

Prefiltering the stock before forming pairs is an essential step in pair trading. Stocks that do not satisfy the following condition will be dropped:

- Stock from companies with missing data
- Stock from companies with a market cap lower than $1B
- Stock from companies with an average daily volume lower than $150k

After the stock screening, 82 out of 136 stocks are selected for pair trading. Stock screening is applied to ensure liquidity, investor scrutiny of the company, and data quality for model training.

3.2 Data preparation

3.2.1 Cointegration with Dickey-Fuller test

Before training the model, data exploration is crucial to producing meaningful information and organizing for the following data processing stage.

Firstly, the price data of two stocks in one pair will be processed by the augmented Dickey-Fuller test (ADF) to generate a 'cointegration constant.' The augmented Dickey-Fuller test to validate the stationarity of a time series sample. Take BP (BP) and Cenovus Energy (CVE) as an example, and we conduct an ordinary least squares (OLS) regression on the one-year stock prices of the two companies. The residuals of OLS will be processed by ADF, and the pair will be classified as "cointegrated" if the p-value of the test is less than 0.05; thus, the "cointegration constant" will be 1. Otherwise, the “cointegration constant” will be 0. The "cointegration constant" is the dependent variable in the model, so the time series of stock prices used will be three quarters before the end date of this quarter plus the next quarter. Table 1 shows an example of the data exploration process to generate an observation.

[1] https://site.financialmodelingprep.com/

3.2.2 Financial accounting data

In this research, three fundamental data - BV, CFC, and EPS will be collected to evaluate the performance of each company. Quirin et al. (2000) researched the survey instrument administered to financial analysts specializing in the oil and gas industry and identified nine fundamental signals instrumental in the equity valuation process. Three of the top five signals are adopted in this research. The remaining signals are "Margin per Barrel" and "Reserve Replacement Efficiency," which will be excluded due to the scarcity of data.

The fundamental data will be normalized and computed as ratio percentage change from the quarter for the four time lags. For each observation, we will calculate the change in the fundamental data ratio of two stocks to generate the performance difference between the two companies. Table 1 demonstrates a data exploration process of BP and CVE in 15Q1; stock prices from 01/07/2014 to 30/06/2015 will be computed into "cointegration constant." This method enables the model's forecasting power to predict the cointegration relationship in the next quarter.

Table 1: Data sample of pair BP & CVE in 22Q1 (Computed on 31 March 2022)	
BV spread (t-1)	$\frac{\left[\left(\frac{\text{BP } Normalized \text{ BV}, 21Q4}{\text{CVE } Normalized \text{ BV}, 21Q4}\right)-\left(\frac{\text{BP } Normalized \text{ BV}, 21Q3}{\text{CVE } Normalized \text{ BV}, 21Q3}\right)\right]}{\frac{\text{BP } Normalized \text{ BV}, 21Q3}{\text{CVE } Normalized \text{ BV}, 21Q3}}$
BV spread (t-2)	$\frac{\left[\left(\frac{\text{BP } Normalized \text{ BV}, 21Q3}{\text{CVE } Normalized \text{ BV}, 21Q3}\right)-\left(\frac{\text{BP } Normalized \text{ BV}, 21Q2}{\text{CVE } Normalized \text{ BV}, 21Q2}\right)\right]}{\frac{\text{BP } Normalized \text{ BV}, 21Q2}{\text{CVE } Normalized \text{ BV}, 21Q2}}$
BV spread (t-3)	$\frac{\left[\left(\frac{\text{BP } Normalized \text{ BV}, 21Q2}{\text{CVE } Normalized \text{ BV}, 21Q2}\right)-\left(\frac{\text{BP } Normalized \text{ BV}, 21Q1}{\text{CVE } Normalized \text{ BV}, 21Q1}\right)\right]}{\frac{\text{BP } Normalized \text{ BV}, 21Q1}{\text{CVE } Normalized \text{ BV}, 21Q1}}$
BV spread (t-4)	$\frac{\left[\left(\frac{\text{BP } Normalized \text{ BV}, 21Q1}{\text{CVE } Normalized \text{ BV}, 21Q1}\right)-\left(\frac{\text{BP } Normalized \text{ BV}, 20Q4}{\text{CVE } Normalized \text{ BV}, 20Q4}\right)\right]}{\frac{\text{BP } Normalized \text{ BV}, 20Q4}{\text{CVE } Normalized \text{ BV}, 20Q4}}$
FCF spread (t-1)	$\frac{\left[\left(\frac{\text{BP } Normalized \text{ FCF}, 21Q4}{\text{CVE } Normalized \text{ FCF}, 21Q4}\right)-\left(\frac{\text{BP } Normalized \text{ FCF}, 21Q3}{\text{CVE } Normalized \text{ FCF}, 21Q3}\right)\right]}{\frac{\text{BP } Normalized \text{ FCF}, 21Q3}{\text{CVE } Normalized\ FCF, 21Q3}}$
FCF spread (t-2)	$\frac{\left[\left(\frac{\text{BP } Normalized \text{ FCF}, 21Q3}{\text{CVE } Normalized \text{ FCF}, 21Q3}\right)-\left(\frac{\text{BP } Normalized \text{ FCF}, 21Q2}{\text{CVE } Normalized \text{ FCF}, 21Q2}\right)\right]}{\frac{\text{BP } Normalized\ FCF, 21Q2}{\text{CVE } Normalized \text{ FCF}, 21Q2}}$

