Interfaces with Other Disciplines

Pairs trading via unsupervised learning

Chulwoo Han [a,*], Zhaodong He [b], Alenson Jun Wei Toh [c]

[a] *Sungkyunkwan University, 25-2 Sungkyunkwan-ro, Jongno-gu, Seoul, 03063, KOREA*
[b] *Durham University, Mill Hill Lane, Durham, DH1 3LB, UK*
[c] *Nanyang Technological University, 50 Nanyang Ave, 639798, Singapore*

ARTICLE INFO

ABSTRACT

This paper develops a pairs trading strategy via unsupervised learning. Unlike conventional pairs trading strategies that identify pairs based on return time series, we identify pairs by incorporating firm characteristics as well as price information. Firm characteristics are revealed to provide important information for pair identification and significantly improve the performance of the pairs trading strategy. Applied to the US stock market from January 1980 to December 2020, the long-short portfolio constructed via the agglomerative clustering earns a statistically significant annualized mean return of 24.8% and a Sharpe ratio of 2.69. The strategy remains profitable after accounting for transaction costs and removing stocks below 20% NYSE-size quantile. A host of robustness tests confirm that the results are not driven by data snooping.

1. Introduction

1.1. Motivation

Machine learning has gained momentum and been widely applied in various fields, including finance, in recent years. It is estimated that 56% of hedge funds utilized machine learning in their investment decisions in 2018, as compared to only 20% in 2017 (Chahn, 2018). However, most of these applications are supervised learning-based, where they aim to predict returns, or reinforcement learning-based, which are used for selecting stocks for portfolio construction. In contrast, unsupervised learning is rarely used in the finance literature.

Unsupervised learning, also known as clustering, classifies a set of data into a group if they exhibit more significant similarity to one another, as compared to data that are classified into other groups. Different clustering methods can give different clustering results with the same dataset due to the difference in their clustering algorithms, which are discussed in greater detail in Section 3. Unsupervised learning has several applications in finance, such as industry classification and portfolio construction. In this paper, we explore the feasibility of unsupervised learning for pairs trading. Stocks with similar characteristics can be classified into a cluster via unsupervised learning. These stocks are likely to have similar

future price movement, and one can find arbitrage opportunities by shorting overpriced stocks and longing underpriced stocks within the same cluster.

Existing pairs trading strategies typically identify pairs based on cointegration or some other time series related measurements (Gatev, Goetzmann, & Rouwenhorst, 2006; Huck, 2009; 2010; Jacobs & Weber, 2015; Rad, Low, & Faff, 2016; Vidyamurthy, 2004). However, using only past price data can falsely identify spuriously correlated pairs that may not move in tandem in the future. Unsupervised learning allows us to identify similar stocks considering not only past price movement but also other firm characteristics. Stocks with similar characteristics are believed to co-move with a higher probability.

1.2. Objectives

This research aims to investigate the feasibility of unsupervised learning for finding potential pairs of stocks to construct a long-short portfolio. We propose a framework to identify clusters from high dimensional financial data and implement three popular clustering algorithms to construct pair portfolios for statistical arbitrage. We choose k-means clustering, density-based spatial clustering of applications with noise (DBSCAN), and agglomerative clustering as representatives of each category of clustering methods.

The k-means clustering (MacQueen et al., 1967) is the pioneering and most popular clustering method, which assigns data points to various clusters by minimizing the within-cluster sum of

* Corresponding author.
E-mail addresses: chulwoo.han@skku.edu (C. Han), zhaodong.he@durham.ac.uk (Z. He), atoh009@e.ntu.edu.sg (A.J.W. Toh).

squares (WCSS) between the centroids of the assigned clusters and the data points.

DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) is a density-based clustering method, where clusters are identified based on a continuous region of high density and are separated from each other by a continuous region of low density. Unlike k-means clustering that assigns all data points to clusters, the data points in the low-density area are treated as outliers and are not assigned to any cluster.

The agglomerative clustering (Johnson, 1967) is one of the most representative hierarchical clustering methods. It is known as a "bottom-up" approach, where each data point is initially treated as a cluster on its own before being merged with other clusters as one moves up the hierarchy.

Our clustering-based pairs trading strategy consists of two steps. In the first step, stocks are clustered based on their past returns and characteristics. Stocks are grouped together if their prices move in tandem historically and have similar characteristics. We employ various firm characteristics as they are believed to be forward-looking and reduce the chance of falsely identifying spuriously correlated pairs.

The second step involves setting up a trading rule based on the divergence of the past one-month returns. Low (high) return stocks within the same cluster are deemed undervalued (overvalued), and a contrarian strategy is adopted by going long on the undervalued stocks and short on the overvalued stocks. More specifically, we open long and short positions if the return difference is greater than one cross-sectional standard deviation of the difference of past one-month returns. The long-short portfolio is rebalanced and reinvested at the end of every month.

Pairs trading reduces the market risk and makes the trading strategy profitable with a minimal market beta. In practice, investors do not trade only one pair but trade multiple pairs to form a well-diversified long-short portfolio. Our strategy allows the investor to choose a desired number of stocks by adjusting the threshold for the return divergence.

Applying clustering for pairs trading entails several difficulties and unexpected issues to be resolved. Firstly, clusters are unbalanced in their size. Some clusters have a sufficient number of stocks to form a portfolio, whereas other clusters have only a minimal number of stocks. In extreme cases, there are clusters that consist of only two stocks. DBSCAN, in particular, tends to yield a single colossal cluster comprised of the majority of the stocks, while the remaining stocks are identified as outliers, which suggests that the high-dimensional data structure has one big cluster that is exceptionally dense, and the rest of the stocks are scattered elsewhere.

Another issue is that some clustering methods, such as k-means clustering, assign all the stocks into clusters without identifying and excluding outliers. Hence, some of these clusters may contain dispersed outliers that are geometrically far from their assigned group centroids. Forming portfolios with such outliers will diminish profitability.

To address these issues, we test the clustering methods varying the clustering parameters and examine the impact of the number of clusters on the strategy performance. We also propose a method to remove outliers when using the k-means clustering.

1.3. Main findings

We test our strategies on the stocks listed in the US stock market during the sample period of December 1979 to November 2020. The out-of-sample period is from January 1980 to December 2020 as we form a portfolio at the end of each month and hold it for one month.

Comparing the performance of the long-short portfolios constructed by the three clustering methods, we find that the agglomerative clustering significantly outperforms the others, attaining an annualised mean return of 24.8%, an annualised Sharpe ratio of 2.69, and a low maximum drawdown of 12.3%, which lasts only for two months. The k-means clustering and DBSCAN attain an annualised Sharpe ratio of 2.34 and 2.04, respectively, exhibiting the great potential of unsupervised learning for pairs trading. They all outperform the S&P500 index and the conventional short-term reversal strategy significantly.

Our strategies also discover a significant alpha. The agglomerative clustering-based portfolio produces an economically and statistically significant monthly alphas of 1.9%, when regressed on the Fama-French three factors, Fama-French three factors plus momentum and short-term reversal factors, Fama-French five factors, and q^5 factors.

When stocks are clustered using only price information, the performance diminishes considerably; the Sharpe ratio decreases from 2.34 to 1.76 for k-means clustering, 2.04 to 1.57 for DBSCAN, and 2.69 to 1.44 for agglomerative clustering, and the differences are statistically significant. This result supports our hypothesis that firm characteristics play a non-trivial role in identifying pairs.

We attempt to shed light on the black-box of the clustering methods and unveil sources of profitability. We find that all three clustering methods select stocks mainly from the manufacturing and financial sectors. This implies that the strategies are scalable as these are the sectors that contain the majority of the stocks in the market and they are usually larger firms. This also implies that firms in these sectors are more similar to each other within their respective sector and are more likely to move in tandem. Clusters are usually homogeneous in terms of industry, consisting of only one or few sectors. However, we also find that when a cluster contains stocks from multiple industry sectors, the long portfolio tends to consist of stocks from one sector, while the short portfolio tends to consist of stocks from another sector. This implies that there are two sources of profit: the profit resulting from the divergence of the stocks in the same industry and the profit resulting from the divergence of two sectors that tend to co-move.

The breakdown of the portfolios in size reveals that they choose more stocks from the upper half of the size deciles, which suggests that the profits are not driven by small-cap stocks.

Several robustness tests are conducted to ensure that the results are not sensitive to parameters and not driven by data snooping. The robustness check reveals that our models perform robustly in different parameter settings.

Excluding stocks below 20% NYSE-size quantile reduces the profitability of the strategies only slightly, suggesting that the profits are not driven by small-cap stocks, and the strategies are highly scalable for asset management firms and hedge funds that typically trade large positions.

The turnover of the strategies is high at about 180%, which means half of the positions are replaced when the portfolio is rebalanced. This is mainly due to the nature of the contrarian strategy. Adding transaction costs lowers the Sharpe ratio, but the financial performance is still promising. The agglomerative strategy attains an annualised Sharpe ratio of 1.73 and a mean return of 15.9% after subtracting 20 basis point transaction costs.

Our strategies perform exceptionally well during a financial crisis. All three strategies perform robustly and significantly outperform the benchmarks during the 2007 financial crisis and the 2020 market crash due to the COVID-19 pandemic.

1.4. Contributions

The contribution of this paper is twofold. Firstly, to our knowledge, this is one of the earliest attempts to apply unsupervised

learning to pairs trading, which extends the empirical asset pricing literature. We offer a framework to identify pairs via clustering methods and construct portfolios from them. Unsupervised learning appears to extract helpful information from high-noise financial data and identify meaningful pairs. The long-short portfolios derived from the clusters are more profitable and less volatile compared to those from conventional pairs trading strategies. This result suggests that stocks' behaviour and characteristics should be considered more thoroughly when selecting pairs for pairs trading, rather than relying on the statistical measures used in traditional pairs trading strategies.

Our study provides a big picture on how stock data in high-dimensional space are perceived through the eyes of the various clustering methods. The analysis of clustering characteristics shows that all clustering methods identify at least 60% of available stocks as outliers, implying more than half of the stocks are isolated and should not enter the pairs trading portfolios. For the remaining stocks, the *k*-means clustering and agglomerative clustering tend to form more than a hundred clusters of various sizes, whereas DBSCAN tends to form one enormous cluster that encompasses almost all the remaining stocks, and another tiny cluster containing less than ten stocks on average.

Secondly, our study reveals the importance of firm-level characteristics in identifying pairs. Unlike traditional pairs trading strategies, unsupervised learning can incorporate information from firm characteristics as well as price-related information. In our study, it is found that adding firm characteristics reduces volatility and significantly improves the performance of the pairs trading strategies. To the best of our knowledge, this paper is the first attempt to incorporate firm characteristics in pairs trading.

This paper is organised as follows. Section 2 reviews existing pairs trading strategies in the literature as well as the financial applications of machine learning. Section 3 describes the data and feature generation process. It also details the working principle of the clustering methods and the trading algorithm. Section 4 presents the empirical results, including clustering behaviour, financial performance, factor regression analysis, and sub-period analysis. Section 5 performs various robustness checks, and Section 6 concludes.

