Pair-a-dice: Exploring Machine Learning for Pairs Trading

Caleb Lin | yl12129
Kinjalk Srivastava | ks7157

1 Introduction

Pairs trading is a market-neutral investment strategy that capitalizes on the historical correlation between two stocks. This project explores the application of machine learning techniques to enhance the pairs trading strategy. We utilize unsupervised learning algorithms, including DBSCAN, HDBSCAN, and OPTICS, to cluster and identify potential pairs among 500 S&P stocks. After preprocessing the data, including dimensionality reduction with Principal Component Analysis (PCA), we apply these clustering algorithms to generate candidate pairs. These pairs are then validated using the Dickey-Fuller test for stationarity and correlation analysis to ensure their suitability for trading.

We employ Long Short-Term Memory (LSTM) networks to predict spreads and compute trading strategies for the validated pairs. Our approach is benchmarked against a standard trading strategy provided in our class. The results demonstrate a slight improvement in performance, highlighting the effectiveness of integrating machine learning techniques into pairs trading. This report provides a comprehensive analysis of the methodology, experimental design, and key findings, showcasing the potential of machine learning to enhance the profitability and robustness of pairs trading strategies.

2 Conceptual Framework

2.1 Overview: Possible Area of Improvement with Machine Learning

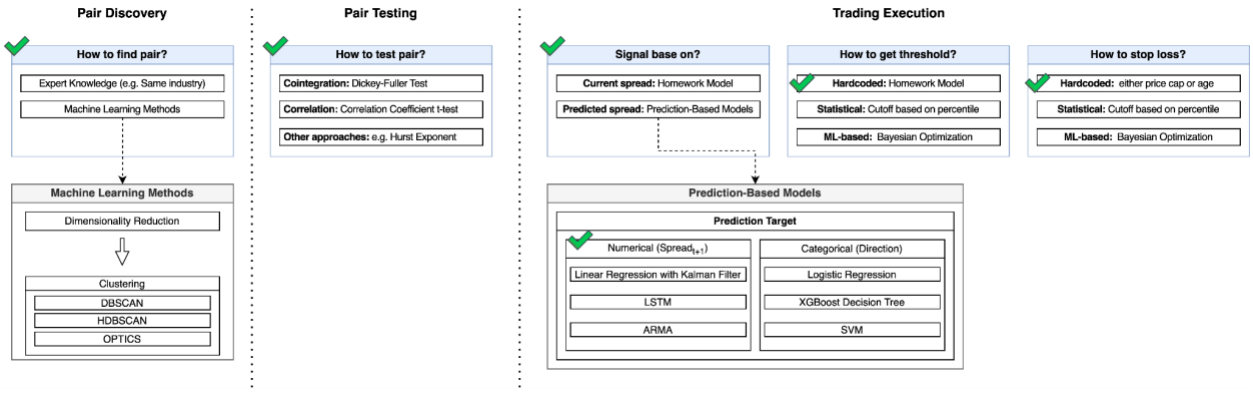


Figure 1: Conceptual Overview

Pairs trading normally involves 3 key steps: pair discovery, pair testing, and trade execution. Each of these areas presents opportunities for enhancing the process through the application of machine learning techniques.

In pair discovery, the conventional approach relies on expert knowledge, such as identifying pairs within the same industry based on fundamental analysis. However, this method may overlook potentially profitable pairs that are less obvious. Additionally, testing every possible combination of stocks leads to the multiple

comparisons problem, where the likelihood of finding spurious relationships increases with the number of pairs tested. Machine learning offers an alternative by first employing dimensionality reduction techniques like PCA to capture the most important information in a lower-dimensional space. Then, clustering algorithms such as DBSCAN, HDBSCAN, and OPTICS are applied to uncover novel pair relationships based on the inherent structure of the data. By focusing on testing pairs within clusters, the total number of potential pairs is significantly reduced, mitigating the multiple comparisons problem while still capturing meaningful relationships.

For pair testing, this project does not introduce new methods but rather investigates whether established tests, such as assessing cointegration using the Dickey-Fuller test, and evaluating correlation through the correlation coefficient test, perform better when combined with specific clustering methods from the pair discovery phase. The aim is to identify the most effective combination of clustering and testing techniques for identifying profitable pairs.

In the trade execution phase, the signal to enter or exit a trade is typically based on the current spread in relation to a threshold derived from a hardcoded or statistical model. Machine learning, particularly prediction-based models like Long short-term memory (LSTM) neural networks, can be employed to predict the future spread and generate trading signals dynamically. This approach has the potential to capture more nuanced patterns in the data and adapt to changing market conditions.

By systematically exploring the application of machine learning techniques in each stage of the pairs trading process, Pair-a-dice aims to identify the most effective methods for enhancing profitability and robustness.

2.2 Pair Discovery

2.2.1 Machine Learning in Lieu of Expert Knowledge

In this project, we propose a machine learning-based approach to pair discovery that consists of two main steps: dimensionality reduction and clustering. First, we employ dimensionality reduction techniques such as PCA to capture the most important information from the stock data in a lower-dimensional space. This step helps to reduce noise and focus on the most relevant features. Then, we apply clustering algorithms such as DBSCAN, HDBSCAN, or OPTICS to group stocks based on their inherent similarities. By focusing on testing pairs within clusters, we significantly reduce the total number of potential pairs while still capturing meaningful relationships.

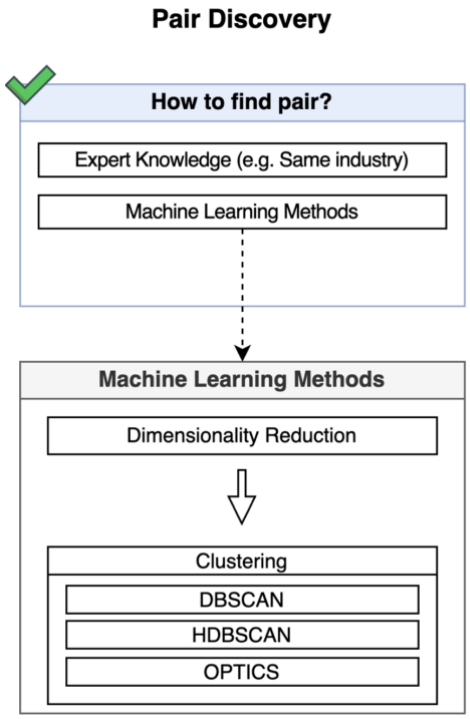


Figure 2: Pair Discovery Framework.

This approach addresses several limitations of the traditional expert knowledge-based method:

1. Uncovering hidden patterns: Machine learning algorithms can identify complex, non-linear relationships between stocks that may not be apparent to human analysts, potentially uncovering less obvious but profitable pairs.
2. Scalability: Automated pair discovery can efficiently analyze vast amounts of data and handle many stocks, enabling the discovery of a wider range of potential pairs. This is particularly important as the number of stocks and market complexity grows, making manual analysis increasingly difficult and time-consuming.
3. Adaptability: Machine learning models can be updated and retrained as new data becomes available, allowing them to adapt to changing market conditions and identify new pair relationships over time. This is an advantage over expert knowledge, which may not quickly adapt to evolving market dynamics.

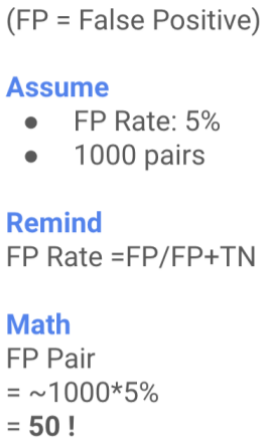


Figure 2.1: Multiple Comparison Problem Example.