FCF spread (t-3)	$\dfrac{\left[\left(\dfrac{\text{BP } Normalized \text{ FCF}, 21Q2}{\text{CVE } Normalized \text{ FCF}, 21Q2}\right) - \left(\dfrac{\text{BP } Normalized \text{ FCF}, 21Q1}{\text{CVE } Normalized \text{ FCF}, 21Q1}\right)\right]}{\dfrac{\text{BP } Normalized \text{ FCF}, 21Q1}{\text{CVE } Normalized \text{ FCF}, 21Q1}}$
FCF spread (t-4)	$\dfrac{\left[\left(\dfrac{\text{BP } Normalized \text{ FCF}, 21Q1}{\text{CVE } Normalized \text{ FCF}, 21Q1}\right) - \left(\dfrac{\text{BP } Normalized \text{ FCF}, 21Q4}{\text{CVE } Normalized \text{ FCF}, 21Q4}\right)\right]}{\dfrac{\text{BP } Normalized \text{ FCF}, 20Q4}{\text{CVE } Normalized \text{ FCF}, 20Q4}}$
EPS spread (t-1)	$\dfrac{\left[\left(\dfrac{\text{BP } Normalized \text{ EPS}, 21Q4}{\text{CVE } Normalized \text{ EPS}, 21Q4}\right) - \left(\dfrac{\text{BP } Normalized \text{ EPS}, 21Q3}{\text{CVE } Normalized \text{ EPS}, 21Q3}\right)\right]}{\dfrac{\text{BP } Normalized \text{ EPS}, 21Q3}{\text{CVE } Normalized \text{ EPS}, 21Q3}}$
EPS spread (t-2)	$\dfrac{\left[\left(\dfrac{\text{BP } Normalized \text{ EPS}, 21Q3}{\text{CVE } Normalized \text{ EPS}, 21Q3}\right) - \left(\dfrac{\text{BP } Normalized \text{ EPS}, 21Q2}{\text{CVE } Normalized \text{ EPS}, 21Q2}\right)\right]}{\dfrac{\text{BP } Normalized \text{ EPS}, 21Q2}{\text{CVE } Normalized \text{ EPS}, 21Q2}}$
EPS spread (t-3)	$\dfrac{\left[\left(\dfrac{\text{BP } Normalized \text{ EPS}, 21Q2}{\text{CVE } Normalized \text{ EPS}, 21Q2}\right) - \left(\dfrac{\text{BP } Normalized \text{ EPS}, 21Q1}{\text{CVE } Normalized \text{ EPS}, 21Q1}\right)\right]}{\dfrac{\text{BP } Normalized \text{ EPS}, 21Q1}{\text{CVE } Normalized \text{ EPS}, 21Q1}}$
EPS spread (t-4)	$\dfrac{\left[\left(\dfrac{\text{BP } Normalized \text{ EPS}, 20Q1}{\text{CVE } Normalized \text{ EPS}, 20Q1}\right) - \left(\dfrac{\text{BP } Normalized \text{ EPS}, 20Q4}{\text{CVE } Normalized \text{ EPS}, 20Q4}\right)\right]}{\dfrac{\text{BP } Normalized \text{ EPS}, 20Q4}{\text{CVE } Normalized \text{ EPS}, 20Q4}}$
Cointegration constant	ADF(BP stock price from 21Q3-22Q2, CVE stock price from 21Q3-22Q2)

For cointegration constant, the stock price is ranged from 21Q3 to 22Q2. The model is trained to predict whether the pair is cointegrated after a quarter.

In each quarter, all stocks will form pairs with all other stocks, and there will be 6,642 (82 * 81) data. As a result, we would have 265,680 observations in 40 quarters.

4. Methodology

4.1 Introduction

The following section will be described the pair selection methodology and the ANN used for cointegration relationship prediction. On top of that, we will propose a trading strategy using the result above, and details include trade criteria to open entry and exit a pair trade. Also, the transaction costs adopted in backtesting will be described.

Although fundamental analysis is less researched than technical analysis, the fundamental signal has been proven to have a strong predictive power of return. It has become an irreplaceable part of traders. Fundamental analysis focuses on the underlying aspect of stocks, industrial prospects, and macro-economy, which can be summarized as the performance of individual companies, industries, and markets. As an arbitrage strategy, pair trading is capable of hedging the industrial and market risk of two stocks. Therefore, if the fundamental signal of two companies with cointegrated stock prices diverges to a certain level, the signal may indicate that the two companies have different abilities to generate revenue and growth, eventually leading to different stock price performances. The divergence of the financial accounting figures is an important signal to foresee the loss of the cointegration relationship and a prolonged divergent pair.

4.2 Industry selection

After building the concept of the model, the next question is how stocks are picked to form pairs. Unlike previous research that picks stocks in S&P500 or NASDAQ 100 (Jskobsson, 2015), only one specific industry is picked in this research. The reason is that stocks in the same sub-sector perform the best portfolio. The second reason is that no model is suitable for capturing the stock trends of all industries but only analyzing one industry will train an optimal model (Lv et al., 2019). Moreover, this approach is less likely to find spurious relations that may break in an unpredictable way (Sarmento et al., 2020).

This paper focuses on trading in the oil and gas exploration and production industry. The industry is a capital-intensive, heavy-industry, and value stock-oriented subsector. Value stocks present an opportunity to buy shares close to intrinsic value, while growth stocks like SaaS and semiconductor companies are expected to grow significantly faster. As a result, traders focus more on fundamentals that gauge company performance, like production efficiencies, than on indicators of marketing, sales, or product price traits when trading in the oil and gas industry. The difference implies that growth stock is relatively high volatility in the industry and may lose cointegration relationships more quickly and frequently.

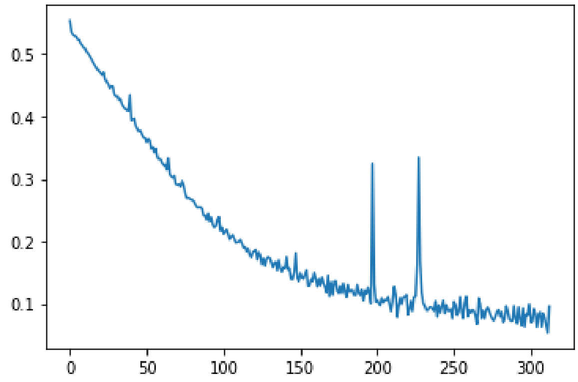
4.3 Neural Network design

ANN is an advanced pattern recognition algorithm that is able to deal with non-linearity among variables. As mentioned in section 1.3, ANN will be applied to solve the classification problem.