2. Literature review

This paper contributes to the literature on statistical arbitrage by proposing new ways to construct pairs trading strategies. More broadly, it extends the literature on financial applications of machine learning by employing unsupervised learning.

2.1. Existing pairs trading strategies

Krauss (2017) classifies pairs trading strategies into four groups. The most mainstream approach is the cointegration approach, which involves the Engle & Granger (1987) cointegration test. Using this framework, Rad et al. (2016) implement a pairs trading strategy with time-varying trading costs. The strategy yields a mean monthly excess return of 33 basis points after transaction costs from 1962 to 2014. Intuitively, they discover that their pairs trading strategy is more profitable during a period of high volatility. Huck & Afawubo (2015) compare the cointegration method with the distance approach on the constituents of S&P500 and demonstrate that the cointegration method is stable and can generate profits consistently. They also address the meagre excess return of the distance approach after factoring in transaction costs. Clegg & Krauss (2018) extract pairs from the S&P500 constituents via partial cointegration. After including a mean-reverting component in the cointegration residuals, the portfolio yields an annual-

ized return of more than 12% after transaction costs from 1990 to 2015.

The second approach is the distance approach, which identifies co-moving pairs based on a particular distance metric. Gatev et al. (2006) utilize the Euclidean squared distance between the normalized prices of securities to identify pairs and form a long-short portfolio. Positions are opened when the distance of a pair is greater than two standard deviations of the historical distances between the pair. The positions are then closed when either mean-reversion occurs, or at the end of the trading period. This paper provides the main ideas and building blocks of pairs trading. Do & Faff (2010) extend the study of Gatev et al. (2006) to post 2008 financial crisis and discover that the strategy performs strongly during the recession period. The same study finds that pairs trading is more profitable for bank stocks. Jacobs & Weber (2015) extend the distance-based pairs trading strategy to 34 countries and discover that it is profitable in different markets, despite its varying performance with time.

The next approach to pairs trading is the time-series approach. Cummins & Bucca (2012) adopt a novel time-series model to identify pairs in the oil market, with a focus on West Texas Intermediate (WTI), Brent, heating oil, and gas oil from 2003 to 2010. Their pairs trading strategy yields a mean daily return between 0.07% and 0.55% and a Sharpe ratio greater than 2, prior to transaction costs. However, contrary to other pairs trading strategies that are usually more profitable during the 2008 financial crisis, the returns generated from this time-series model diminish during the recession. Elliott, Van Der Hoek, & Malcolm (2005) propose another time-series approach for pairs trading based on the mean-reverting Gaussian Markov chain, but no backtesting is conducted on any market.

The most novel and relatable approach to this study is to construct pairs via supervised learning. Huck (2009, 2010) implement an ensemble method that combines an Elman neural networks, a recurrent neural network, and ELECTRE III to identify pairs from the S&P500 constituent stocks.[1] When backtested from 1992 to 2006, the portfolio yields a weekly excess return of more than 0.8%. Krauss, Do, & Huck (2017) convert pairs trading into a supervised learning problem by feeding several features into machine learning algorithms (deep neural networks, gradient-boosted-trees, and random forests). They forecast the probability of stock prices' upward moving and construct a daily rebalanced portfolio by entering long positions in the stocks with the highest probabilities and short positions in the stocks with the lowest probabilities. From 1992 to 2015, an equally-weighted portfolio that ensembles the three machine learning methods yields a daily excess return of 0.45% prior to transaction costs. Huck (2019) adds more than 600 predictors to his machine learning models hoping that the learning algorithms can digest a large dataset and extract useful information from the extremely noisy data. Besides lagged returns, the feature set consists of time information, size of stocks, indices, commodity prices, and many frequently used risk factors. However, the positive excess return is negated by transaction costs, which implies that adding too many features may not boost the performance of machine learning-based pairs trading. Kim & Kim (2019) applly deep reinforcement learning on pairs trading, to determine the trading threshold and the stop-loss boundary dynamically and report promising results in comparison to a static boundary.

There are other non-conventional methods to identify pairs for pairs trading strategies, which are less frequently used in the literature. Avellaneda & Lee (2010) utilize the principal component analysis (PCA) on the exchange-traded funds (ETFs) listed in the US

---

[1] ELECTRE III is an outranking method based on the concept of fuzzy logic.

stock market and attain an annualized Sharpe ratio of 1.51 from 2003 to 2007. Rad et al. (2016) construct pairs based on a Copula method and gain a monthly excess return of 5 basis points after transaction costs, while Xie, Liew, Wu, & Zou (2016) demonstrate that the Copula method outperforms the distance approach in terms of average excess returns generated.

2.2. Financial applications of unsupervised learning

Financial applications of unsupervised learning are relatively rare. For the distance-based clustering method, Nanda, Mahanty, & Tiwari (2010) develop a portfolio strategy using k-means clustering, followed by the Markowitz model to create a portfolio. Fung, Yu, & Lam (2002) use a modified k-means clustering to cluster news articles, with the goal of analyzing the impact of news articles on stock prices. Ta & Liu (2016) adopt k-means clustering as one of their clustering methods to cluster stocks listed on the Ho Chih Minh Stock Exchange and study the impact of foreign ownership on Vietnamese stocks' volatility.

Yang, Lian, Li, Chen, & Li (2014) adopt DBSCAN to identify suspicious financial transactions. DBSCAN is also one of the three unsupervised clustering methods explored by Pavlidis, Plagianakos, Tasoulis, & Vrahatis (2006) to cluster time series data before applying artificial neural networks to forecast daily foreign exchange spot rates. Bini & Mathew (2016) also use DBSCAN as one of their clustering algorithms to cluster stock market data before applying regression models to forecast prices of the stocks listed on the National Stock Exchange of India.

Dose & Cincotti (2005) apply complete-link agglomerative clustering to the daily closing prices of the S&P500 constituents to create an index and an enhanced index tracking portfolio. Babu, Geethanjali, & Satyanarayana (2012) apply agglomerative clustering on financial data before applying a recursive k-means clustering within the agglomerative clusters to predict the direction of stock prices.

2.3. Financial applications of supervised learning

The volume of the literature on the financial application of machine learning has been blooming in recent years. Exceptional financial performances are reported when machine learning is applied to asset return prediction, which can be easily defined as a supervised learning problem.

Kaucic (2010) proposes an evolutionary learning method to find the best combination of almost 5000 technical indicators. Three learning methods; the Plurality Voting Committee, the Bayesian Model Averaging, and the Boosting method, are employed to transform the raw indicators into a unique trading signal. The proposed trading strategy demonstrates a good predicting accuracy during the period from 2000 to 2004. Using a classification tree, Moritz & Zimmermann (2016) develop a two-step portfolio sorting method and report significant financial performance. Messmer (2017) obtains significant excess returns by constructing a deep learning model with 68 firm characteristics as input features. Fischer & Krauss (2018) apply a long short-term memory (LSTM) network to form a daily-based long-short portfolio from the S&P500 constituents. The LSTM model gains an annualized Sharpe ratio of 5.8 prior to the transaction cost.

Paiva, Cardoso, Hanaoka, & Duarte (2019) employ a support vector machine to classify the stocks in the So Paulo Stock Exchange Index using technical indicators as input features. The out-of-sample test during the period from 2002 to 2016 finds the trading strategy significantly outperforms the market. Gu, Kelly, & Xiu (2020) conduct a comparative analysis of popular machine learning algorithms in the US stock market. They find that a deep neural network outperforms tree-based models and linear benchmarks, and the long-short portfolio obtained from the deep neural network achieves an annualized Sharpe ratio above 2.0. Barucci, Bonollo, Poli, & Rroji (2021) employ a Naive-Bayes classification algorithm to build a trading strategy that receives as input indicators defined through outliers in the time series of stocks (return, volume, volatility, bid-ask spread). They apply the methodology to the EURO-STOXX50 and the DOW JONES indices and find that it yields a higher risk-adjusted return than the buy-and-hold strategy. However, the outperformance vanishes even under a mild transaction cost assumption. Han (2021) employs a deep neural network to predict the cross-section of stock returns in the US market and finds that the long-short portfolio strategy derived from the model can earn a Shape ratio greater than 2.5.

Machine learning is also found to perform well in other asset classes. Lim, Zohren, & Roberts (2019) employ a deep neural network for time-series momentum and show that it performs robustly in the futures market. Bianchi, Büchner, & Tamoni (2020) apply the methodology of Gu et al. (2020) to the US Treasury bond market and find machine learning-based strategies are profitable also in the bond market.

3. Methodology

3.1. Data description and prepossessing

This section describes the procedure to generate and process the input features. The feature set consists of 48 return factors and 78 firm characteristics generated every month for the sample period from December 1979 to November 2020. The out-of-sample period is from January 1980 to December 2020.

3.1.1. Data sample

The US equity market data available from the Center for Research in Security Prices (CRSP) are used for our empirical studies. All stocks with common shares (share code 10 or 11) listed on the New York Stock Exchange (NYSE), American Stock Exchange (AMEX), and Nasdaq (exchange code 1, 2, and 3) are included. During the sample period, stocks must have a valid price on the portfolio formation date (at the end of the month t) and returns from $t - 48$ to t. For stocks delisted from the exchanges, the delisted returns are computed using the method of Shumway & Warther (1999): should the delisted return with dividend is available in the CRSP delist file, it is used; otherwise, the delisted return is assumed to be -35% for NYSE and American stock exchange stocks, and -55% for Nasdaq stocks.

3.1.2. Feature generation

The feature set consists of two parts. The first set comprises 48 return factors, i.e., 1- to 48-month past returns. The other set consists of 78 firm characteristics chosen from Green, Hand, & Zhang (2017). Stocks are clustered using the features available at the end of month $t - 1$ and the previous one-month return is used to identify undervalued and overvalued stocks.

The return features reflect the historical movement of stock prices and stocks with similar return features are expected to keep the similar pace in the future. Firm characteristics can reveal more information from accounting and asset pricing perspectives. A pair of stocks that have moved in tandem may diverge from each other in the future should they have distinct firm characteristics: we consider firm characteristics to be more forward-looking. This motivates us to use both return features and firm characteristics for clustering. A pairs trading strategy typically uses only prices or returns, and this is the first work that employs firm characteristics to identify pairs.

Return features The i-month return feature at the end of month $t - 1$ is defined as the cumulative return from month $t - i$ to $t - 2$

for $i > 1$ and as the previous one-month return for $i = 1$:

$$mom_i = r_{t-1}, \quad i = 1,$$

$$mom_i = \prod_{j=t-i}^{t-2} (r_j + 1) - 1, \quad i \in 2, \ldots, 48, \tag{1}$$

where r_j denotes the return in month j.