Furthermore, our approach helps to mitigate the multiple comparisons problem that arises when testing every possible combination of stocks. The multiple comparisons problem refers to the increased likelihood of finding false positive pairs as the number of pairs tested grows. The problem is illustrated in the Figure 2.1. By reducing the search space through dimensionality reduction and clustering, we can focus on testing a smaller number of more promising pairs, increasing the likelihood of discovering genuine, profitable relationships.

In contrast, the traditional expert knowledge-based approach relies heavily on analysts or traders using their understanding of the market and industry dynamics to identify potential pairs. While this can be effective for identifying well-known or obvious pairs, it may overlook less apparent but potentially profitable relationships. Additionally, the manual nature of this approach limits its scalability and adaptability to changing market conditions.

It is important to note that the machine learning approach can work in conjunction with expert knowledge. The ML-based method can serve as a prescreening tool to identify pairs with potential, while analysts can later validate and further investigate the real connections between these pairs. This combination of machine learning and human expertise can provide analysts with valuable intuition and a more targeted set of pairs to analyze, ultimately improving the efficiency and effectiveness of the pair discovery process.

2.2.2 Dimensionality Reduction using PCA

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction. It transforms the data into a new coordinate system where the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. This helps in simplifying the dataset while retaining most of the variation present in the data.

In our project, PCA was applied to the financial data from WRDS for the following reasons:

1. Data Simplification: The original dataset had dates as columns and various financial ratios, resulting in high-dimensional data. PCA simplified this complex dataset by reducing its dimensionality.
2. Variance Preservation: PCA retained the most significant variances in the dataset, ensuring that the essential information was preserved for further analysis.
3. Cluster Formation: By reducing the number of variables, PCA made it easier to identify clusters of similar stock series, enhancing the effectiveness and accuracy of the clustering process.
4. Improved Pair Computation: The resultant clusters from PCA-enabled data provided a clear basis for computing pairs, leading to more meaningful and reliable pairs of stock series for subsequent analysis.

2.2.3 Clustering with OPTICS, DBSCAN and HDBSCAN

1. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN is a density-based clustering algorithm that identifies clusters based on the density of data points. It groups together points that are closely packed together while marking points that lie alone in low-density regions as outliers.

2. OPTICS (Ordering Points To Identify the Clustering Structure)
OPTICS is an extension of DBSCAN that creates an augmented ordering of the database representing its density-based clustering structure. It can handle varying densities more effectively than DBSCAN.

3. HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)
HDBSCAN is an extension of DBSCAN that performs hierarchical clustering by converting the DBSCAN clustering process into a hierarchical one. It can find clusters of varying densities and is robust to parameter selection.

Justification:

The choice of OPTICS, DBSCAN, and HDBSCAN as clustering techniques for our financial data project is optimal due to their ability to handle clusters of varying shapes and densities, which is crucial for financial datasets characterized by irregular and non-uniform distributions. These density-based algorithms excel in identifying meaningful clusters while effectively managing noise and outliers, a common feature in financial data. Unlike K-Means, which assumes spherical cluster shapes and requires the pre-specification of the number of clusters, these algorithms adapt to the data's intrinsic structure without such rigid assumptions. K-Means is suboptimal for financial data as it is sensitive to outliers, struggles with varying cluster densities, and can produce misleading results if the data does not conform to its underlying assumptions of evenly sized, spherical clusters. This flexibility and robustness make OPTICS, DBSCAN, and HDBSCAN more suited to the complexities of financial data analysis.

2.3 Pair Testing

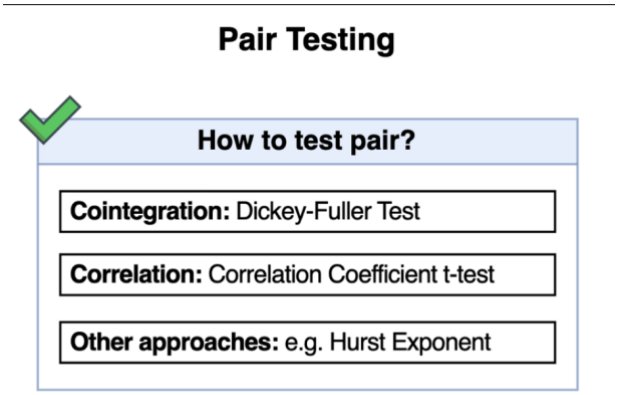


Figure 3: Possible Tests for Successful Pairs.

Cointegration

Cointegration is a statistical property of a collection (usually a pair) of time-series variables. If two or more series are individually integrated (i.e., non-stationary) but some linear combination of them results in a stationary series, they are said to be cointegrated. This suggests a long-term equilibrium relationship despite short-term fluctuations. In finance, cointegration between stock prices typically indicates that their prices, although they may move apart at times, will tend to revert to a mean value, maintaining a stable relationship over time.

Cointegration: Augmented Dickey-Fuller (ADF) Test

The Augmented Dickey-Fuller (ADF) Test is commonly used to test for a unit root in a univariate time series sample. It is an enhancement of the Dickey-Fuller test that allows for higher-order regressive processes. We use this test in the context of cointegration to check the null hypothesis that a time series sample involves a unit root, suggesting non-stationarity. In cointegration analysis, the ADF test is applied to the residuals obtained from the regression model of the two time series. If the ADF test indicates stationarity of the residuals, it supports the presence of cointegration between the series.

Correlation

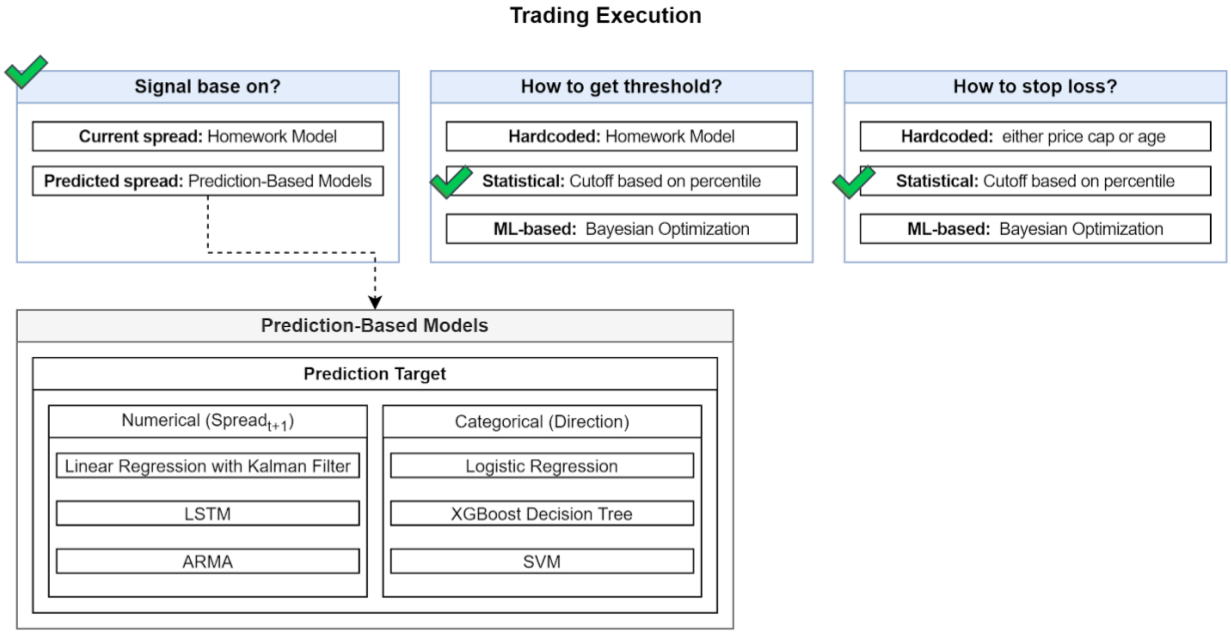
Correlation measures the linear relationship between two variables, reflecting how changes in one variable are associated with changes in another. In financial terms, correlation can help identify how securities move in relation to one another. A high correlation between two stocks implies that their prices move similarly, which can be essential for diversification, risk management, or pair trading strategies.