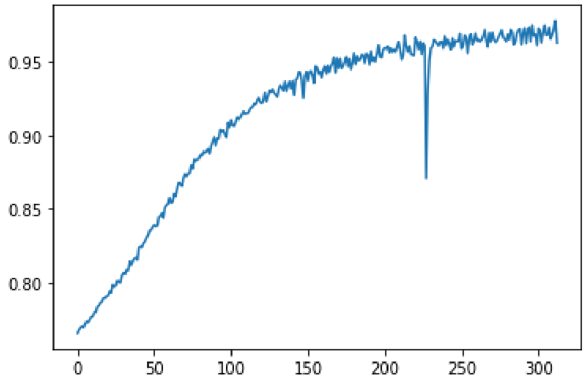
Some suggest using other neural networks, such as long short-term memory (LSTM) and gated recurrent units (GRUs) with backpropagation, that train time series data with better performance (Lopez et al., 2020). Despite the fact that the Recurrent neural network (RNN) is one of the most advanced models for training time-series data, they are not applicable to this research. As the model is required to deal with multivariable for multiple companies, RNN is only capable of analyzing multivariable for one company. The remedy for RNN is to train one model for one company and then ensemble the model, but that approach may lead to overfitting and the curse of dimensionality. As a result, an unusual approach is adopted in this paper that leverages ANN for analyzing multivariable for multiple companies.

Designing the architecture of ANN is essential in building the optimal model and avoiding underfit and overfit. This research applied the testing subset to gauge the model performance and tune the architecture and parameters accordingly. After computationally intensive optimization, the model reached its local optimal. The architecture and parameters are described as follows:

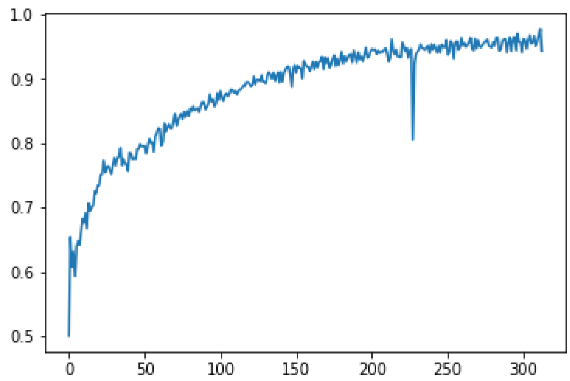
- Network layers and neurons: The proposed network has four layers, including one input layer, two hidden layers, and one output layer. As ANN is a feedforward network, the input layer has 16 neurons that match the number of input variables. The hidden layers have 80 and 40 neurons each, and the output layer has one neuron because the output is the cointegration constant prediction for the coming quarter.
- Activation functions: ReLU is adopted in the input and hidden layers due to the sparsity attribute and is widely used in practice. The output layer has a sigmoid function to normalize the output and generate values between 0 and 1, which is the range of probability. The "1" signal is given if the output is greater than 0.5. The model solves a classification problem rather than a regression problem, as previous research suggested that the classification model has better performance than the latter one when it comes to financial market data (Leung et al., 2000).
- Other: Adam optimizer is adopted in training. The network stops training when there is no further improvement in the binary cross-entropy for 20 consecutive epochs to balance overfitting and the generalization ability. The model eventually runs for 313 epochs, and the performance is exhibited in graph 3-6



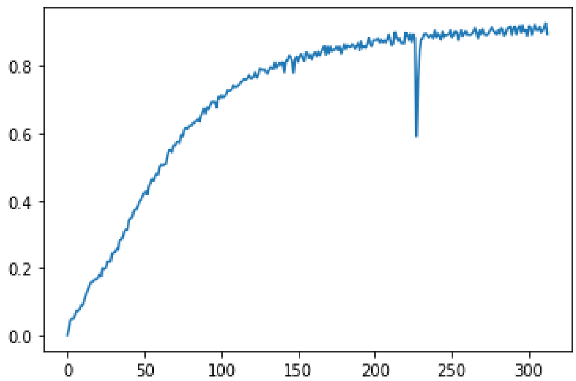
Graph 3: MSE (Loss function)



Graph 4: Recall



Graph 5: Precision



Graph 6: Accuracy

The study period included 40 quarters, and the first three quarters were dropped to generate features. Under the condition, we split the training-testing set as 80:20 in each quarter, resulting in 15,138 observations in the training set and 3,784 observations in the test set.

All the coding work is built in Python. Some of the well-developed packages foster development and are very helpful for quantitative trading research. Firstly, the sklearn package helps in the data preparation process, including normalization and data clearing. Regarding data scraping and data collection, requests, yahoo_fin, and finnhub are used. Last but not least, we adopt TensorFlow to construct the neural network and produce the final output.

4.4 Trading logic

The new alternative will be compared with the conventional approach to determine which logic performs better and to prove whether the neural network improves the logic.