Firm characteristics

The 78 firm characteristics are chosen from the firm characteristics used in Green et al. (2017), excluding the characteristics with missing values of more than 15%. They are listed in Table Appendix A, and their exact definitions can be found in Green et al. (2017) or the references therein. The firm characteristics are generated using the SAS code in Jeremiah Green's website.[2]

3.1.3. Data preprocessing

Scaling data and performing PCA prior to unsupervised clustering is common in various clustering applications (Chen, Lach, Amft, Altini, & Penders, 2013; Sarmento & Horta, 2020; Siraj, Maarof, & Hashim, 2009). We first normalize features using their cross-sectional means and standard deviations to eliminate any skewed clustering. Unsupervised clustering methods usually cluster data points based on a certain distance measure and features with a higher magnitude, such as market capitalization, will have higher weightage in distance calculation without normalization. This would result in a skewed clustering, where stocks are clustered chiefly based on those features that have a higher magnitude by nature.

Some features must contain more information than others for the identification of similar stocks, but the distance measures assign the same weight to all features. Therefore, inputting the features directly into a clustering algorithm can dilute the information contained in important features. By applying PCA to the features and using the principal components as the input, we can emphasize those important features when clustering the stocks. Another benefit of PCA is to alleviate the curse of dimensionality. Clustering algorithms are resource-intensive, and reducing the feature dimension via PCA can save the computational cost significantly.[3]

We apply PCA on the normalized feature set on each portfolio formation date and choose for clustering the principal components that explain at least 99% of the total variation. Note that the PCA is applied to the cross-sectional data and only the features in month t is used for PCA in the same month. On average, 86 principal components out of 126 are selected as input.

3.2. Clustering methods

This section provides an in-depth discussion of the clustering algorithms we employ: k-means clustering, DBSCAN, and agglomerative clustering, which are representatives of partition-based, density-based, and hierarchical clustering, respectively. Clustering algorithms are unsupervised, which means they do not have the true values (labels) to target. Therefore, they do not require a large set of historical data to train and validate the model. By feeding the input data, clustering algorithms find meaningful groupings that cannot be conceived easily by the human brain.

Unlike supervised learning models, *e.g.*, deep learning or gradient boosting, which often require many hyperparameters, clustering algorithms require very few hyperparameters and there is little room for data snooping. We test the models with different hyperparameter values and assess their impact on the performance of the pairs trading strategy.

---

[2] https://sites.google.com/site/jeremiahrgreenacctg/home

[3] For example, the time complexity of k-means clustering for each iteration is $O(KNM)$, where K is the number of clusters, N is the number of samples, and M is the feature dimension.

3.2.1. k-Means Clustering

The k-means clustering (MacQueen et al., 1967) is perhaps the most popular clustering method. It requires the number of clusters, K, to be first specified. It then identifies K centroids and groups all data points to one of these clusters by minimizing the within-cluster sum of squares (WCSS) between data points and their respective centroids. The objective function is given by

$$W = WCSS = \sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik} \left\| x^i - \mu_k \right\|^2, \tag{2}$$

where x^i refers to ith data point, μ_k the centroid of cluster k, $w_{ik} = 1$ if x^i belongs to cluster k, otherwise $w_{ik} = 0$, and N is the total number of data points. $\|\cdot\|$ denotes l_2 norm.

The k-means clustering is a two-step minimization problem. W is first minimized with respect to w_{ik} while keeping μ_k fixed to update the assignment of data points to clusters (Eq. (3)). After that, W is minimized with respect to μ_k while keeping w_{ik} fixed to recompute the centroids (Eq. (4)). The above steps are repeated until W is minimized.

$$w_{ik} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \left\| x^i - \mu_j \right\|_2 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\frac{\partial W}{\partial \mu_k} = 2 \sum_{i=1}^{N} w_{ik} (x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{N} w_{ik} x^i}{\sum_{i=1}^{N} w_{ik}} \tag{4}$$

As k-means clustering assigns all data points into the specified number of clusters, it does not have the capability to identify outliers. Stocks that are located far from others are still assigned to the nearest centroid. Hence, we adopt and modify the method used by Hautamäki, Cherednichenko, Kärkkäinen, Kinnunen, & Fränti (2005) to identify these stocks and remove them from the asset pool.

The procedure of removing outliers is as follows. For each data point, measure the distance to its respective centroid and also the distance to its nearest neighbour. Sort the distances to the nearest neighbour in ascending order and choose the distance at α percentile as the threshold, ϵ. Any stock whose distance to its centroid is greater than ϵ is regarded as an outlier and removed. The pseudo-code for this process is described in Algorithm 2 in the appendix. We set $\alpha = 0.5$, *i.e.*, identify a stock as an outlier when its distance to its centroid is greater than the median distance between nearest stocks.

For the number of clusters, $K = 5, 10, 50, 100, 500, 1000,$ and 1500 are tested.

3.2.2. DBSCAN

The density-based spatial clustering of applications with noise (DBSCAN) identifies areas of high density in a high dimensional data space separated by regions of low-density (Ester et al., 1996). It has two parameters; the minimum number of data points per cluster, *MinPts*, and the maximum distance between data points, ϵ, to be considered in the same cluster. Distance metrics such as Manhattan (l_1 norm), Euclidean (l_2 norm), or Minkowski distance can be used to measure the distance between data points.

DBSCAN first selects an arbitrary data point as a core point. Data points with at least *MinPts* neighbours within ϵ are considered for the core point. All neighbouring points within ϵ from a core point are clustered together with the core point, in a process called direct density reachable. Any neighbours of these neighbouring points within ϵ are also included in the same cluster. This process is called density reachable. Non-core points in a cluster are called border points, and points in the same cluster are said to be
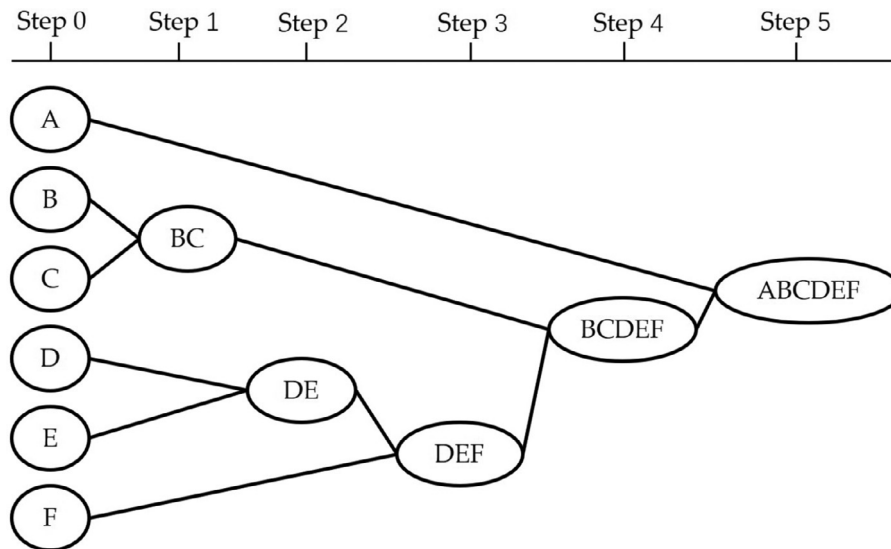
Fig. 1. Agglomerative clustering. This figure presents a six-point dataset's hierarchical structure, showing how the agglomerative clustering algorithm groups data points from individual data point to the final cluster that includes all the data points.

density connected. Outliers or noises are points that are not density reachable from any of the core points, and are not assigned to any cluster (Schubert, Sander, Ester, Kriegel, & Xu, 2017). The above process is repeated until all data points are visited. The pseudocode of DBSCAN is described in Algorithm 3.

Hyperparameter Selection

Following Aggarwal, Hinneburg, & Keim (2001), who demonstrate that l_1 norm provides more significant and meaningful clustering results for high-dimensional datasets, we choose l_1 norm as the distance metric for DBSCAN. Following Birant & Kut (2007), *MinPts* is set to be the natural logarithm of the total number of data points N. The minimum distance ϵ is set as the α percentile of the average distances to the nearest *MinPts* neighbour points. This method is recommended by Ester et al. (1996), who proposed DB-SCAN. Elbatta (2012) and Rahmah & Sitanggang (2016) also adopt the same method to determine ϵ. We use $\alpha = 0.1, \ldots, 0.9$ for the empirical analysis.

3.2.3. Agglomerative clustering

Agglomerative clustering is a hierarchical clustering method and starts by treating individual data points as a cluster on its own before merging the clusters step by step until termination criteria are met (Gowda & Ravi, 1995). It requires the user to specify one of two hyperparameters: the number of clusters K or the maximum distance for clusters to be merged, ϵ, known as linkage distance.

Fig. 1 illustrates the agglomerative clustering process. The six individual data points are initially treated as individual clusters. Their proximity to each other is calculated, and similar clusters are merged to form a new cluster (BC and DE), should their distance is less than ϵ. This merging process is repeated until there are no more clusters whose distance to another cluster is less than ϵ. If the distance between merged clusters is always below the specified ϵ, eventually, all cluster are merged to form cluster ABCDEF. In general, the final clustering result consists of many small clusters, rather than a few giant clusters that contain most data points.

Hyperparameter Selection

In our empirical study, we specify the maximum distance ϵ rather than the number of clusters K, using a method similar to the method adopted for *k*-means clustering: ϵ is set as an α percentile of the distances between a pair of nearest data points. We implement agglomerative clustering with $\alpha = 0.1, \ldots, 0.9$. Similarly to DBSCAN, l_1 norm is used as the distance metric. For the linkage

distance, *i.e.*, the distance between a pair of clusters, we adopt the average linkage, which is defined as the average distance between the data points in one cluster and the data points in another cluster. For instance, if there are two clusters G and H with N_G and N_H data points, respectively, the average linkage distance between G and H is defined as follows:

$$d_{\text{average}}(G, H) = \frac{1}{N_G \cdot N_H} \sum_{x^i \in G, x^j \in H} ||x^i - x^j||. \tag{5}$$

3.3. Portfolio formation and trading strategy

At the end of each month during the sample period, we cluster the stocks using one of the clustering methods described in the previous section. Through clustering, stocks are either assigned to a cluster or identified as an outlier.

As we cluster stocks based on their price movements and firm characteristics up to month $t - 1$, we expect the stocks in the same cluster that temporarily diverge from each other in month t to converge in the following month. Therefore, we detect pairs by examining the previous one-month return (mom_1). Stocks with a lower mom_1 are deemed undervalued and expected to rebound in the following month. Conversely, stocks with a higher mom_1 are deemed overvalued and expected to revert in the following month.

We sort the stocks within each cluster based on mom_1 and pair the highest mom_1 stock with the lowest mom_1 stock, the second-highest mom_1 stock with the second-lowest mom_1 stock, and so forth. We then construct an equally-weighted long-short portfolio using the pairs whose mom_1 difference is greater than the cross-sectional standard deviation of all pairs' mom_1 differences. It is possible for some clusters not to have any pairs that meet the trading criterion if the mom_1 differences of their pairs are not wide enough. We hold the positions for one month and rebalance the portfolio at the end of each month. This cycle repeats until the end of the sample period.[4]

The clustering-based long-short portfolios are compared with two benchmarks; the S&P500 index and the short-term reversal

---

[4] While pairs trading strategies often utilize daily returns, we use one-month holding period because firm characteristics are updated monthly or at a lower frequency (quarterly or annually). Given the high turnover of our strategy, it is also likely that rebalancing the portfolio at a higher frequency will incur higher transaction costs.