Test

In our project, we first filtered out noise from our clustered stock data to focus on significant groupings. We then transposed the stock price data, arranging the stocks in columns to simplify comparative analysis. We set a stringent correlation threshold to identify only the most closely related stock pairs within each cluster. By iterating through each cluster, calculating correlation matrices, and selecting stock pairs exceeding our correlation threshold, we were able to pinpoint potential trading opportunities based on

closely correlated movements. This method effectively utilizes advanced clustering techniques and correlation analysis to uncover meaningful relationships in financial data, aiding in the development of strategies like pairs trading.

2.4 Trade Execution



. Figure 4: Methodology for Trade Execution.

In the trade execution phase, there are three main areas where we can potentially improve upon the traditional pairs trading model: signal generation, threshold tuning, and risk management. In this project, we focus on enhancing the signal generation process.

Traditionally, the trading signal is based on the current spread between the two stocks in a pair, comparing it to a predefined threshold derived from a hardcoded or statistical model. We propose using machine learning, specifically time series models like LSTM, to predict the future spread and generate trading signals dynamically.

Our approach involves using a lookback window to train the time series model to predict the spread at time t+1. We then act on this prediction at time t. The intuition behind this approach is that by predicting the future spread, we can anticipate potential mean-reversion opportunities and enter trades more promptly, potentially capturing more profit.

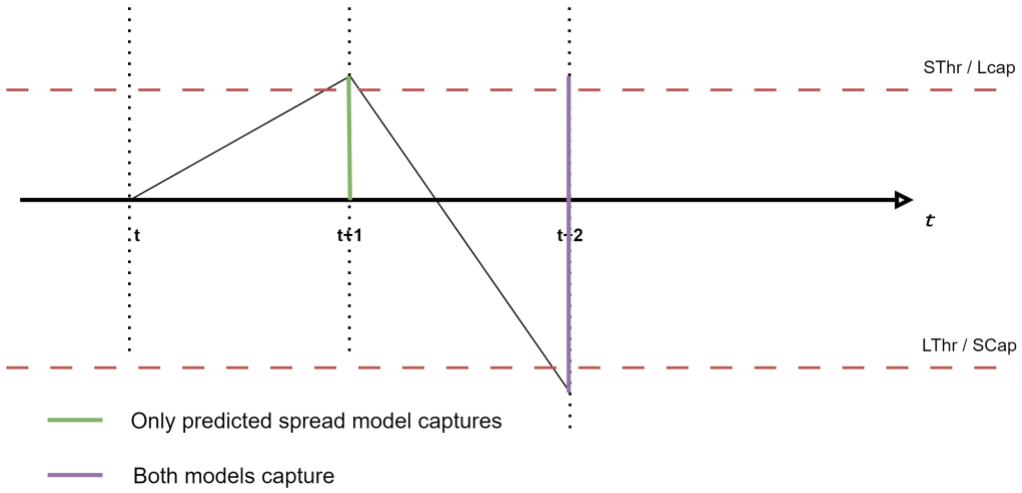


Figure 5: Overview for Current and Predicted Spreads Models

As illustrated in the figure 5, the predicted spread crosses the threshold earlier than the actual spread. By acting on the predicted signal, we can enter the trade sooner (at t) compared to waiting for the actual spread to cross the threshold (at t+1). This earlier entry point allows us to capture more of the potential profit from the mean-reversion of the spread (the green line).

	Actual (spread)	OldSignal	Predicted	NewSignal
t-(lookback_period-1)				
…				
t-2				
t-1	0	0		
Today	0.5	0	2 !	-1
	2	1	0.5	0
	0			

Figure 6: Illustration on how Predicted Spread Captures Information

In practice, instead of predicting the spread directly, we predict the three future returns used in the homework model ("zdiff5", "zdiff10", "zdiff20") and use these predictions as trading signals, as illustrated in the figure 6. By doing so, we maintain consistency with the established framework while leveraging the power of machine learning to generate more timely and potentially profitable signals.

It's important to note that while we focus on signal generation in this project, position sizing and risk management are also crucial aspects of trade execution that could benefit from machine learning techniques. These areas present opportunities for future research and improvement in the pairs trading strategy.

3 Methodology

3.1 Data Source and Tools

3.1.1 Data

In our project, we utilized data from the Wharton Research Data Services (WRDS), specifically drawing from the CRSP (Center for Research in Security Prices) and Compustat databases. Our dataset included daily stock prices and monthly fundamental ratios, covering a period from January 1, 2016, to December 31, 2020. WRDS provides a comprehensive collection of financial data, where CRSP is renowned for its high-quality stock price data, which is crucial for accurate financial analysis and trading strategy development. Compustat offers detailed and standardized financial and accounting information on a wide range of companies, making it invaluable for performing in-depth financial analysis and understanding market trends. Following is a table listing all the ratios used in the project.

Ratio	Full Form	Definition
bm	Book-to-Market Ratio	Measures a company's book value relative to its market value, indicating how much a company would be worth if liquidated at book values compared to its market price.
pe_exi	Price-to-Earnings Excluding Extraordinary Items Ratio	A valuation ratio comparing the current share price relative to its per-share earnings, excluding non-recurring items, to assess underlying profitability.
cfm	Cash Flow Margin	Indicates the percentage of revenue that becomes cash flow from operations, showing operational efficiency and liquidity.
npm	Net Profit Margin	Shows the percentage of revenue that has turned into profit, indicating how well a company converts sales into net income.
roa	Return on Assets	Measures how effectively a company uses its assets to generate earnings, reflecting the profitability of total investments.
roe	Return on Equity	Indicates how effectively a company uses investments from equity shareholders to generate earnings and growth.
gprof	Gross Profit Margin	Represents the portion of each dollar of revenue that the company retains as gross profit after accounting for the cost of goods sold.
totdebt_invcap	Total Debt to Invested Capital	The ratio of total debt to the total capital invested in the company, indicating the leverage and potential financial risk.

capital_ratio	Capital Ratio	Generally reflects the financial stability of a company by comparing total capital to risky assets.
debt	Debt Ratio	Measures the proportion of a company's total liabilities to its total assets, showing the degree of leverage and financial risk.
ebitda	Earnings Before Interest, Taxes, Depreciation, and Amortization	A measure of a company's overall financial performance and is used as an alternative to simple earnings or net income.
it_debt	Interest Coverage Ratio	A measure of a company's ability to meet its interest payments, calculated as earnings before interest and taxes divided by interest expenses.
cash_debt	Cash to Debt Ratio	Shows how financially capable a company is of covering its total debt with its total cash.
fcf_ocf	Free Cash Flow to Operating Cash Flow Ratio	Compares free cash flow to operating cash flow, indicating the proportion of cash flow from operations that remains after capital expenditure.
de_ratio	Debt to Equity Ratio	Compares total liabilities to shareholders' equity, indicating the proportion of equity and debt used to finance a company's assets.
at_turn	Asset Turnover Ratio	Measures a firm's efficiency at using its assets in generating sales or revenue; the higher the number the better.
ptb	Price to Book Ratio	Compares a firm's market to book value by dividing price per share by book value per share, used to identify undervalued or overvalued securities.