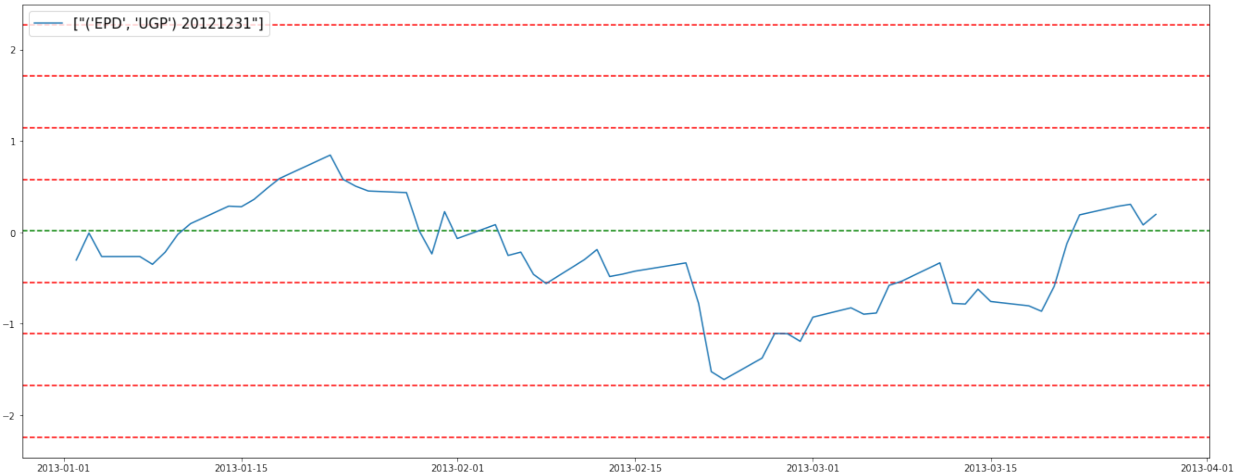
4.4.1 Threshold

In previous literature, three widely applied threshold design in pair trading is proposed - the fixed volatility threshold (Gatev et al., 1999), the conditional volatility threshold (Ferretti et al, 2017), and the breakout volatility threshold(Bui, 2021). The fixed volatility threshold is computed by the standard deviation of the whole period, the conditional volatility threshold is computed by rolling standard deviation, and the breakout volatility threshold is constructed by rolling standard deviation and momentum. In the paper, the fixed volatility threshold is adopted.

This threshold design computes constant volatility to define the threshold. The fixed threshold is defined as follows:

Upper fix threshold 4 = $\mu + 2\sigma$
Upper fix threshold 3 = $\mu + 1.5\sigma$
Upper fix threshold 2 = $\mu + \sigma$
Upper fix threshold 1 = $\mu + 0.5\sigma$
Lower fix threshold 1 = $\mu - 0.5\sigma$
Lower fix threshold 2 = $\mu - \sigma$
Lower fix threshold 3 = $\mu - 1.5\sigma$
Lower fix threshold 4 = $\mu - 2\sigma$

Where μ represents the mean and σ represents the standard deviation of the 1-year spread. Graph 7 demonstrates the fixed volatility threshold of the pair Enterprise Products Partners LP (EPD) & Ultrapar Participacoes (UGP) in 13Q1



Graph 7: Fixed threshold for pair EPD & UGP: The red line represent the spread (0.5 sd separate from each other). The green line represents the mean, and the blue line represents the 3-months spread.

4.4.2 Entry condition

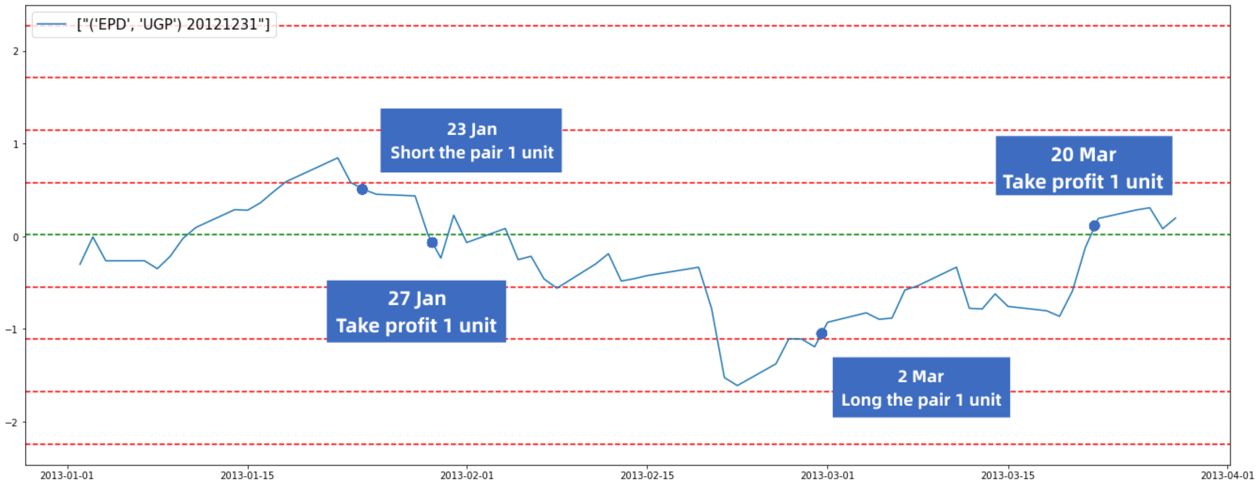
The conventional approach proposes entry when the spread touches the threshold (Kim et al., 2019). In the optimized logic, we will not open an entry immediately when the spread touches the upper (lower) threshold because the trend may keep the momentum and remain bullish (bearish), eventually becoming a divergent pair. In the new approach, we will open an entry when the spread breaks the upper (lower) threshold from the top down. On top of that, as a more significant spread divergence may lead to a stronger mean-reverting process, we will allocate higher trade capital if the spread breaks a more divergent threshold. Simply put, we will enter 1 unit of capital if the spread breaks upper (lower) threshold 1 and 3 units for upper (lower) threshold 3.

4.4.3 Exit condition

When a position is executed in the condition above, there are three scenarios to close the position, described as follows:

- Take-profit: The position will be closed when the spread returns to its mean, which is the ideal case of the strategy.
- Stop-loss: The position will be closed when the spread overcomes upper (lower) threshold 4 to avoid a rampant loss. The extreme spread of return of the two companies may imply that the pair may have lost the cointegration relationship.
- End of a trading period: The position will be closed if the position is kept open at the end of the quarter. This is because new pairs will be generated at the beginning of each quarter, so the pairs in this quarter may lose timeliness.