Table 1

Clustering characteristics. This table reports the clustering results from the *k*-means clustering, DBSCAN, and agglomerative clustering. Stocks are clustered every month during the test period from 1990.01 to 2020.12, and the reported values are time-series averages. The values in parentheses are the percentages against the total number of stocks.

	k-means	DBSCAN	Agglomerative
Number of clusters	129	2	188
Number of stocks in total	3157	3157	3157
Number of stocks in clusters	1257 (39.78)	376 (12.05)	769 (24.58)
Number of outliers	1900 (60.22)	2781 (87.95)	2388 (75.42)
Number of stocks in the biggest cluster	63 (5.18)	336 (88.89)	107 (13.72)
Number of stocks in the second biggest cluster	51 (4.13)	32 (8.74)	47 (6.09)
Number of stocks in the third biggest cluster	45 (3.63)	6 (1.88)	29 (3.74)
Number of stocks traded	330 (10.33)	75 (2.47)	126 (4.05)

portfolio. We use the S&P500 index including dividends available from CRSP. For the short-term reversal portfolio, we sort all the stocks based on their mom_1, and initiate long positions on the stocks in the first decile and short positions on the stocks in the bottom decile.

4. Empirical results

This section presents the empirical results. We first summarize the clustering characteristics of each clustering method and then assess the financial performances of the pairs trading strategies from various perspectives. A host of robustness tests are conducted to ensure the findings are not driven by data snooping.

We test each clustering method with different parameter values and use the best performing case for the main results and later analyze the impact of the parameters on the strategy performance. For *k*-means clustering, the number of clusters $K = 5, 10, 50, 100, 500, 1000$, and 1500 are tested and $K = 500$ is chosen for the main results as it gives the highest Sharpe ratio. For DBSCAN and agglomerative clustering, $\alpha = 0.1, \ldots, 0.9$ are tested, and $\alpha = 0.1$ is chosen for DBSCAN and $\alpha = 0.3$ for agglomerative clustering for the same reason.

4.1. Clustering characteristics

To give a big picture of how stocks are clustered, we analyze the clustering characteristics of the three clustering methods. The number of clusters varies significantly across the clustering methods. Table 1 and Fig. 2 summarize the results.

The *k*-means clustering forms an average of 129 clusters after removing outliers, with a range of 80 to 165 clusters.[5] The *k*-means clustering clusters average 1257 (39.8%) stocks every month and identifies the rest as outliers. Among the clustered stocks, average 63 (5.2%) stocks are contained in the largest cluster, and 51 (4.1%) stocks are contained in the second largest cluster. Fig. 2 shows that the number of clusters formed every month is consistent over the sample period. After applying the one standard deviation threshold, an average of 330 stocks remain and are used to form a long-short portfolio.

In contrast to *k*-means clustering, DBSCAN tends to form one giant cluster and several small clusters, with an average of only two clusters every month. DBSCAN clusters only 376 (12.1%) stocks on average, and 88.0% of the stocks are deemed as outliers. The giant cluster comprises almost all the clustered stocks (88.9%), and the

remaining clustered stocks are assigned to one to six mini clusters. On average, the DBSCAN strategy trades 75 stocks every month. Compared to *k*-means and DBSCAN, agglomerative clustering tends to form many small clusters (average 188), which comprises an average of 769 (24.6%) stocks. Fig. 2 reveals that there are no giant clusters as observed in DBSCAN, and the number of clusters is rather consistent over the sample period. Agglomerative clustering trades average 126 (4.1%) stocks every month.

4.2. Strategy performance

Table 2 reports the performances of the long-short portfolios constructed via the three clustering methods. The equally-weighted long-short portfolios outperform the S&P500 index and the benchmark reversal portfolio in terms of various financial performance indicators, revealing the clustering methods' capability to form meaningful clusters and generate significant profits.

Table 2 (a) reports the monthly return statistics of the clustering-based pairs trading strategies. It shows that the returns of the long portfolio from all three clustering methods is economically and statistically significant with the *t*-statistic greater than 7. The returns of the long-short portfolio, albeit smaller in magnitude, is even more significant with the *t*-statistic greater than 12. In particular, the agglomerative clustering yields the highest and most significant returns: 2.3% ($t = 8.95$) for the long portfolio and 2.1% ($t = 16.12$) for the long-short portfolio.

The mean returns of the long-short portfolios are greater than that of S&P500 and comparable to the mean return of the reversal portfolio. However, the long-short portfolios have significantly lower standard deviations.

Fig. 3 presents the monthly return distributions in a box plot. It is apparent that all the clustering-based trading strategies have a narrower range of returns and exhibit smaller downside returns than the benchmarks, indicating that the clustering-based strategies are less volatile and less likely to experience drastic drops in returns.

Table 2 (b) reports annualized risk-return metrics. Five metrics are employed to assess the financial performance of the portfolios:

- Sharpe ratio: $\frac{Mean\ return}{standard\ deviation}$
- Sortino ratio: $\frac{Mean\ return}{Downside\ deviation}$
- Profit factor: $\frac{Gross\ profit}{Gross\ loss}$
- Maximum drawdown (MDD): the maximum observed loss from a peak to a trough over the portfolio formation period
- Calmar ratio: $\frac{Mean\ return}{MDD}$

The results suggest that agglomerative clustering is the best performing clustering algorithm among the three clustering methods. The long-short portfolio has a high annualized mean excess return of 24.8% and a remarkably low standard deviation of 9.2%, resulting in the highest annualized Sharpe ratio of 2.69. Other performance metrics also favour agglomerative clustering: *e.g.*, it yields a low maximum drawdown of mere 12.3%. During the 41 years of out-of-sample period, there is not a single unprofitable year for the agglomerative clustering-based strategy.

The long-short portfolio derived from *k*-means clustering also performs superbly, but has a slightly lower mean return (21.5%), compared to the one derived from agglomerative clustering, resulting in a Sharpe ratio of 2.34. Still, it significantly outperforms the benchmarks and suffers loss only in one year out of 41 years.

DBSCAN appears to be the worst-performing among the clustering methods. It has the highest mean return of 25.6%, but due to the relatively high standard deviation of 12.6%, it has the lowest Sharpe ratio. Other performance metrics also suggest that it is the least favourable method: *e.g.*, the maximum drawdown is 14.9%, higher than those of *k*-means and agglomerative clustering.

---

[5] The number of clusters, *K*, is set to 500, but the outlier removing process eliminates some clusters whose data points are far from each other and the remaining clusters are less than 200.

(a) *k*-means

(b) DBSCAN
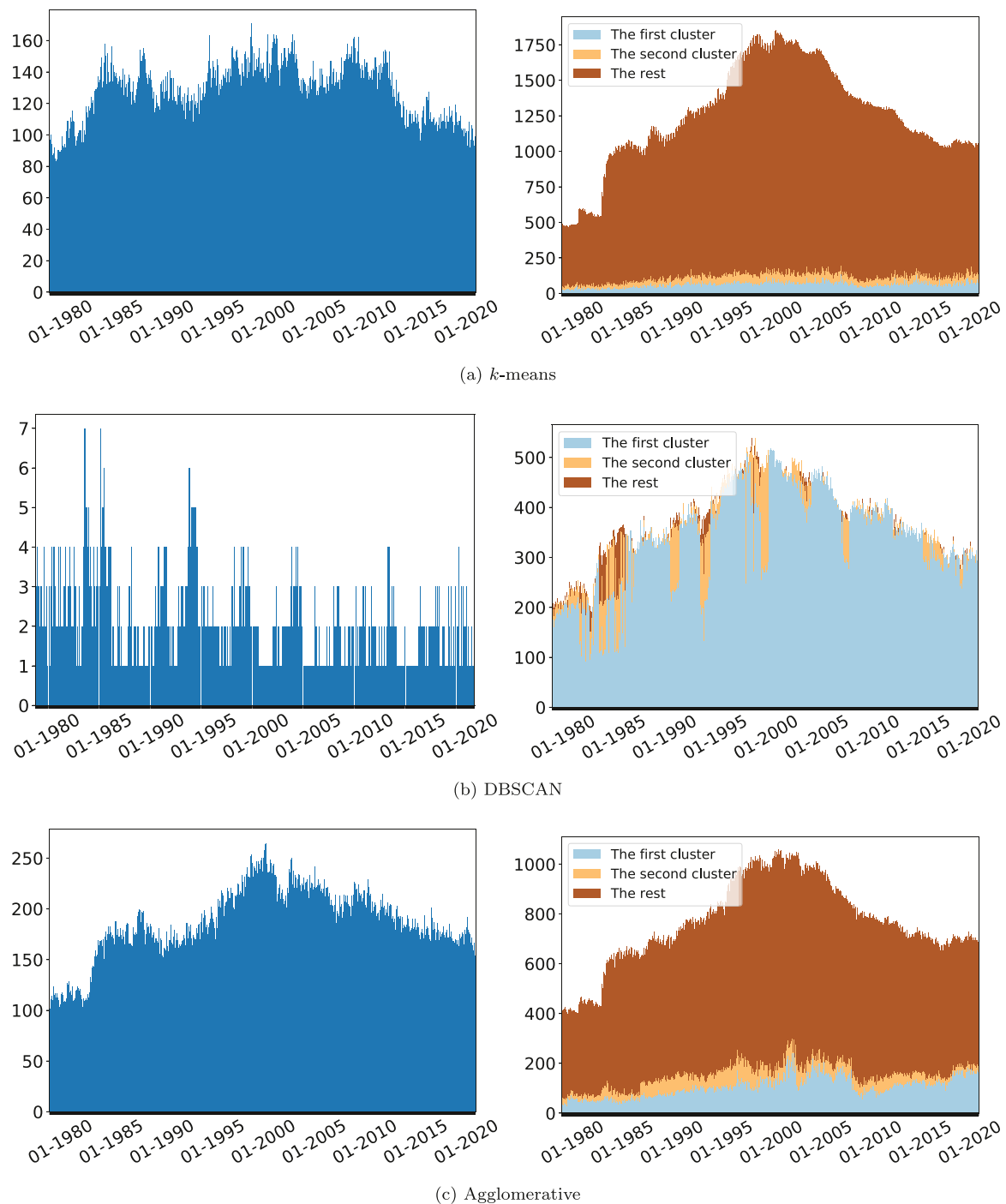
(c) Agglomerative

Fig. 2. Number of clusters and the number of stocks in the clusters. This figure presents the number of clusters and the number of stocks in the clusters formed by *k*-means clustering, DBSCAN, and agglomerative clustering. The graphs on the left show the number of clusters and the graphs on the right show the number of stocks in the first two biggest clusters every month during the test period from 1980.01 to 2020.12.

The reason that DBSCAN tends to form a giant cluster and perform rather poorly can be attributed to its nature of forming a cluster. DBSCAN assigns a datapoint to a cluster if it is within a distance from any of the data points in the cluster. This approach often results in an elongated cluster and the data points (stocks) at one end of the cluster can be significantly different from those at the other end of the cluster.