3.1.2 Environment

Google Collaboratory (Colab), a cloud-based platform by Google, was our coding environment of choice for its collaborative features and easy accessibility. With real-time collaboration, both team members could work simultaneously, enhancing efficiency and facilitating quick problem-solving. Additionally, Colab's access to powerful computing resources allowed us to handle our dataset seamlessly. Overall, it proved instrumental in streamlining our project workflow and promoting effective teamwork.

3.1.3 Tools

In our project, we extensively utilized a variety of Python libraries to streamline our data analysis and modeling tasks. *Pandas* and *NumPy* provided powerful tools for data manipulation and numerical computations, enabling us to efficiently handle and preprocess our dataset. We employed *statsmodels* specifically for its Augmented Dickey-Fuller (ADF) test implementation, which was essential for testing stationarity in time series data. For our prediction model, we leveraged *TensorFlow* and *Keras*, which

offered comprehensive frameworks for building and training deep learning models. Additionally, *scikit-learn* was instrumental for implementing various clustering algorithms and performing model evaluation and selection. Finally, *Matplotlib* and *Seaborn* were invaluable for visualizing our data and model outputs, allowing us to gain insights and communicate results effectively. Overall, these libraries collectively provided a robust toolkit for conducting comprehensive data analysis, modeling, and visualization tasks in our project.

3.2 Data Preprocessing

In our data preparation process, we meticulously cleaned the financial dataset to ensure robust analysis. First, we filtered out stocks with low trading volumes, defined as a turnover of less than 0.1% of total shares outstanding, to eliminate noise and anomalies. We then split the dataset into training and testing sets based on a cutoff date of January 1, 2020, facilitating effective model training and validation. To address missing values, we created a pivot table for stock prices, dropped stocks with more than half of their price data missing, and applied forward filling for remaining gaps. Additionally, we checked for NaN values and filled them with the mean for each specific stock identifier (PERMNO), ensuring data completeness. We refined our financial ratios data to focus on the relevant period, ensuring consistency and accuracy. Using the `*missingno*` library, we visualized missing data to further assess and manage data quality, resulting in a well-prepared dataset ready for advanced financial analysis and modeling.

3.3 Experimental Design

3.3.1 Overview of experiments

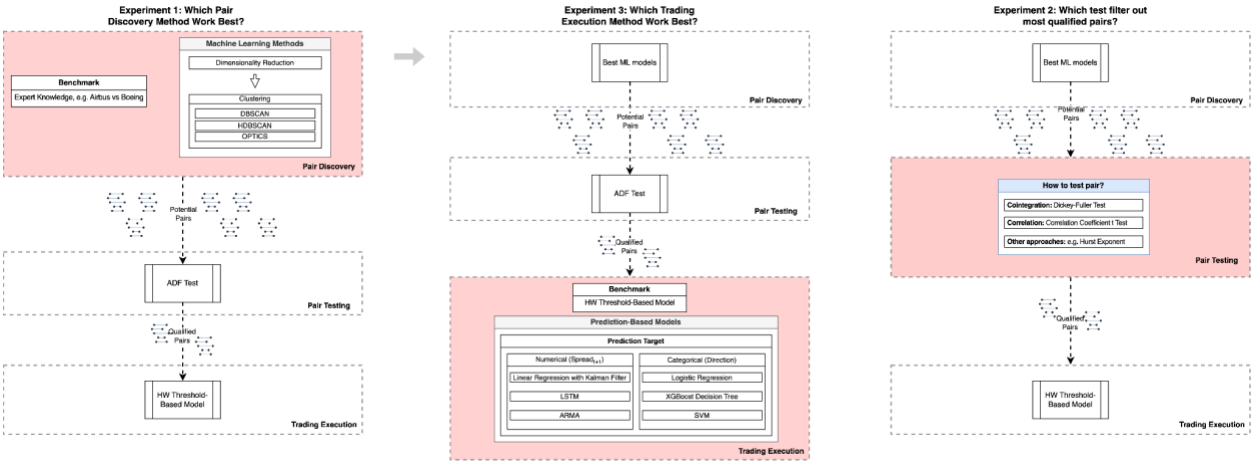


Figure 7: Overview of Experiments

In our project, we delve into ML applications in the three key steps: pair discovery, pair testing, and trade execution (mostly about signal generation). To assess the impact of machine learning techniques on each step, we design a series of controlled experiments. In each experiment, we alter only one step of the process

while keeping the others consistent with the benchmark model. This approach allows us to isolate the effect of the machine learning intervention and evaluate its performance compared to the traditional methods, as indicated in figure 7.

For the experiments involving Pair Discovery and Pair Testing, we use the benchmark trade execution model, which is the same model used in our pair trading homework. To facilitate easier back testing, we implement this model using Python instead of the original Excel implementation.

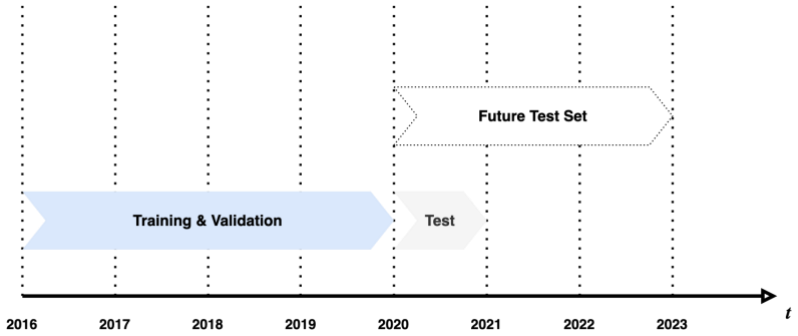
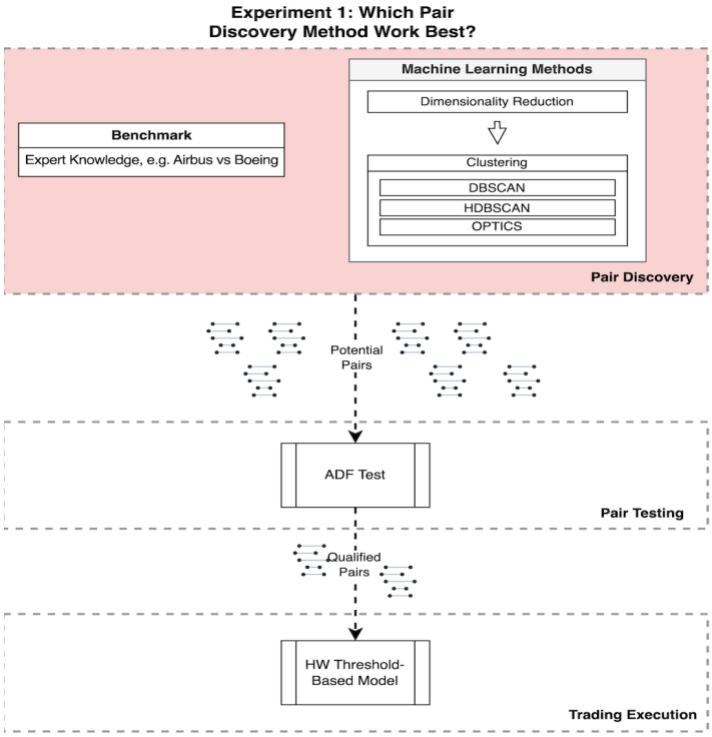


Figure 8: Train and Test Split

As shown in figure 8, the data is split into train and test sets as follows:

- Train: January 1, 2016 - December 31, 2019
- Test: January 1, 2020 - December 31, 2020

3.3.2 Experiment for Pair Discovery



In this experiment, we focus on improving the pair discovery step using machine learning techniques while keeping the pair testing and trade execution steps consistent with the benchmark model. We apply dimensionality reduction using PCA and clustering algorithms such as DBSCAN, HDBSCAN, and OPTICS to identify potential pairs. The performance of these machine learning-based pair discovery methods is then compared to the traditional approach using the benchmark pair testing and trade execution models (the homework result).