Take graph 8 as an example, the graph demonstrates the pair EPD and UGP in trading period 13Q1.



Graph 8: The blue line represents the spread of the pair. The spread breaks top-down the $upper_1$ threshold (23 Jan, open entry 1 unit) and reverts to the mean (27 Jan, take-profit 1 unit). Then the spread breaks bottom up the $lower_2$ threshold (2 Mar, open entry 2 units) and reverts to the mean (20 Mar, take-profit 2 units).

4.4.4 Capital allocation

Regarding portfolio building, capital is equally weighted for each unit of entry. As mentioned in the previous section, the capital allocated may differ in different entries. Based on the idea, we construct a framework that the total amount involved in each trading unit is \$500; the mathematical logic is described as follows:

$$|long_1| * n + |short_1| * n = n * \$500,$$

where $long_1$ and $short_1$ represent the capital invested in each pair's constituent, and n represents the number of units.

Pair trading usually requires a small size of capital as the capital generated from the short position will then be used in the long position. The long and short positions are not equally weighted as two stocks usually have different volatility. In this strategy, we will use the beta of the stocks to allocate the long and short capital in the pair trade.

4.5 Performance metrics

We will evaluate the strategy's performance based on Profit, Maximum Drawdown (MDD), the Sharpe ratio, and the Sortino ratio.

Gatev et al. (2006) proposed two measures for returns in pair trading - return on committed capital and return on actual employed capital. The return on committed capital computes the return on the capital invested in the pair at the start of the quarter. The return on actual employed capital employed excluded capital applied to the trading pair. A significant flaw of the latter is that the approach assumes the capital can be transferred between multiple portfolios instantaneously. As margin comes into play when shorting stocks, the approach is not feasible in practice, and return on committed capital is adopted to compute the return (Jakobsson, 2015).

Apart from the return, the Sharpe ratio is one of the most widely used metrics to measure risk-adjusted relative returns. The Sharpe ratio is computed as follows:

$$Sharpe\ ratio = \frac{r_{daily} - r_f}{\sigma_{\text{daily}}} * \sqrt{252},$$

where r_{daily} represents the daily return, r_f represents the risk-free rate, σ_{daily} represents the standard deviation of daily return, and $\sqrt{252}$ is the annualization factor (Fernando, 2022).

The Sortino ratio is a variant of the Sharpe ratio that uses the asset's standard deviation of negative portfolio returns, or downside deviation, rather than the total standard deviation of portfolio returns for computation. Using the Sortino ratio, investors can evaluate an investment's return for a specific degree of downside risk. The calculation of the Sortino ratio is described as follows:

$$Sortino\ ratio = \frac{r_{daily} - r_f}{\sigma_{\mathrm{d}}} * \sqrt{252},$$

where r_{daily} represents the daily return, r_f represents the risk-free rate, and σ_d represents the downside standard deviation. (Kenton, 2020)

The MDD is defined as the most significant decrease in portfolio net value from the rolling maximum up to a given period. MDD emphasizes capital preservation, the major concern for most investors, and is an indicator to compare the relative riskiness of different strategies.

A higher number of trades may increase the overall profit, but it requires considerable trading and borrowing costs simultaneously. Therefore, we will take trading costs and borrowing costs into account in backtesting. In this study, we set a trading cost of 0.4% and a borrowing cost of 3% annually (Hayes, 2022).

5. Result and Discussion

In this section, we will evaluate the proposed strategy and discuss the results obtained by applying the methodology discussed in section 4 to show the effectiveness of the proposed strategy.

5.1 Neural Network model performance

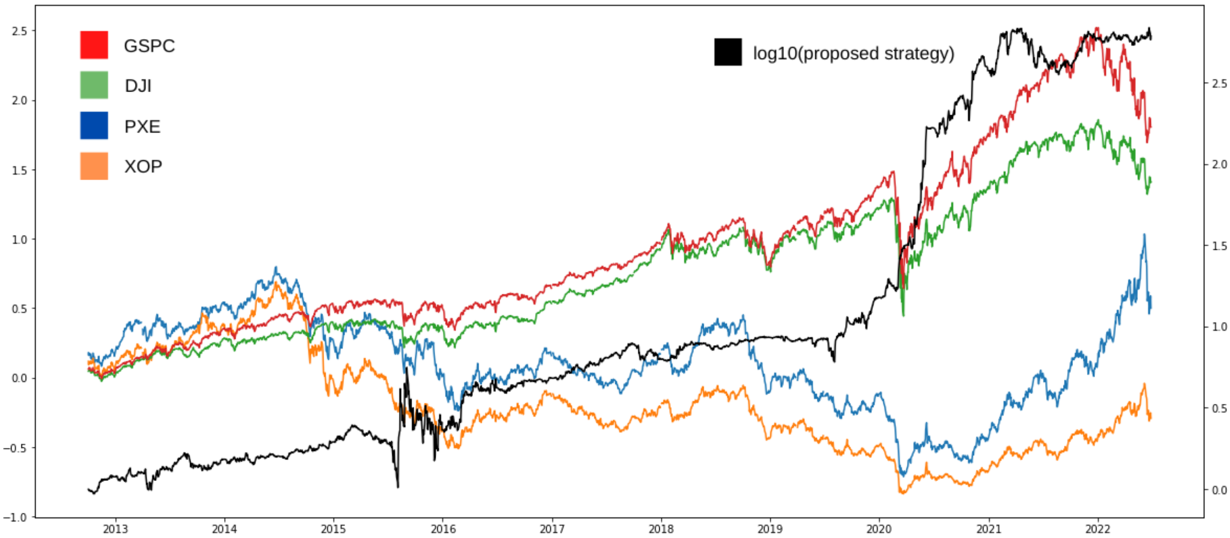
Before discussing the trading performance, we will start with the result of the neural network model. The neural network predicts the cointegrated relationship and filters the pairs that potentially lead to a bad trade. The test set size consists of 2,882 observations, and the performance is described as follows:

- Recall: 79.23% of all cointegrated pairs in each quarter were captured, and only 20.77% of cointegrated pairs are failed to capture.
- Precision: 79.74% of prediction is cointegrated among all predictions, and only 20.26% of predicted pairs are not cointegrated.