Nevertheless, it significantly outperforms the two benchmarks with an annualized Sharpe ratio of 2.04. An advantage of DBSCAN is that it requires on average only 75 stocks (38 pairs) to be traded every month, while the *k*-means and agglomerative clustering require more than 126 stocks to be traded every month.

It is notable that the long and the short legs of the pairs trading portfolios contribute to the performance more or less equally. Compared to the mean return of the S&P500, 12.6%, the long legs' returns are about 15% higher, whilst the short legs' returns are 10% lower. This implies that the strategy can be profitable even when short-sale is restricted. In contrast, many long-short portfolio strategies derive their profits from the short leg that consists of small stocks and are difficult to implement.

Table 2

Strategy performance. This table reports the performance of the equally-weighted pairs trading portfolios constructed via k-means, DBSCAN, and agglomerative clustering. The test period is from 1980.01 to 2020.12. Panel (a) reports the summary statistics of the monthly returns, and panel (b) reports annualized risk-return metrics. The t-statistic in panel (a) is the Newey-West t-statistic. The t-statistic in panel (b) tests the difference of the Sharpe ratio from the Sharpe ratio of the benchmark reversal portfolio. It is computed using the method of Ledoit & Wolf (2008). S&P500 denotes the S&P500 index with dividends and Reversal denotes the equally-weighted short-term reversal portfolio.

(a) Monthly return summary statistics

| | k-means | | | DBSCAN | | | Agglomerative | | | Benchmarks | |
	L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
Mean	0.023	0.005	0.018	0.022	0.001	0.021	0.023	0.002	0.021	0.011	0.020
Standard deviation	0.059	0.048	0.027	0.052	0.043	0.036	0.054	0.045	0.027	0.044	0.070
Standard error	0.003	0.002	0.001	0.002	0.002	0.002	0.003	0.002	0.001	0.002	0.003
t-statistic	7.905	2.056	14.280	8.999	0.267	12.842	8.951	1.116	16.115	5.275	6.362
Min	-0.285	-0.266	-0.079	-0.221	-0.291	-0.104	-0.240	-0.303	-0.087	-0.216	-0.363
25%	-0.007	-0.021	0.001	-0.006	-0.019	-0.001	-0.004	-0.018	0.004	-0.015	-0.011
50%	0.024	0.009	0.015	0.022	0.004	0.017	0.023	0.005	0.019	0.014	0.014
75%	0.052	0.034	0.031	0.052	0.024	0.038	0.050	0.028	0.035	0.037	0.042
Max	0.330	0.198	0.174	0.211	0.145	0.180	0.311	0.193	0.160	0.135	0.800
Skew	-0.226	-0.818	1.044	-0.320	-1.218	0.762	-0.108	-1.162	0.854	-0.597	3.135
Kurtosis	4.496	3.872	4.246	2.485	6.761	2.281	3.599	7.097	4.018	2.127	34.244

(b) Annualized risk-return metrics

| | k-means | | | DBSCAN | | | Agglomerative | | | Benchmarks | |
	L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
Mean return	0.272	0.056	0.215	0.263	0.007	0.256	0.277	0.029	0.248	0.126	0.239
Standard deviation	0.206	0.165	0.092	0.180	0.149	0.126	0.187	0.157	0.092	0.151	0.243
Sharpe ratio	1.318	0.339	2.336	1.457	0.044	2.039	1.481	0.183	2.692	0.835	0.983
t-statistic			9.036			5.365			10.070		
Downside deviation	0.158	0.133	0.046	0.131	0.127	0.063	0.139	0.134	0.048	0.112	0.141
Sortino ratio	1.719	0.423	4.713	2.007	0.052	4.065	1.998	0.215	5.150	1.123	1.693
Gross profit	17.164	9.890	10.165	15.979	7.642	12.832	16.470	8.460	11.422	11.088	15.548
Gross loss	-6.032	-7.588	-1.334	-5.203	-7.371	-2.326	-5.110	-7.283	-1.239	-5.910	-5.755
Profit factor	2.846	1.303	7.618	3.071	1.037	5.517	3.223	1.162	9.218	1.876	2.702
Profitable years	36	29	40	37	23	40	36	25	41	34	33
Unprofitable years	5	12	1	4	18	1	5	16	0	7	8
Maximum drawdown	-0.505	-0.580	-0.113	-0.481	-0.796	-0.149	-0.424	-0.595	-0.123	-0.502	-0.503
Calmar ratio	0.537	0.097	1.900	0.547	0.008	1.717	0.653	0.048	2.013	0.251	0.475
Turnover	0.900	0.923	1.823	0.929	0.951	1.880	0.939	0.955	1.894		
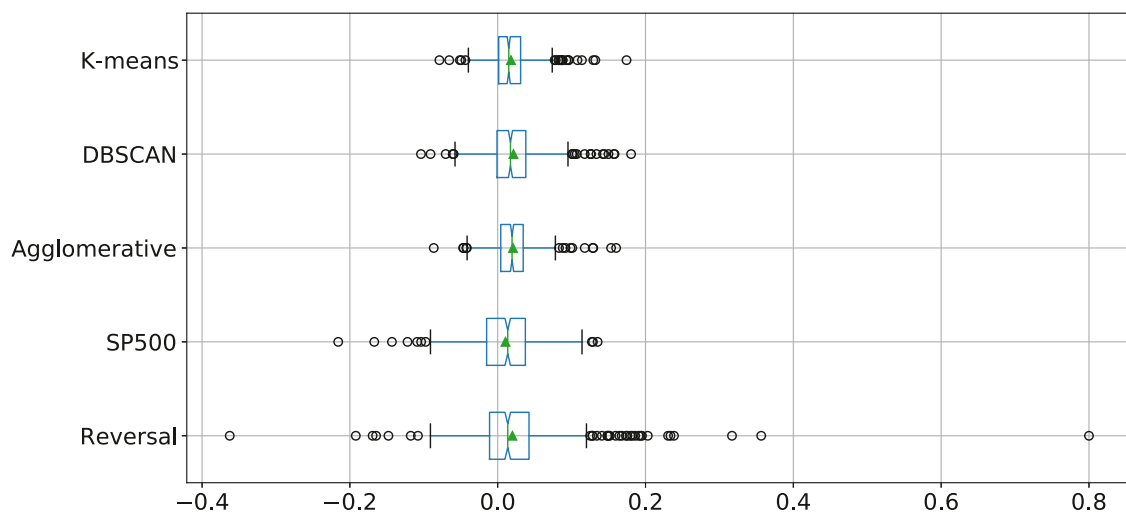
Fig. 3. Monthly return distribution. This box plot presents the monthly return distributions of the equally-weighted pairs trading portfolios constructed via k-means clustering, DBSCAN, and agglomerative clustering during the test period from 1980.01 to 2020.12.

Fig. 4 shows the log-scale cumulative returns of the strategies over the out-of-sample period. The graph reaffirms that the pairs trading strategy via agglomerative clustering performs best with the highest cumulative return. DBSCAN performs remarkably well after the financial crisis in 2008 and yields a comparable cumulative return. The reversal portfolio performs exceptionally well before 2000, but its performance diminishes considerably thereafter. Remarkably, the portfolios constructed by the clustering meth-

ods perform even better during the financial crisis in 2008, when S&P500 and the reversal portfolio suffer a considerable loss.

4.3. Effects of firm characteristics

We hypothesize that the firm characteristics are forward-looking and would help clustering stocks and enhance the pairs trading strategies. To test the hypothesis, we repeat the same back-
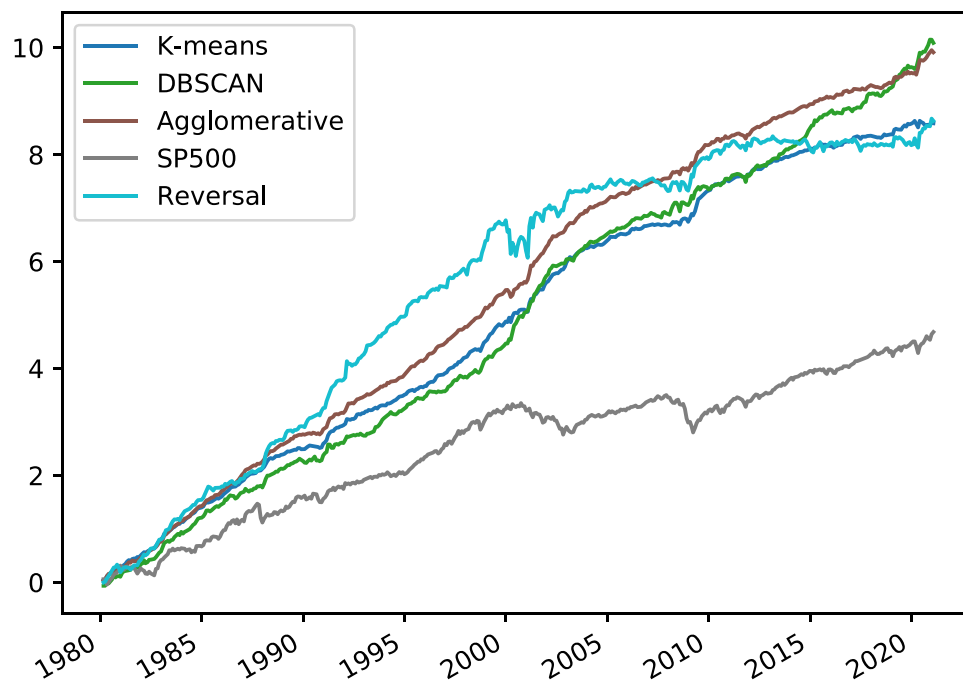
Fig. 4. Cumulative returns of pairs trading portfolios. This figure presents the log-scale cumulative returns of the equally-weighted pairs trading portfolios constructed via k-means clustering, DBSCAN, and agglomerative clustering during the test period from 1980.01 to 2020.12. The y-axis is the logarithm of $(1 + \text{cumulative return})$.

test using only the return features and compare the results with those reported earlier. Table 3 reports the results obtained from the clustering methods utilizing only the return features.

All the clustering methods perform worse after excluding the firm characteristics in terms of the Sharpe ratio and other risk-adjusted performance metrics. Interestingly, they yield a similar magnitude of returns, but the returns are more volatile, resulting in lower Sharpe ratios. The mean returns of the long-short portfolios derived from k-means, DBSCAN, and agglomerative clustering are respectively 24.4%, 23.8%, and 24.1%, whereas they are respectively 21.5%, 25.6% and 24.8% when the firm characteristics are included. In contrast, the Sharpe ratios are reduced from 2.34 to 1.76 (k-means), 2.04 to 1.57 (DBSCAN), and 2.69 to 1.44 (agglomerative), and the differences are statistically significant with the t-statistic greater than 3 except for DBSCAN, whose t-statistic is 2.88. The maximum drawdown also increases substantially when the firm characteristics are excluded: *e.g.*, the maximum drawdown of the agglomerative clustering increases from 12.3% to 50.3%.