3.3.3 Experiment for Pair Testing

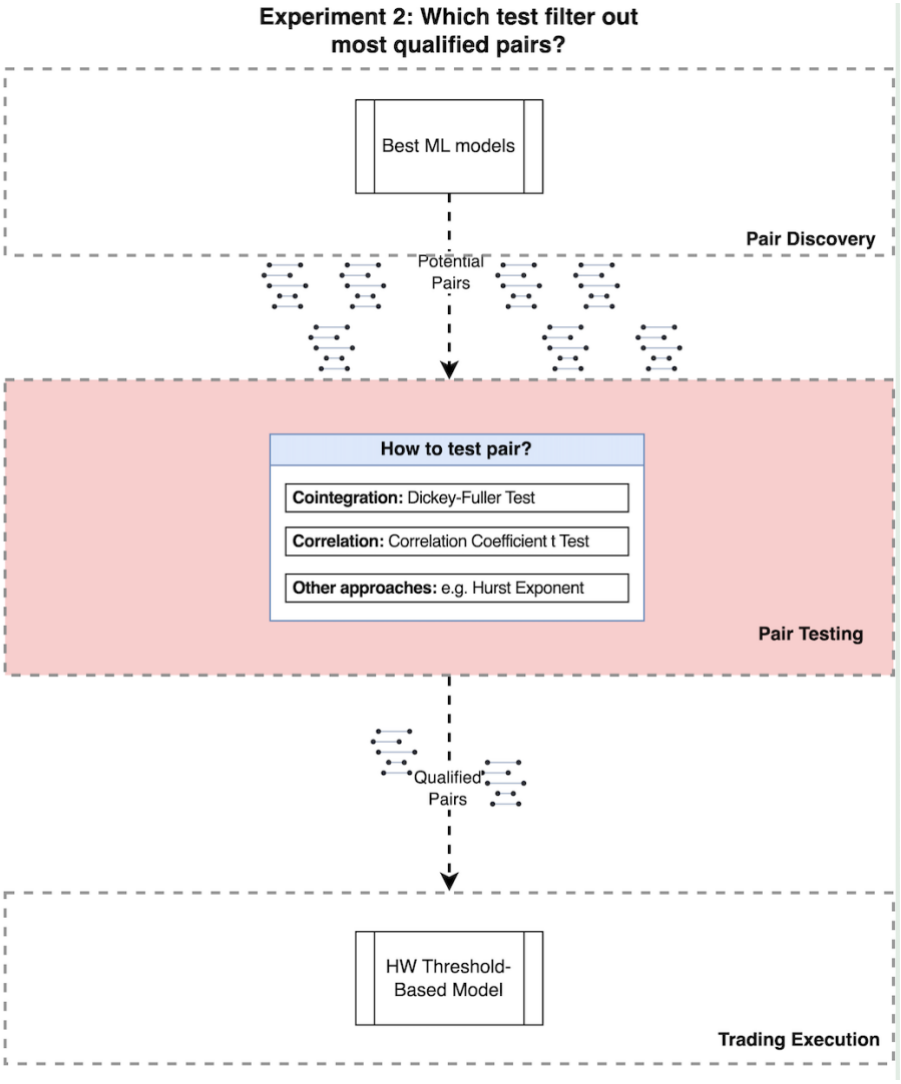


Figure 9: Methodology for finding Qualified Pairs

For the pair testing experiment, we investigate whether different pair testing methods, such as cointegration tests, correlation tests, and the Hurst exponent, perform better when combined with specific clustering methods from the pair discovery phase. We keep the pair discovery and trade execution steps fixed, using the benchmark models for both. By comparing the performance of various combinations of pair testing and clustering methods, we aim to identify the most effective approach for selecting tradable pairs.

3.3.4 Experiment for Trade Execution

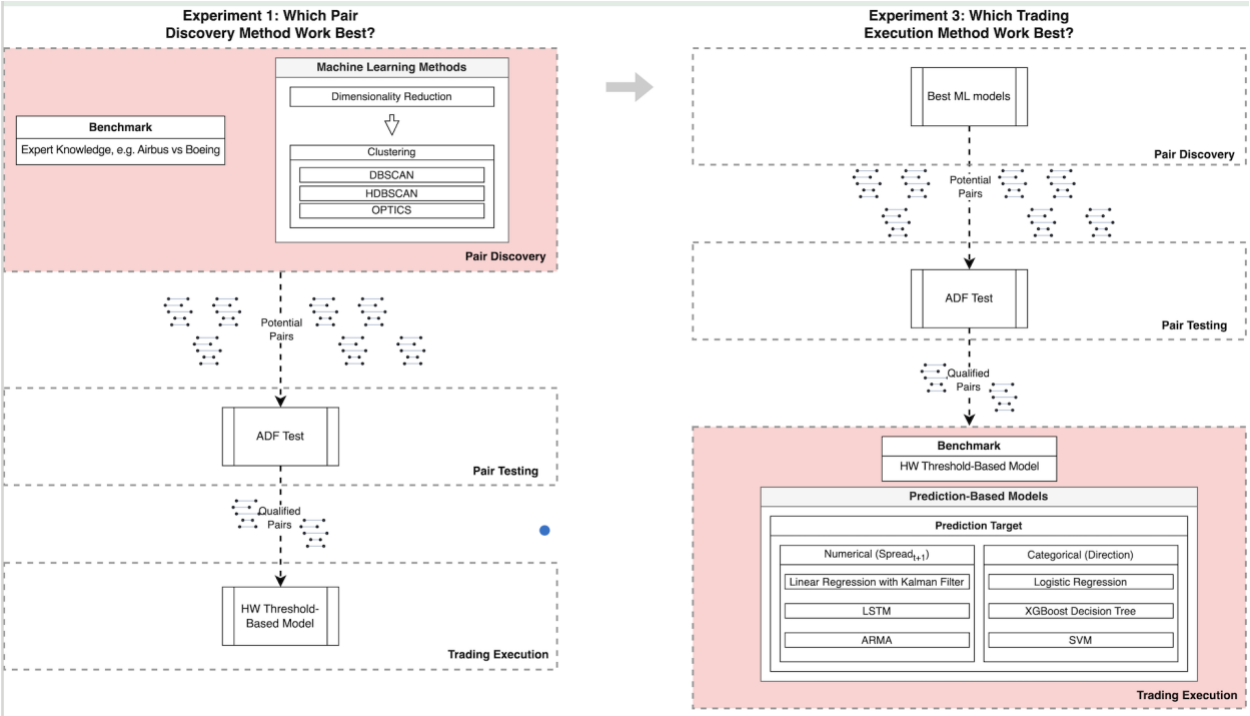


Figure 10: Finalizing Trade Execution

In the trade execution experiment, we focus on enhancing the signal generation process using machine learning techniques, specifically time series models like LSTM. We train the LSTM model using a lookback window to predict the spread at time t+1 and generate trading signals based on these predictions. The pair discovery and pair testing steps remain consistent with the benchmark model. We compare the performance of the machine learning-based trade execution model to the benchmark model to assess its effectiveness in capturing profits and managing risks.

By conducting these controlled experiments and systematically evaluating the impact of machine learning techniques on each step of the pairs trading process, we aim to identify the most promising approaches for improving the overall strategy's profitability and robustness.