It is essential to check whether our learning algorithm is trained well. Graph 4-5 suggest that a steadily increasing average recall and precision is evidence that the network is learning well.

5.2 Strategy performance

5.2.1 Comparison with baseline index



Graph 9 shows the logarithm of cumulative returns from the proposed strategy compared to two commonly followed equity indices and two oil & gas ETFs. In this research, S&P 500 (GSPC) and Dow Jones Industrial Average (DJI) are selected as market baseline indexes for evaluating the performance of the US market; Invesco Dynamic Energy Exploration & Production ETF (PXE) and SPDR S&P Oil & Gas Exploration & Production ETF (XOP) are selected to reflect the performance of the oil and gas industry, tracking oil & gas exploration & production space.

The baseline indexes had stable growth until 2020 and then began to lose value and rebound rapidly due to COVID-19 and US loose monetary policy afterward. From 19Q3 to 20Q2, the strategy had an awe-inspiring value growth, 1,725.86%, despite the market crash. The phenom supports that the proposed strategy effectively hedges the market risks and is able to generate stable profit during a market crash.

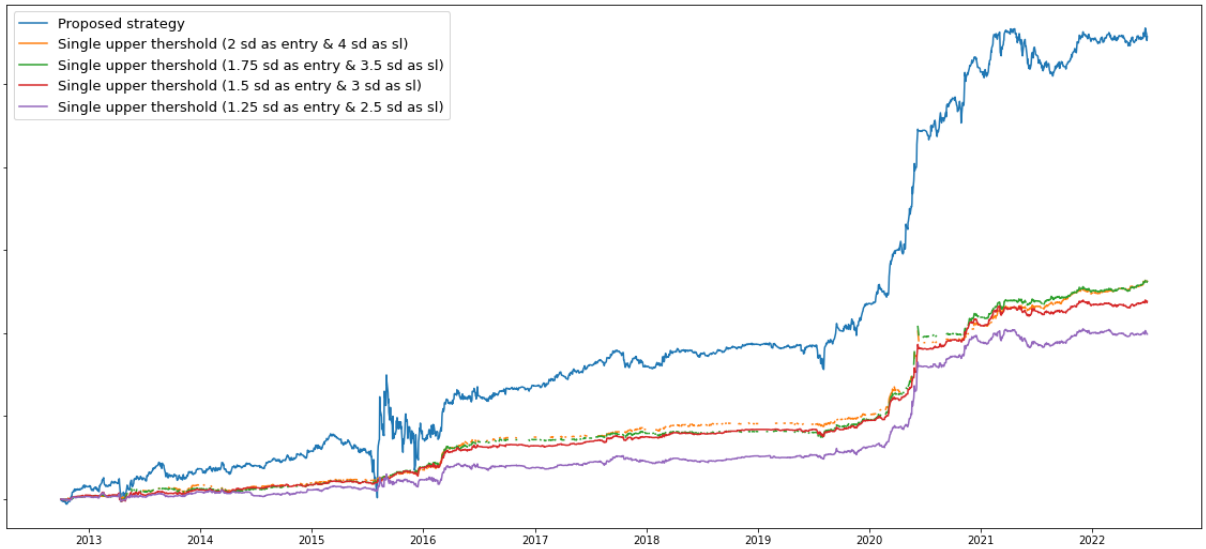
Table 2: Performance of proposed strategy & baseline index

Portfolio	MDD	Sharpe ratio	Sortino ratio	Return	CAGR
Proposed strategy	73.31%	1.218	2.113	60,954.91%	89.89%
GSPC	33.92%	0.446	0.493	180.3511%	10.85%
DJI	37.09%	0.500	0.550	140.9091%	9.19%
PXE	83.99%	0.145	0.199	50.1034%	4.15%
XOP	90.27%	-0.108	-0.151	-29.2051%	-3.39%

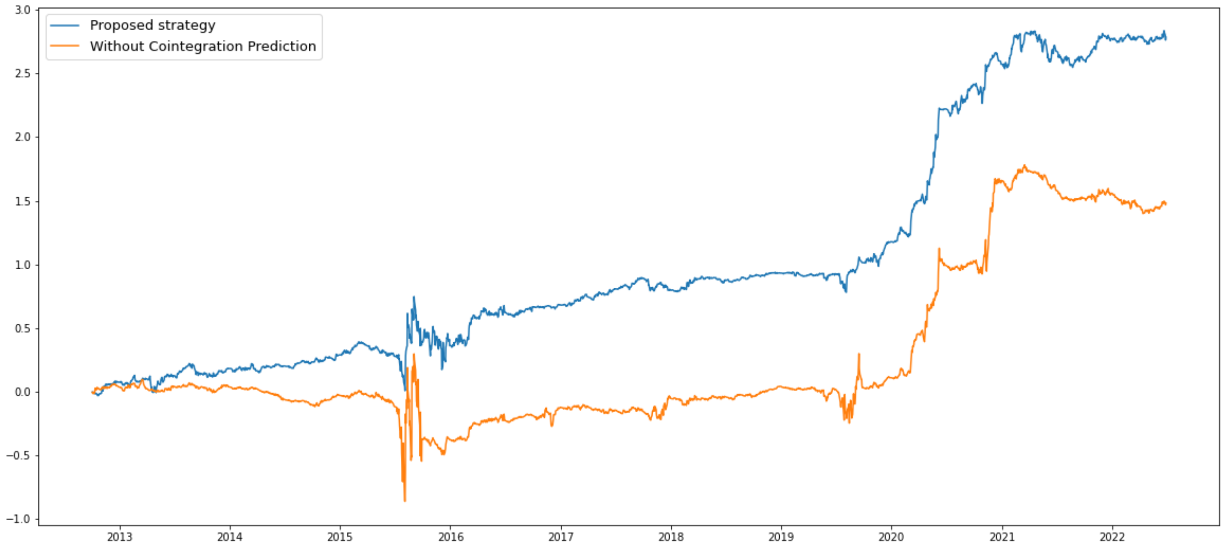
Table 2 shows the annualized risk-return metrics of the proposed strategy compared to baseline indexes after transaction costs. Annualized returns for the proposed strategy are 89.92%, compared to 10.85% for the S&P 500 index and 9.19% for the Dow Jones Industrial Average.