The results above clearly suggest that the firm characteristics help identify similar stocks and enhance the performance of the pairs trading strategy. It is particularly notable that they reduce the volatility and downside risk substantially, making the strategy attractive to investors.

4.4. Factor regression

To examine whether systematic risk factors can explain the long-short portfolios' returns, we employ four factor models and apply them to the equally-weighted portfolios constructed via agglomerative clustering. We focus on agglomerative clustering as it demonstrates the most promising performance among the clustering algorithms. The factor models considered are Fama & French (1996)'s three-factor model (FF3), FF3 plus the momentum and the short-term reversal factors (FF3+2), Fama & French (2015)'s five-factor model, and Hou, Mo, Xue, & Zhang (2020)'s q^5 factor model. We include FF3+2 since the feature set includes momentum

features and the trading strategy is a contrarian strategy, which closely relates to the short-term reversal factor. The Fama-French factors are downloaded from Kenneth French's website, and the q^5 factors are downloaded from the global-q website. For the q^5 factor model, the sample period ends in December 2019 as the site provides the factors only up to 2019. Table 4 reports the regression results.

The long-short portfolio constructed from agglomerative clustering yields an economically and statistically significant monthly alpha of 1.9% ($t = 14.16$) when regressed against FF3. The market beta is also positive and significant, suggesting that the trading strategy is dollar-neutral but not risk-neutral. The factor loading on HML is positive and weakly significant, implying that it favours firms with a higher book-to-market ratio. The adjusted R^2 from FF3 is 0.08.

The FF3+2 appears to have the best explanatory power for the strategy with the adjusted R^2 of 0.28. Nevertheless, the strategy generates a significant alpha of 2.0% ($t = 15.01$), which is the most significant among the four factor models. A positive, significant loading on the market factor is observed again. In addition, the momentum and the short-term reversal factors are negative and positive, respectively, and statistically significant. The positive loading on the short-term reversal factor is expected as our strategy is essentially a short-term reversal strategy with a carefully selected trading asset pool.

The FF5 and q^5 models have a similar explanatory power to FF3, and the alpha remains economically and statistically significant. The profitability and investment factors do not appear to explain the returns of our strategy. The factors in q^5 are also unable to explain the returns, suggesting that the significant return of the strategy is not a result of excessive risk taking.

4.5. Size and industry breakdown of the clusters

To investigate what drives the clustering-based pairs trading strategies' returns, we categorise the portfolios into size deciles

Table 3

Strategy performance without firm characteristics. This table reports the performance of the equally-weighted pairs trading portfolios constructed via k-means, DBSCAN, and agglomerative clustering utilizing only return features. The test period is from 1980.01 to 2020.12. Panel (a) reports the summary statistics of the monthly returns, and panel (b) reports annualized risk-return metrics. The t-statistic in panel (a) is the Newey-West t-statistic. The t-statistic in panel (b) tests the difference of the Sharpe ratio from the corresponding Sharpe ratio in Table 2. It is computed using the method of Ledoit & Wolf (2008).

(a) Monthly return summary statistics

	k-means			DBSCAN			Agglomerative			Benchmarks	
	L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
Mean	0.024	0.003	0.020	0.024	0.004	0.020	0.025	0.005	0.020	0.011	0.020
Standard deviation	0.074	0.053	0.040	0.070	0.054	0.044	0.081	0.060	0.048	0.044	0.070
Standard error	0.004	0.003	0.002	0.003	0.003	0.002	0.004	0.003	0.002	0.002	0.003
t-statistic	6.657	1.291	10.610	7.297	1.688	9.454	6.341	1.575	8.736	5.275	6.362
Min	-0.359	-0.283	-0.159	-0.369	-0.464	-0.083	-0.332	-0.262	-0.229	-0.216	-0.363
25%	-0.015	-0.021	-0.001	-0.004	-0.015	-0.003	-0.013	-0.025	-0.003	-0.015	-0.011
50%	0.020	0.007	0.016	0.021	0.006	0.014	0.020	0.003	0.016	0.014	0.014
75%	0.057	0.030	0.034	0.047	0.025	0.037	0.056	0.030	0.037	0.037	0.042
Max	0.460	0.323	0.282	0.615	0.388	0.395	0.510	0.374	0.282	0.135	0.800
Skew	0.746	-0.119	1.237	1.840	-0.348	2.259	1.266	0.891	0.685	-0.597	3.135
Kurtosis	7.295	5.663	7.257	17.883	20.097	13.615	7.650	6.598	6.507	2.127	34.244

(b) Annualized risk-return metrics

	k-means			DBSCAN			Agglomerative			Benchmarks	
	L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
Mean return	0.283	0.039	0.244	0.291	0.053	0.238	0.296	0.054	0.241	0.126	0.239
Standard deviation	0.256	0.182	0.139	0.244	0.188	0.152	0.282	0.208	0.167	0.151	0.243
Sharpe ratio	1.105	0.214	1.757	1.195	0.281	1.571	1.049	0.260	1.442	0.835	0.983
t-statistic	2.520	2.158	3.640	2.296	2.215	2.875	3.684	0.236	5.434		
Downside deviation	0.168	0.138	0.081	0.171	0.162	0.066	0.171	0.136	0.118	0.112	0.141
Sortino ratio	1.688	0.283	2.997	1.703	0.327	3.591	1.733	0.399	2.045	1.123	1.693
Gross profit	18.956	9.884	12.507	17.544	9.051	12.733	19.923	10.972	13.550	11.088	15.548
Gross loss	-7.356	-8.281	-2.510	-5.603	-6.882	-2.960	-7.804	-8.749	-3.654	-5.910	-5.755
Profit factor	2.577	1.194	4.983	3.131	1.315	4.302	2.553	1.254	3.708	1.876	2.702
Profitable years	36	25	38	37	27	38	34	25	39	34	33
Unprofitable years	5	16	3	4	14	3	7	16	2	7	8
Maximum drawdown	-0.536	-0.656	-0.227	-0.477	-0.624	-0.129	-0.489	-0.642	-0.503	-0.502	-0.503
Calmar ratio	0.528	0.060	1.075	0.611	0.085	1.846	0.604	0.084	0.480	0.251	0.475
Turnover	0.857	0.932	1.789	0.932	0.981	1.913	0.908	0.967	1.875		

and industry sectors. The results are summarised in Table 5, where the numbers are the average percentage of the stocks in each category.

Table 5 (a) reports the average proportion of the stocks in each size decile. Decile 1 represents the smallest stocks, and decile 10 the largest stocks. All three clustering methods have a similar distribution of the stocks across size deciles and have the average size decile above 6.0. They select more than 10% of the stocks from each size decile above the median. This result corroborates the finding in Section 5 that the profits of the pairs trading strategies are not driven by small-cap stocks.

Table 5 (b) categorises all stocks into eleven industry sectors using the North American Industry Classification System (NAICS).[6] The last column (Market) represents the proportions of all the stocks in the US market.

The k-means and agglomerative clustering have a similar distribution of the stocks across industries and select stocks mainly from the manufacturing sector, followed by the financial sector. Their industry composition is also similar to the industry composition of the entire market. DBSCAN, on the other hand, does not follow the distribution of the market, but prefers stocks from the financial sector, followed by the manufacturing sector. Having many stocks from the financial sector is in line with Do & Faff (2010), who find that bank stocks are the most profitable when it comes to pairs trading.

The above results imply that the stocks in these industries are more similar to each other within their respective industry

in terms of price movement and firm characteristics. Should any stock in these industries be overpriced or underpriced relative to the others in the same industry, its return is likely to revert back to the mean level offering a profit opportunity. Involving stocks mainly from these industries also explains why our clustering-based strategies remain profitable after excluding small-cap stocks: manufacturing and financial firms are usually large in market capitalisation.

The number of industries within each cluster reveals that the clusters are homogeneous in terms of industry, consisting of stocks from only one or a few industries. The long-short portfolios constructed from k-means, DBSCAN and agglomerative clustering respectively consist of 1.83, 3.47, and 1.32 industries, on average. The long and short portfolios individually consist of fewer industries: average 1.32 industries for k-means clustering, 2.74 for DBSCAN, and 1.03 for agglomerative clustering. The smaller number of industries within each side indicates that the stocks in the long side and those in the short side are often from different industries, which implies that the profit does not only result from the divergence of the stocks in the same industry but also from the divergence between industries.

4.6. Sub-period analysis

We divide the sample period into five sub-periods and examine the performance of the clustering algorithms in each of these sub-periods. The results are reported in Table 6.

The first sub-period commences from January 1980 and ends in December 1999, prior to the widespread use of quantitative and

---

[6] https://www.naics.com/sic-codes-industry-drilldown/

Table 4

Factor regression. This table reports the factor regression results of the equally-weighted pairs trading portfolio constructed via agglomerative clustering. FF3, FF3+2, FF5, and q^5 respectively denote the Fama-French three factors, FF3 plus momentum and short-term reversal factors, Fama-French five factors, and Hou-Mo-Xue-Zhang q^5 factors. The sample period is from 1980.01 to 2020.12 except for q^5, for which the period ends in 2019.12. The values in parentheses are the Newey-West t-statistics.

	FF3	FF3+2	FF5	q^5
Intercept	0.019***	0.020***	0.019***	0.019***
	(14.160)	(15.009)	(14.014)	(12.355)
Market	0.149***	0.054	0.159***	0.154***
	(3.853)	(1.469)	(4.302)	(4.067)
SMB	0.040	0.019		
	(0.568)	(0.418)		
HML	0.180**	0.085		
	(2.158)	(0.997)		
Momentum		-0.125***		
		(-2.822)		
Reversal		0.293***		
		(4.847)		
SMB5			0.075	
			(1.148)	
HML5			0.145	
			(1.216)	
RMW5			0.095	
			(1.059)	
CMA5			0.039	
			(0.284)	
R_ME				0.037
				(0.445)
R_IA				0.231***
				(2.723)
R_ROE				-0.145*
				(-1.915)
R_EG				0.048
				(0.535)
R2	0.086	0.285	0.092	0.103
Adj. R2	0.080	0.277	0.083	0.093
Num. obs.	492	492	492	480

* $p < .10$, ** $p < .05$, and *** $p < .01$

machine learning algorithms in investment (Krauss et al., 2017). Even the benchmark reversal portfolio performs superbly during this period, attaining an annualized mean return of 34.5% and a Sharpe ratio of 2.08. However, the portfolios constructed via *k*-means and agglomerative clustering still outperform the benchmark significantly during this period: they attain annualized mean returns of 24.8% and 27.9%, and the Sharpe ratios of 3.52 and 4.14, respectively. DBSCAN yields a Sharpe ratio of 2.19, comparable to that of the reversal portfolio, but it has a lower maximum drawdown and higher Sortino ratio. Overall, all three clustering methods perform superbly during this period.