4 Results

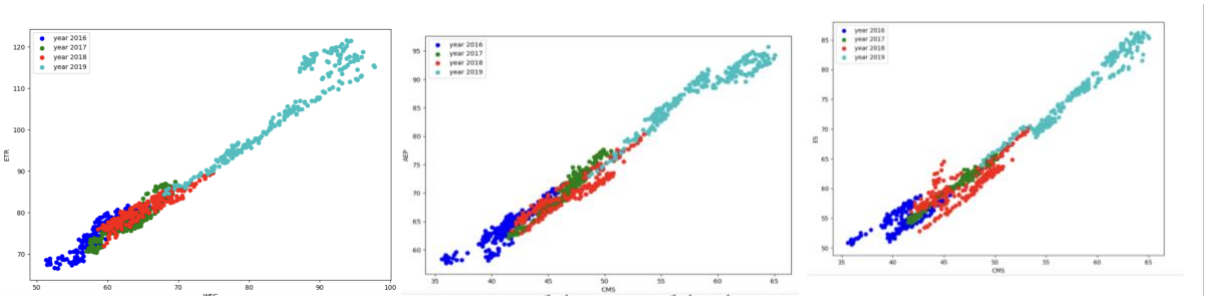
4.1 Results on Pair Discovery

4.1.1 Clustering Significantly Reduces Pairs for Cointegration Testing

In our analysis, we aimed to significantly reduce the number of stock pairs subjected to the Dickey-Fuller Test for cointegration by leveraging advanced clustering techniques. The initial dataset consisted of 498 stocks, which could potentially form approximately **61,876** pairs for evaluation. To streamline this process, we employed clustering algorithms such as HDBSCAN, DBSCAN, and OPTICS to identify subsets of pairs that are more likely to exhibit meaningful relationships. HDBSCAN identified 342 pairs, DBSCAN yielded 1,066 pairs, and OPTICS resulted in 162 pairs for further evaluation. These refined sets were subjected to cointegration tests, which further narrowed down the pairs: only 11 pairs from HDBSCAN, 20 from DBSCAN and 4 pairs from OPTICS displayed cointegration. This multi-step approach effectively reduced the workload and focused our analysis on the most promising stock pairs.

Clustering Technique	Pairs Yielded	Pairs with Cointegration
HDBSCAN	342	11
DBSCAN	1,066	20
OPTICS	162	4

4.1.2 ML Techniques Successfully Identify Stationary Pairs



OPTICS CLUSTERING HDBSCAN CLUSTERING DBSCAN CLUSTERING

Figure 11: Stationary Pairs using OPTICS, HDBSCAN, and DBSCAN

These plots in figure 11 illustrate the identification of stationary pairs using three clustering techniques: OPTICS, HDBSCAN, and DBSCAN, and the cointegration test. Each scatter plot shows data points from different years (2016-2019), with the clusters highlighting potential pairs that maintain a stable relationship over time.

4.1.3 DBSCAN Excels at Identifying Out-of-Sector Pairs

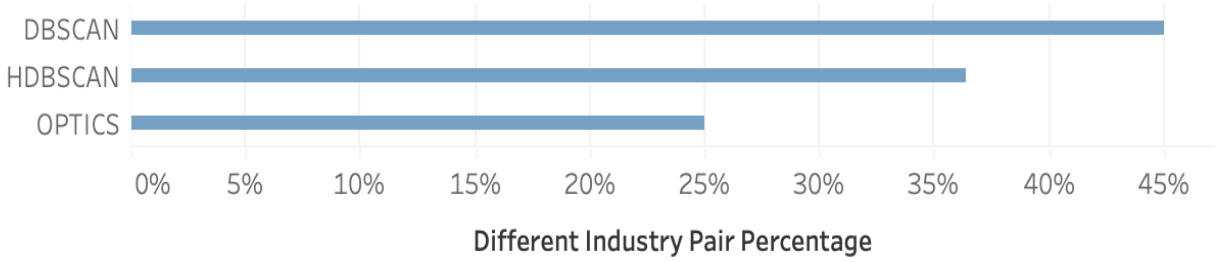


Figure 12: Different Industry Pair Percentage

This bar chart compares the effectiveness of three clustering techniques—DBSCAN, HDBSCAN, and OPTICS—in identifying out-of-industry pairs after the cointegration test. The percentage of pairs that are not in the same industry indicates our models' ability to find less obvious pairs that are less likely to be identified by expert knowledge. In the figure 12, DBSCAN is shown to identify the highest percentage of different industry pairs at around 45%, followed by HDBSCAN at approximately 30%, and OPTICS at around 25%. This indicates that DBSCAN may be more effective at recognizing pairs from different industries.

Figure 13: Common Pair Sector Distribution

This graph above identifies industry pairs that are most frequently found in our pair trading strategy. The combination of 'Electric Power Generation, Transmission, and Distribution' with 'Natural Gas Distribution' stands out, representing a significant portion of pairs, indicating a strong co-movement and potential for profitable trades. Other notable pairs include 'Depository Credit Intermediation' with 'Management of Companies and Enterprises', suggesting that diverse sector combinations can be effectively leveraged for pair trading strategies. This diversity in pair combinations enhances the robustness of the trading strategy by capturing different market dynamics.

4.1.4 Sharpe Ratio Filter in the Train Set Enhances Performance in the Test Set

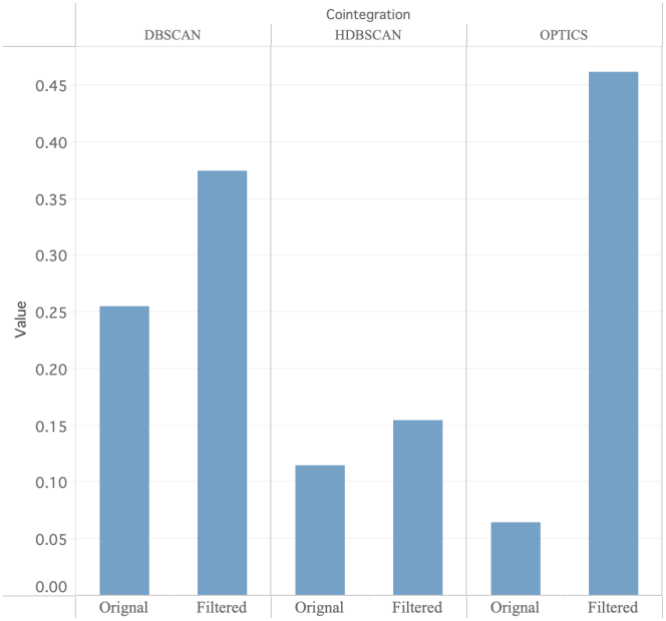


Figure 14: Original vs Filtered Pairs Sharpe Distribution

The graph above demonstrates that filtering out pairs with a Sharpe Ratio below 0.3 using the training set improves the performance of all clustering approaches (DBSCAN, HDBSCAN, and OPTICS) on the test set, as indicated by higher Sharpe Ratio. This suggests that incorporating a Sharpe ratio filter into the pair selection process is an effective way to identify and remove poor-performing pairs, leading to a more profitable and robust pairs trading strategy that can be applied to new, unseen data.