The proposed strategy obtains a Sharpe ratio of 1.218 versus 0.49976 produced by the DJI. To put it another way, the strategy generates a 2.4x return compared to just buying and holding DJI at the same degree of risk. At the same time, the Sortino ratio supports the performance of the proposed strategy with a value of 2.11, as contrasted to the Buy & Hold strategy policy with a maximum of 0.55.

5.2.2 Comparison with the conventional approach



Graph 10: log10(return) of proposed strategy and single entry threshold approach



Graph 11: log10(return) of proposed strategy and strategy without cointegration prediction

Table 3: Performance of proposed strategy & conventional approach

Portfolio	MDD	Sharpe ratio	Sortino ratio	Return	CAGR	No. of trade	Win rate
Proposed strategy	73.31%	1.218	2.113	60,954.91%	89.89%	27,704	68.54%
Single entry threshold (1.25 sd & 2.5 sd)	22.30%	1.349	2.427	3,844.71%	44.41%	11,180	69.73%
Single entry threshold (1.5 sd & 3 sd)	12.96%	2.023	3.490	5,908.65%	50.62%	8,512	70.65%
Single entry threshold (1.75 sd & 3.5 sd)	13.63%	1.909	4.728	10,244.24%	59.03%	6,482	72.13%
Single entry threshold (2 sd & 4 sd)	14.69%	2.358	5.831	15,399.60%	65.59%	4,910	73.64%
Multiple thresholds without cointegration prediction	88.98%	0.653	1.566	2,710.19%	39.60%	87,090	65.18%

Propose strategy represents the strategy described in section 4.4 with cointegration prediction in the pair selection process

Single entry threshold (1.25 sd & 2.5 sd) represents the strategy with single entry threshold with 1.25 sd as open entry threshold and 2.5 sd as stop-loss threshold.

Table 3 shows a better performance with cointegration prediction with respect to the conventional approach in terms of overall return, risk-adjusted return, downside volatility, and Maximum Drawdown. The strategy without cointegration prediction has only a 39.60% annual return compared to 89.89% of the proposed strategy. The potential reasons behind this include heavy transaction costs and lower hit-rate per operation.

However, the proposed strategy only outperforms the strategy with single entry threshold in terms of the overall return. Strategy with single entry threshold (2 sd as entry and 4 sd as sl) has an impressive and stable performance during the trading period, with a Sharpe ratio of 2.358 and Sortino ratio of 5.831. In addition, the MDD of the single entry threshold strategy (2 sd as entry and 4 sd as sl) is only around 15%, compared to over 90% of the drawdown of XOP, supporting the effectiveness of the cointegration prediction model.

5.3 Summary

To sum up, the cointegration prediction outperforms the traditional pair trading strategies. The key to better outperformance of the method is selecting pairs and eliminating prolonged divergent pairs to avoid bad trades. However, the single entry threshold generates a higher risk-adjusted return than the multiple entry threshold proposed in this paper. Overall, our findings support that the neural network model, which seeks to profit by longing undervalued stocks while simultaneously shorting overpriced ones, offers a higher return to variance.

Referring to graph 10 and graph 11, we want to point out that trading with the proposed strategy and other pair trading method generate strong positive spikes in return during significant stock market declines and high market turmoil (19Q3-20Q2). Previous research also addresses the same idea. Researchers found that a substantial spike in the return of pair trading strategy happened during the dot-com bubble and the 2008 financial crisis in the US market. The same performance occurred in the Moroccan market during the 2019 market crash (Krauss, 2016; Touzani, 2021). As the market exits its operations irrationally and the market’s inefficiency arises, all stocks have higher volatility during extended bear markets.

6. Conclusion and Recommendations

6.1 Discussion of the result

The primary goals of this research were to test whether leveraging the neural network and fundamental ratios of two firms may enhance the performance of pairs trading strategies. We propose an ANN for cointegration prediction and multiple entry threshold methods. The conventional strategy employed in previous literature was compared to the proposed trading strategies.

To summarize, the ANN cointegration prediction improves the strategy's performance, giving positive returns after risk-adjusted in the majority of the quarter with a high hit rate, demonstrating that the inclusion of the model improves the pair selection process. In addition, the research shows that the multiple entry threshold strategies did not outperform the single entry threshold strategy after transaction costs. As a result, it can be said that including fundamental ratios results in a more profitable trading approach than the conventional pair trading method, which relies solely on the cointegration relationship that already exists.

The proposed strategy yielded a CAGR of 89.89%. However, the result is not valid in real trading scenario because the MDD reached around 70% and lead to margin call. The proposed strategy with a single entry criterion yielded a CAGR of 65.59%. However, a few quarters with high returns inflated the graphs, misrepresenting the strategy's effectiveness. If the trade is removed after 19Q3, the strategy's annual average return is 33.7%.