The second sub-period ranges from January 2000 to December 2006, which coincides with the advent of quantitative and machine learning algorithms for investment (Dempster & Leemans, 2006; Jae Won Lee, 2001; Vidyamurthy, 2004). Cheaper and higher performing computing hardware accelerated the adoption of quantitative approaches in this period, allowing more quantitative algorithms to be employed for investment. Hence, it is not surprising to observe a dip in the performance of all portfolios in this period compared to the previous period. Nevertheless, the clustering methods still manage to yield annualized mean returns above 26% and Sharpe ratios above 2.2. These values are significantly higher than those of the benchmark reversal portfolio, which are 21.6% and 0.51. DBSCAN performs particularly well in this period attaining an annualized mean return of 35.6%, and a Sharpe ratio of 2.96.

The third sub-period, January 2007 to December 2009, corresponds to the financial crisis caused by the collapse of housing prices in the US and its recovery thereafter. All three clus-

tering methods remain profitable during the financial crisis and fare better than the benchmark reversal portfolio, except for DBSCAN, which performs comparably to the benchmark. To our surprise, agglomerative clustering performs superbly during the recession, with an annualized mean return of 25.1% and a Sharpe ratio of 1.90. Moreover, the maximum drawdown is mere 4.7%, which is significantly lower than those of the benchmarks; 50.2% for S&P500 and 21.2% for the reversal portfolio. This finding is in line with Do & Faff (2010) and Huck & Afawubo (2015), who find that pairs trading strategies perform better during a financial crisis.

The fourth sub-period ranges from January 2010 to December 2019. We discover that the returns generated by *k*-means and agglomerative clustering deteriorate during this period: they earn an annualized mean return below 15%. This finding is in line with Clegg & Krauss (2018), Rad et al. (2016), and Krauss et al. (2017), who observe that pairs trading strategies generate lower returns in recent years due to the widespread use of quantitative and machine learning algorithms. Any market inefficiency is rapidly arbitraged away, eroding any potential profits to be generated. However, we still observe a significant amount of profits generated by DBSCAN in this period, with an annualized mean return of 23.6% and a Sharpe ratio of 1.98. Agglomerative clustering attains a comparatively lower return of 13.6% and a Sharpe ratio of 1.75. The *k*-means clustering is found to be the worst-performing clustering method in this period, generating an annualized mean return of 12.9% and a Sharpe ratio of 1.72. Nevertheless, all three clustering methods outperform both benchmarks in terms of all metrics. The benchmark reversal portfolio underperforms S&P500 with a mean return of 3.7% and a Sharpe ratio of 0.23, suggesting that a naïve reversal strategy can no longer generate profits.

The last sub-period is the year 2020 when the COVID-19 pandemic swept the world. To contain the spread of the virus, governments worldwide have implemented various degrees of lockdowns, which have severely depressed the global economy. Equity prices plummeted while the stock market volatility skyrocketed to an unprecedented level, which surpassed what was observed during the global financial crisis in 2008, as well as the Great Depression in 1929 (Baker et al., 2020).

The S&P500 index plunges by 19.4% till March 2020, while the reversal portfolio loses 15.9% in the same period. In contrast, the portfolios based on clustering fall by a much smaller magnitude. The agglomerative clustering loses only 3.8%, and the other clustering methods also perform robustly with a maximum drawdown below 12%.

During the recovery from April 2020 onward, S&P500 recovers all the losses incurred till March 2020. The clustering-based strategies recover more rapidly and achieve significantly positive returns. DBSCAN performs particularly well generating a mean return higher than 50%. This result reaffirms the superiority and resilience of our strategies in a period of economic crisis.

The deterioration of the performance since 2000 is commonly observed across quantitative trading strategies. One reason for this attenuation of equity return anomalies is the decimalization in the U.S. stock exchanges in 2001 (Chordia, Subrahmanyam, & Tong, 2014; Green et al., 2017). The advent of quantitative trading algorithms and their sophistication in the 21st century must have contributed to the reduction of arbitrage opportunities and profitability as well. Our strategies also perform worse post 2000 but the attenuation is far less compared to the reversal strategy.

5. Robustness check

Given the clustering methods' strong financial performance, coupled with the silent reproach of data snooping with machine learning algorithms, we perform a series of robustness check to ensure the performance is not sensitive to model parameters. Un-

Table 5

Size and industry breakdown of the clusters. This table reports the composition of the stocks traded under each clustering method during the test period from 1980.01 to 2020.12. Panel (a) reports the percentage of the stocks in each size decile, where 1 (10) denotes the smallest (largest) stocks. Panel (b) reports the industry sector composition of the stocks. The industry classification is by the North American Industry Classification System (NAICS). The column Market denotes the composition of the entire market. The reported values are time-series averages over the sample period.

(a) Size deciles

	k-means			DBSCAN			Agglomerative		
	L	S	L-S	L	S	L-S	L	S	L-S
10	9.82	11.71	10.77	14.31	16.21	15.26	11.68	13.47	12.57
9	12.24	13.52	12.88	16.96	17.63	17.30	15.29	16.18	15.74
8	12.47	13.78	13.13	15.66	15.63	15.65	15.17	15.69	15.43
7	12.84	13.34	13.09	12.43	13.24	12.84	14.05	14.31	14.18
6	12.27	12.38	12.32	10.99	10.82	10.91	12.33	12.27	12.30
5	10.97	10.90	10.94	8.88	8.54	8.71	9.88	9.81	9.84
4	9.94	9.09	9.52	7.80	7.43	7.62	8.59	8.05	8.32
3	8.69	7.44	8.07	7.30	6.41	6.85	6.76	5.83	6.30
2	7.02	5.46	6.24	4.60	3.56	4.08	4.80	3.64	4.22
1	3.72	2.37	3.04	1.05	0.54	0.80	1.44	0.77	1.10
Average	6.11	6.42	6.27	6.78	6.91	6.84	6.59	6.79	6.69

(b) Industry breakdown

	k-means			DBSCAN			Agglomerative			Market
	L	S	L-S	L	S	L-S	L	S	L-S	
Manufacturing	47.59	47.39	47.49	27.89	26.83	27.36	42.99	42.81	42.90	43.56
Finance, insurance, real estate	18.94	19.00	18.97	49.55	51.97	50.76	31.78	31.86	31.82	16.06
Services	11.21	11.51	11.36	2.16	2.24	2.20	7.20	7.23	7.21	13.78
Transportation, communications, utilities	8.22	8.02	8.12	15.95	14.72	15.33	8.84	8.86	8.85	8.79
Retail trade	5.69	5.84	5.76	1.25	1.23	1.24	3.64	3.72	3.68	6.51
Wholesale trade	3.33	3.41	3.37	1.10	1.04	1.07	2.49	2.43	2.46	3.97
Mining	3.08	2.84	2.96	0.81	0.70	0.76	1.63	1.57	1.60	4.37
Construction	0.93	0.91	0.92	0.32	0.44	0.38	0.69	0.68	0.68	1.26
Nonclassifiable establishments	0.70	0.75	0.72	0.67	0.58	0.63	0.50	0.58	0.54	1.23
Agriculture, forestry, fishing	0.16	0.19	0.18	0.05	0.04	0.04	0.06	0.11	0.09	0.29
Public administration (government)	0.02	0.02	0.02	0.00	0.00	0.00	0.00	0.01	0.01	0.02
Number of industries within each cluster	1.32	1.32	1.83	2.74	2.74	3.47	1.03	1.03	1.32	

like many machine learning methods, which often involve dozens of parameters and have high degrees of freedom, the clustering algorithms have only few parameters, e.g., the number of clusters K for the k-means clustering and the outlier threshold α for DBSCAN and agglomerative clustering, and there is not much room for data snooping. Besides the model parameters, we also check the robustness against firm size, transaction cost, and the number of principal components.

5.1. Parameter sensitivity of the strategy

Table 7 analyzes the sensitivity of the pairs trading strategies to the model parameters. We find that the strategies are robust across a wide range of parameter values.

When we increase the number of clusters for k-means clustering from 5 to 1500, the number of clusters after removing outliers also increases. In contrast, the number of clustered stocks and the number of traded stocks decrease. This is because a larger K generates many tiny clusters whose elements do not meet the trading criteria. The mean return increases with K as only a small number of stocks that are closest to each other are paired and form the portfolio. Nevertheless, the Sharpe ratio does not exhibit a monotonic trend and has its highest value when $K = 500$. Achieving good performance requires a balance between selecting the most profitable pairs and increasing diversification. Overall, the strategy is robust to K, with the mean return ranging between 17.1% and 25.2% and the Sharpe ratio ranging between 1.50 and 2.34.

DBSCAN clusters fewer stocks when α decreases, i.e. when the maximum distance for a pair to be connected decreases. However, the number of clusters remains at two, except when $\alpha = 0.9$ that generates only one cluster. A smaller α results in a portfolio with fewer stocks and yields higher returns and Sharpe ratio, and the

best performance occurs when $\alpha = 0.1$, the minimum value we test. Decreasing α below 0.1 may further improve the portfolio performance. The strategy based on DBSCAN appears to be more sensitive than the one from k-means clustering: the mean return ranges between 12.9% and 25.6% and the Sharpe ratio ranges between 0.87 and 2.04.

Agglomerative clustering trades fewer stocks as the maximum distance for linkage decreases. Like in k-means clustering, selecting fewer stocks yields a higher return, but the Sharpe ratio starts to decrease when α is smaller than 0.3. The performance varies rather widely, with the range of the mean return between 15.3% and 31.5% and the range of the Sharpe ratio between 1.27 and 2.69. Still, the strategy earns a mean return of 15.3% and a Sharpe ratio of 1.27 even in the worst case ($\alpha=0.9$), in which only a small portion of the stocks (16.5%) are removed as outliers, and the portfolio consists of almost 1000 stocks, on average. Overall, all three clustering methods perform robustly within a reasonable range of the parameter values.

5.2. Size effects

It might be that the returns generated by the clustering methods are attributed to illiquid, small-cap stocks. To assess the role of small-cap stocks, we repeat the empirical analysis after removing stocks with market capitalization smaller than the 10% or 20% NYSE-size quantile. Table 8 (rows 'NYSE10' and 'NYSE20') reports the results.