4.2 Results on Pair Testing

4.2.1 Correlation Tests Outperform Cointegration in Clustering Combinations

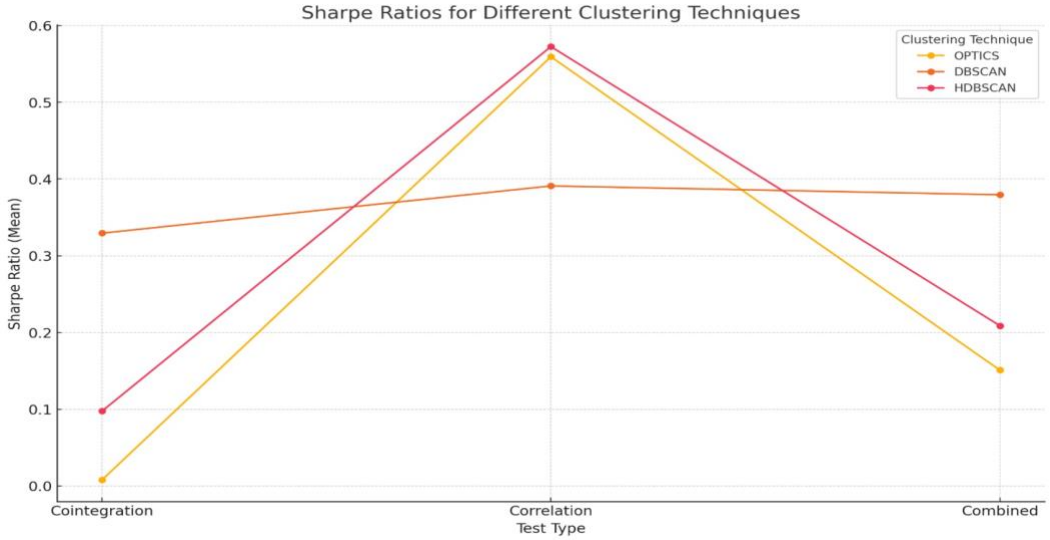


Figure 15: Sharpe Ratios for Different Clustering Techniques

The graph showing Sharpe ratios for different clustering techniques (OPTICS, DBSCAN, and HDBSCAN) across three test types (Cointegration, Correlation, and Combined) highlights significant variations in risk-adjusted returns, where "Combined" means applying both tests. All clustering techniques exhibit their highest Sharpe ratios with the Correlation test, with DBSCAN peaking around 0.55, followed closely by HDBSCAN at 0.5. OPTICS shows the lowest Sharpe ratio increase, peaking around 0.4. This indicates that correlation-based strategies are more effective in enhancing risk-adjusted returns. While the combined test type shows a slight decline from the peak correlation values, it still maintains higher Sharpe ratios than the Cointegration test, suggesting a balance between different testing methods might be beneficial.

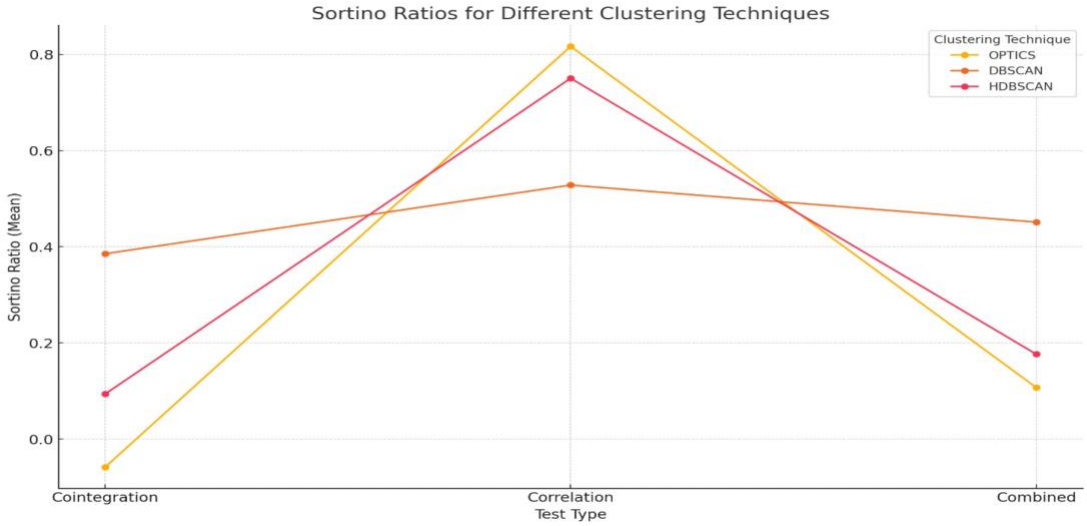


Figure 16: Sortino Ratios for Different Clustering Technique

The Sortino ratios graph illustrates the downside risk-adjusted returns for OPTICS, DBSCAN, and HDBSCAN across Cointegration, Correlation, and Combined tests. Like the Sharpe ratio trends, the Correlation test type consistently provides the highest Sortino ratios across all clustering techniques. DBSCAN achieves the highest peak around 0.85, followed by OPTICS at 0.8, and HDBSCAN at 0.7. This suggests that correlation-based tests are particularly effective in mitigating downside risk while enhancing returns. The combined test results in a noticeable decline in Sortino ratios compared to the correlation test but still outperforms the cointegration test, reinforcing the importance of correlation in trading strategies.

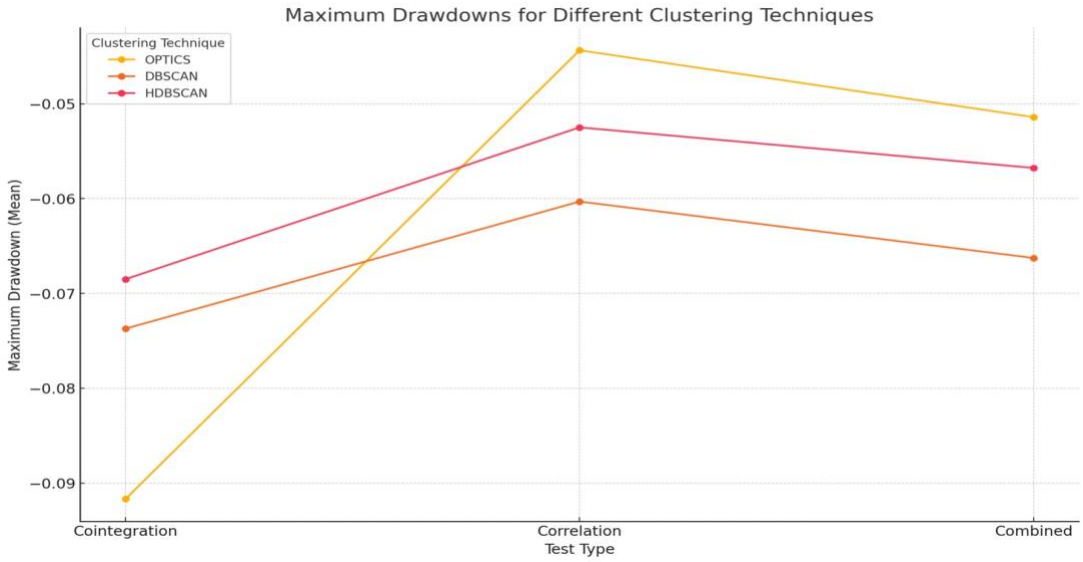


Figure 17: Maximum Drawdowns for Different Clustering Techniques

The maximum drawdown graph presents the mean drawdowns for OPTICS, DBSCAN, and HDBSCAN across the three test types, indicating the extent of potential losses. DBSCAN demonstrates the best performance with the least drawdown during the Correlation test, reaching around -0.05, which signifies minimal losses compared to OPTICS and HDBSCAN. Both OPTICS and HDBSCAN show improvement in drawdowns during the Correlation test, with drawdowns improving to around -0.06. However, in the combined test, OPTICS experiences a slight increase in drawdown to around -0.07, whereas HDBSCAN's drawdown remains relatively stable. These results underscore the efficacy of the DBSCAN clustering technique in minimizing losses, particularly when combined with correlation-based testing.