6.2 Future plan

It is recommended for future research to optimize the machine learning efficiency; it is suggested to combine various vastly different mode types, such as gradient boosting trees and random forests, to achieve higher accuracy than the base learner.

For better prediction of intrinsic value, it is also important to learn from non-public information, for example, oil reserve replacement efficiency, by attending the company's presentation for analysts, in addition to the publicly available accounting data.

7. References

1. Gatev et al., E. (1999). Pairs trading: Performance of a relative value arbitrage rule. doi:10.3386/w7032
2. Dai, M., Zhong, Y., & Kwok, Y. K. (2011). Optimal Arbitrage Strategies on Stock Index Futures under position limits. Journal of Futures Markets, 31(4), 394-406. doi:10.1002/fut.20472
3. Kanamura, T., Rachev, S. T., & Fabozzi, F. J. (2009). A profit model for spread trading with an application to Energy Futures. The Journal of Trading, 5(1), 48-62. doi:10.3905/jot.2010.5.1.048
4. Breyer, C. B., & Frois, C. J. (2016). Pairs Trading: Different Weights, Methods and Markets.
5. Brunetti, M., & De Luca, R. (2020). Pre-selection methods for Cointegration-based pairs trading. SSRN Electronic Journal. doi:10.2139/ssrn.3634797
6. Lee, D., & Leung, T. (2020). On the efficacy of optimized exit rule for mean reversion trading. *SSRN Electronic Journal*. doi:10.2139/ssrn.3626471
7. Flori, A., & Regoli, D. (2020). Revealing pairs-trading opportunities with long short-term memory networks. *European Journal of Operational Research, 295*(2), 772-791. doi:10.1016/j.ejor.2021.03.009
8. Karsi, S. (2019). Pair Trading With Long-Short Term Memory.
9. Do, B. (2006). A New Approach to Modeling and Estimation for Pair Trading.
10. Diego, I., Jiang, Y., Wang, K., & Wang, W. (2019). Statistical Arbitrage by Pair Trading using Clustering and Machine Learning.
11. Bui, Q., & Ślepaczuk, R. (2020). Applying Hurst exponent in pair trading strategies on NASDAQ 100 index. *Physica A: Statistical Mechanics and Its Applications, 592*, 126784. doi:10.1016/j.physa.2021.126784
12. Yildiz, B., & Yezegel, A. (2010). Fundamental Analysis With Articial Neural Network.
13. Quirin, J. J., Berry, K. T., & O'Brien, D. (1999). A fundamental analysis approach to oil and gas firm valuation. *Journal of Business Finance & Accounting, 27*(7-8), 785-820. doi:10.1111/1468-5957.00335
14. Lv, D., Huang, Z., Li, M., & Xiang, Y. (2018). Selection of the optimal trading model for stock investment in different industries. *PLOS ONE, 14*(2). doi:10.1371/journal.pone.0212137
15. Sarmento, S. M., & Horta, N. (2020). Enhancing a pairs trading strategy with the application of machine learning. *Expert Systems with Applications, 158*, 113490. doi:10.1016/j.eswa.2020.113490
16. Jakobsson, E. (2015). Using fundamental data to find optimal portfolios.

17. Ospino, J. F., Fernandez, L., & Carchano, O. (2020). Improving pairs trading using neural network techniques and fundamental ratios. *SSRN Electronic Journal*. doi:10.2139/ssrn.3653071
18. Kim, T., & Kim, H. Y. (2019). Optimizing the pairs-trading strategy using Deep Reinforcement Learning with trading and stop-loss boundaries. *Complexity, 2019*, 1-20. doi:10.1155/2019/3582516
19. Krauss, C., Do, X. A., & Huck, N. (2016). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research, 259*(2), 689-702. doi:10.1016/j.ejor.2016.10.031
20. Touzani, Y., & Douzi, K. (2021). An LSTM and GRU based trading strategy adapted to the Moroccan market. doi:10.21203/rs.3.rs-526234/v1
21. Huck, N., & Afawubo, K. (2015). Pairs trading and selection methods: Is cointegration superior? *Applied Economics, 47*(6), 599-613. doi:10.1080/00036846.2014.975417
22. IBM Cloud Education. (2020). What are neural networks? Retrieved November 9, 2022, from https://www.ibm.com/hk-en/cloud/learn/neural-networks
23. Baheti, P. (2022). Activation functions in neural networks [12 types & use cases]. Retrieved November 9, 2022, from https://www.v7labs.com/blog/neural-networks-activation-functions#h1
24. CFI Team. (2022, November 04). Cointegration. Retrieved November 9, 2022, from https://corporatefinanceinstitute.com/resources/data-science/cointegration/
25. E. Sorensen, B. (2022). *ECONOMICS 266, Spring, 1997*. Lecture.
26. Leung, M. T., Daouk, H., & Chen, A. (2000). Forecasting stock indices: A comparison of classification and level estimation models. *SSRN Electronic Journal*. doi:10.2139/ssrn.200429
27. Figuerola-Ferretti, I., Serrano, P., Tang, T., & Vaello-Sebastii, A. (2017). Supercointegrated. *SSRN Electronic Journal*. doi:10.2139/ssrn.3005358
28. Fernando, J. (2022) *Sharpe ratio formula and definition with examples*, *Investopedia*. Investopedia. Available at: https://www.investopedia.com/terms/s/sharperatio.asp (Accessed: November 9, 2022).
29. Hayes, A. (2022) *Maximum drawdown (MDD) defined, with formula for calculation*, *Investopedia*. Investopedia. Available at: https://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp (Accessed: November 9, 2022).
30. Kenton, W. (2022, October 02). Sortino ratio: Definition, formula, calculation, and example. Retrieved November 10, 2022, from https://www.investopedia.com/terms/s/sortinoratio.asp