4.2.2 Cointegration Proves Superior for Out-of-Sector Pair Discovery

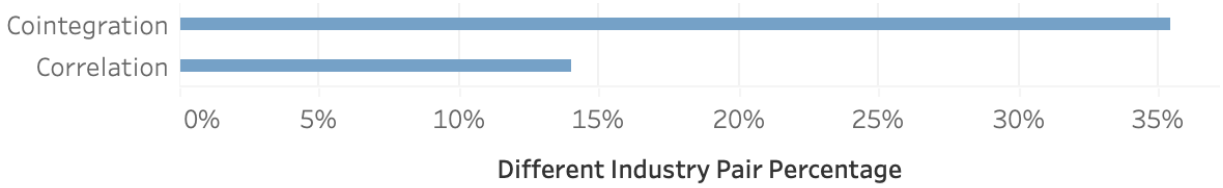


Figure 18: Cointegration Vs Correlation for Out-of-Sector Pairs

This bar chart compares the effectiveness of Cointegration and Correlation tests in identifying out-of-sector pairs. Cointegration is shown to be significantly better, identifying around 35% of different industry pairs, while Correlation identifies roughly 15%. This indicates that Cointegration is more adept at discovering pairs from different sectors, potentially offering more diverse trading opportunities.

4.3 Results on Trade Execution

4.3.1 LSTM Improves Performance Metrics and Adaptability in Spread Prediction

In our analysis, we utilized a Long Short-Term Memory (LSTM) network, a type of recurrent neural network (RNN) particularly well-suited for time series forecasting due to its ability to capture long-term dependencies and patterns in sequential data. LSTM was the ideal choice for predicting spreads, specifically the zdiff5, zdiff10, and zdiff20, as it can effectively learn and remember past behaviors in stock prices. After predicting the spreads using LSTM, we executed the trading strategy on the HDBSCAN + Correlation qualified pairs, which initially showed the highest Sharpe Ratio among the clustering techniques.

Metric	Predicted		Current		Improvement
	Mean	Deviation (std)	Mean	Deviation (std)	Mean
Maximum Drawdown	-4.7%	0.03	-5.3%	0.02	88.6%
Percentage of Time in Position	29%	0.71	35%	0.08	83.2%
Sharpe Ratio	0.64	0.79	0.57	0.61	112.4%
Sortino Ratio	0.89	1.38	0.75	1.06	118.1%

General Observations:

- Performance Improvement: The LSTM-based approach has led to improved performance metrics (lower drawdown, higher Sharpe and Sortino ratios), validating the effectiveness of LSTM in predicting spreads and making trading decisions.
- Variability: The predicted strategy shows slightly higher standard deviations across all metrics, suggesting that while the LSTM model improves average performance, it introduces more variability. This could be due to the model's sensitivity to market conditions and its dynamic nature. The higher standard deviation in 'percentage of time in position' for the LSTM-based strategy is actually a positive indicator, as it reflects the model's ability to adapt more dynamically to changing market conditions, allowing for smarter, more flexible trading decisions that can capitalize on short-term opportunities while avoiding prolonged exposure to unfavorable market conditions.

5 Conclusion

5.1 Summary of Key Findings

1. **Efficient Pair Selection:** By significantly reducing the number of pairs subjected to the Dickey-Fuller Test, we streamlined our process, enhancing efficiency without compromising the identification of stationary pairs.
2. **Identification of Stationary Pairs:** Our approach successfully identifies stationary pairs, validating the effectiveness of our methodology in finding pairs suitable for trading.
3. **Enhanced Performance:** The performance of our selected pairs is equal to or better than those identified using traditional methods (homework pairs), demonstrating the robustness of our approach.
4. **Superior Out-of-Sector Pair Identification:** Cointegration proves to be more effective than correlation in identifying out-of-sector pairs, offering more diverse and potentially profitable trading opportunities.
5. **Optimal Test and Clustering Combination:** While cointegration is traditionally favored, our findings indicate that correlation tests yield the best results in combination with clustering techniques, suggesting a tailored approach works best for our model.
6. **Model Efficacy:** Our predicted model shows improved performance metrics, with a decrease in drawdown and increases in both Sharpe and Sortino ratios, confirming its effectiveness in enhancing risk-adjusted returns.

5.3 Future Work

This study opens several avenues for further research in enhancing pairs trading strategies using machine learning. Future work could focus on improvements in the three key areas of pairs trading: pair discovery, pair testing, and trade execution.

In pair discovery, incorporating more unstructured data, such as news sentiment or social media data, could provide additional insights into the relationships between stocks and potentially uncover new profitable pairs. Furthermore, tuning the hyperparameters of clustering algorithms, such as the epsilon and minimum points for DBSCAN or the minimum cluster size and minimum samples for HDBSCAN, could lead to more optimal pair groupings and improve the overall performance of the strategy. Additionally, expanding the scope of analysis to include other asset types such as commodities, currencies, or bonds, and applying transfer learning techniques to leverage insights gained from the S&P stock analysis to other indices or markets could provide further diversification and improve the adaptability of the model.

For pair testing, exploring alternative approaches beyond cointegration and correlation tests could provide a more comprehensive assessment of pair suitability. Measurements such as the Hurst exponent, which measures the long-term memory of a time series, or analyzing the frequency of mean-reversion could offer additional insights into the stability and profitability of potential pairs. Developing a dynamic metric to rank pairs within the portfolio based on their predicted profitability and risk metrics could also allow for a more responsive and optimized trading strategy.

In the trade execution phase, there is significant room for improvement in optimizing the signal and stop-loss thresholds. Instead of relying on hardcoded values, future research could involve developing a more dynamic and data-driven approach to setting these thresholds. This could involve using Bayesian optimization to learn optimal thresholds based on historical data or adapting the thresholds in real-time based on market conditions. Incorporating a pairs' risk exposure to different sectors into the trade execution model could help align the strategy more closely with industry practices and manage sector-specific risks. This could involve setting sector-specific thresholds or dynamically adjusting position sizes based on the pairs' sector exposures.

To enhance the spread prediction model, future work could focus on expanding the feature set to include macroeconomic indicators, sentiment analysis from news articles, sector-specific trends, and technical indicators. Utilizing higher frequency data, exploring categorical labels (binary or quintiles), or incorporating more machine learning models such as GRU (Gated Recurrent Units) or ARIMA could also potentially improve the predictive power of the model.

Finally, implementing real-time data feeds to continuously update the model and make trading decisions could provide a more agile and timely response to market movements, improving the efficacy of the trading strategy.

By addressing these aspects, future research could further refine and enhance the performance of machine learning-based pairs trading strategies, potentially leading to improved profitability and risk management. The integration of unstructured data, optimization of clustering hyperparameters, exploration of alternative pair testing techniques, dynamic threshold setting, sector-specific risk management, and the application of advanced machine learning techniques could all contribute to the development of more sophisticated and effective pairs trading strategies.

6 References

1. Vashi, A. Pair trade: Buy oil (USO), short oil stocks (XLE). Retrieved from https://seekingalpha.com/article/4561954-pair-trade-buy-oil-uso-short-oil-stocks-xle

2. Najera, C. (2019, December 11). Pairs trading suggested for energy stocks. Retrieved from https://caia.org/blog/2019/12/11/pairs-trading-suggested-for-energy-stocks

3. Najera, C. Pairs trading with commodities: Evidence from the energy sector. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3458559

5. Mota, M. (2019, December 2). Employing machine learning for trading pairs selection. Retrieved from https://medium.com/aimonks/pairs-trading-using-machine-learning-for-the-selection-of-pairs-24920cbed1b6

6. Debrincat, A. (n.d.). Employing machine learning for trading pairs selection. Retrieved from https://hudsonthames.org/employing-machine-learning-for-trading-pairs-selection/

7. Kim, D. (n.d.). Pair trading. Retrieved from https://daehkim.github.io/pair-trading/

8. Khandelwa, S (2020, March 25). Predicting stock prices with LSTM and GRU: A step-by-step guide. https://blog.gopenai.com/predicting-stock-prices-with-lstm-and-gru-a-step-by-step-guide-381ec1554edf