Removing small-cap stocks lowers the mean return. However, it also reduces the volatility, and the Sharpe ratio declines only moderately when the stocks below 10% NYSE-size quantile are excluded: the Sharpe ratio is decreased from 2.34 to 2.30 for k-means clustering, 2.04 to 1.79 for DBSCAN, and 2.69 to 2.42 for agglomer-

Table 6

Sub-period analysis. This table reports sub-period performances of the equally-weighted pairs trading portfolios constructed via *k*-means clustering, DBSCAN, and agglomerative clustering. The sub-periods are defined in the table. The values are annualized where applicable.

	k-means	DBSCAN	Agglomerative	S&P500	Reversal
01/1980 - 12/1999					
Mean return	0.248	0.231	0.279	0.178	0.345
Sharpe ratio	3.521	2.189	4.138	1.186	2.078
Sortino ratio	9.429	4.019	12.260	1.585	3.325
Profit factor	17.918	5.467	28.955	2.452	7.493
Maximum drawdown	-0.045	-0.082	-0.032	-0.296	-0.192
Calmar ratio	5.515	2.807	8.742	0.601	1.799
01/2000 - 12/2006					
Mean return	0.268	0.356	0.296	0.023	0.216
Sharpe ratio	2.276	2.964	2.558	0.162	0.510
Sortino ratio	4.347	16.470	3.958	0.235	0.917
Profit factor	8.712	20.465	8.739	1.131	1.764
Maximum drawdown	-0.079	-0.037	-0.123	-0.444	-0.406
Calmar ratio	3.400	9.671	2.402	0.052	0.532
01/2007 - 12/2009					
Mean return	0.231	0.181	0.251	-0.035	0.165
Sharpe ratio	1.690	0.891	1.904	-0.176	0.671
Sortino ratio	4.740	2.189	3.944	-0.223	1.813
Profit factor	4.625	2.087	5.027	0.875	1.764
Maximum drawdown	-0.061	-0.149	-0.047	-0.502	-0.212
Calmar ratio	3.820	1.210	5.331	-0.069	0.782
01/2010 - 12/2019					
Mean return	0.129	0.236	0.136	0.135	0.037
Sharpe ratio	1.716	1.977	1.753	1.086	0.232
Sortino ratio	4.627	3.579	3.619	1.603	0.486
Profit factor	4.216	5.169	3.698	2.214	1.198
Maximum drawdown	-0.076	-0.140	-0.091	-0.162	-0.260
Calmar ratio	1.703	1.684	1.499	0.835	0.141
01/2020 - 12/2020					
Mean return	0.002	0.496	0.420	0.209	0.511
Sharpe ratio	0.013	2.112	2.040	0.805	1.302
Sortino ratio	0.028	6.020	8.433	1.379	3.421
Profit factor	1.011	5.958	6.191	1.777	3.389
Maximum drawdown	-0.113	-0.059	-0.038	-0.194	-0.159
Calmar ratio	0.020	8.390	11.067	1.076	3.224

ative clustering. Excluding the stocks below 20% NYSE-size quantile does not deteriorate the performance further. The mean return and the Sharpe ratio remain almost at the same level and even increase for *k*-means clustering.

Although small-cap stocks appear to have a certain extent of contribution to the strategies' profitability, the strategies can still generate significant profits when applied to relatively large stocks. This result reveals that the profits of the strategies are not a mere small-firm effect and they are scalable for large position sizes.

5.3. Value-weighted portfolio

For the pairs trading strategy to work as intended, the stocks in a pair should be equally weighted, *i.e.*, buy one stock and sell the other for the same amount. As the pairs trading portfolio is simply a group of long-short pairs, it should also be equally weighted. Nevertheless, to further investigate the effects of firm size, we construct value-weighted pairs trading portfolios and report their performance in Table 8 (row 'VW'). For a fair comparison, the reversal portfolio is also value-weighted.

The value-weighted portfolios perform worse compared to their equally-weighted counterparts: the mean returns and Sharp ratios are approximately halved, which is in line with the findings of previous studies, *e.g.*, Gu et al. (2020), Han (2021). Nevertheless, they still outperform the value-weighted reversal portfolio and S&P500, generating higher returns and Sharpe ratios. In particular, the agglomerative clustering yields an annualized mean return of 13.4% and a Sharpe ratio of 1.09. In contrast, the S&P500 index and the reversal portfolio yield mean returns of 12.6% and 2.3% and Sharpe ratios of 0.84 and 0.09, respectively.

5.4. Impact of transaction costs

Our strategies are essentially a contrarian strategy and have a relatively high turnover (about 180%). But they require rebalancing only once a month and transaction costs should have a limited

Table 7

Parameter sensitivity of the strategies. This table reports the performance of the equally-weighted long-short portfolio constructed from each clustering method with different parameter values. The *k*-means clustering is tested with different numbers of clusters (K), and DBSCAN and agglomerative clustering are tested with different percentiles (α) for the outlier threshold.

(a) *k*-means

K	5	10	50	100	500	1000	1500		
Mean return	0.171	0.179	0.195	0.198	0.215	0.248	0.252		
Sharpe ratio	1.499	1.651	1.898	1.950	2.336	1.991	1.629		
Maximum drawdown	-0.213	-0.217	-0.190	-0.215	-0.113	-0.125	-0.307		
Number of clusters	4.000	8.000	30.000	52.000	129.000	134.000	80.000		
Number of stocks in clusters	1561 (50.0)	1561 (49.99)	1555 (49.81)	1543 (49.38)	1257 (39.78)	789 (24.22)	391 (10.94)		
Number of outliers	1596 (50.0)	1596 (50.01)	1602 (50.19)	1614 (50.62)	1900 (60.22)	2368 (75.78)	2766 (89.06)		
Number of stocks traded	536 (17.19)	524 (16.8)	490 (15.7)	472 (15.11)	330 (10.33)	166 (4.91)	60 (1.64)		
(b) DBSCAN									
α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Mean return	0.256	0.200	0.173	0.154	0.148	0.148	0.138	0.134	0.129
Sharpe ratio	2.039	1.763	1.526	1.328	1.190	1.142	1.036	0.954	0.867
Maximum drawdown	-0.149	-0.146	-0.163	-0.206	-0.197	-0.242	-0.238	-0.285	-0.332
Number of clusters	2	2	2	2	2	2	2	2	1
Number of stocks in clusters	376 (12.05)	749 (24.06)	1092 (35.02)	1418 (45.44)	1729 (55.41)	2034 (65.14)	2328 (74.56)	2609 (83.54)	2878 (92.2)
Number of outliers	2781 (87.95)	2408 (75.94)	2065 (64.98)	1739 (54.56)	1428 (44.59)	1123 (34.86)	829 (25.44)	548 (16.46)	279 (7.8)
Number of stocks traded	75 (2.47)	188 (6.17)	314 (10.18)	450 (14.52)	594 (19.09)	743 (23.82)	896 (28.67)	1053 (33.67)	1217 (38.86)
(c) Agglomerative									
α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Mean return	0.315	0.270	0.248	0.237	0.215	0.199	0.181	0.167	0.153
Sharpe ratio	2.126	2.440	2.692	2.681	2.436	2.172	1.823	1.567	1.268
Maximum drawdown	-0.232	-0.117	-0.123	-0.103	-0.132	-0.129	-0.197	-0.185	-0.196
Number of clusters	80	140	188	223	247	258	252	224	161
Number of stocks in clusters	260 (8.34)	515 (16.48)	769 (24.58)	1032 (33.01)	1304 (41.74)	1592 (50.95)	1901 (60.84)	2234 (71.55)	2606 (83.46)
Number of outliers	2897 (91.66)	2642 (83.52)	2388 (75.42)	2125 (66.99)	1853 (58.26)	1565 (49.05)	1256 (39.16)	923 (28.45)	551 (16.54)
Number of stocks traded	23 (0.74)	65 (2.12)	126 (4.05)	207 (6.66)	308 (9.89)	432 (13.84)	584 (18.69)	769 (24.58)	1001 (31.97)

Table 8

Robustness check. This table reports the results from the robustness check in Section 5. 'NYSE*x*' denotes the results after excluding stocks smaller than *x*% NYSE-size quantile, and 'VW' denotes the value-weighted portfolio performance. '*x* bps' denotes the results after considering transaction costs of *x* bps. '*x* std' denotes the results when the trading threshold is set to be *x* times the cross-sectional standard deviation, and '*x* tstd' denotes the results when the trading threshold is set to be *x* times the time-series standard deviation. '*x*%' denotes the results when the portfolios are constructed from the *x*% highest and lowest return stocks. 'PCA*x*' denotes the results when the principal components are chosen so that they can explain *x*% of the variation. The test period is from 1980.01 to 2020.12.

		k-means			DBSCAN			Agglomerative			Benchmarks	
		L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
NYSE10	MR	0.245	0.061	0.184	0.243	0.034	0.210	0.252	0.048	0.204	0.126	0.239
	SR	1.288	0.384	2.297	1.311	0.223	1.789	1.362	0.320	2.421	0.835	0.983
	MDD	-0.473	-0.507	-0.068	-0.587	-0.561	-0.238	-0.454	-0.509	-0.077	-0.502	-0.503
NYSE20	MR	0.225	0.063	0.163	0.236	0.044	0.193	0.240	0.055	0.185	0.126	0.239
	SR	1.212	0.395	2.026	1.314	0.295	1.578	1.355	0.367	2.293	0.835	0.983
	MDD	-0.491	-0.522	-0.108	-0.502	-0.493	-0.256	-0.484	-0.515	-0.097	-0.502	-0.503
VW	MR	0.174	0.067	0.107	0.202	0.054	0.147	0.194	0.061	0.134	0.126	0.023
	SR	0.907	0.413	0.865	1.025	0.337	0.900	0.974	0.350	1.087	0.835	0.089
	MDD	-0.540	-0.619	-0.265	-0.693	-0.565	-0.477	-0.601	-0.551	-0.241	-0.502	-0.747
10 bps	MR	0.249	0.032	0.172	0.241	-0.017	0.213	0.254	0.004	0.204	0.126	0.239
	SR	1.208	0.196	1.868	1.334	-0.117	1.701	1.356	0.023	2.225	0.835	0.983
	MDD	-0.524	-0.597	-0.121	-0.503	-0.878	-0.157	-0.446	-0.693	-0.132	-0.502	-0.503
20 bps	MR	0.227	0.010	0.128	0.219	-0.040	0.167	0.231	-0.019	0.159	0.126	0.239
	SR	1.103	0.062	1.394	1.211	-0.270	1.343	1.235	-0.123	1.731	0.835	0.983
	MDD	-0.541	-0.683	-0.151	-0.525	-0.936	-0.170	-0.467	-0.836	-0.143	-0.502	-0.503
0.5 std	MR	0.243	0.078	0.165	0.222	0.054	0.168	0.241	0.065	0.176	0.126	0.239
	SR	1.317	0.504	2.482	1.353	0.382	1.865	1.423	0.443	2.997	0.835	0.983
	MDD	-0.496	-0.564	-0.061	-0.483	-0.564	-0.131	-0.425	-0.542	-0.050	-0.502	-0.503
1.5 std	MR	0.307	0.033	0.275	0.289	-0.019	0.308	0.318	0.017	0.301	0.126	0.239
	SR	1.334	0.180	2.098	1.363	-0.119	1.749	1.494	0.097	1.933	0.835	0.983
	MDD	-0.492	-0.639	-0.146	-0.635	-0.889	-0.271	-0.400	-0.707	-0.477	-0.502	-0.503
0.5 tstd	MR	0.221	0.094	0.127	0.202	0.083	0.119	0.216	0.088	0.128	0.126	0.239
	SR	1.306	0.639	2.533	1.322	0.612	1.685	1.388	0.631	3.047	0.835	0.983
	MDD	-0.484	-0.547	-0.043	-0.461	-0.551	-0.099	-0.434	-0.516	-0.039	-0.502	-0.503
1.0 tstd	MR	0.229	0.083	0.146	0.211	0.070	0.141	0.225	0.077	0.147	0.126	0.239
	SR	1.334	0.565	2.594	1.349	0.518	1.784	1.425	0.560	3.066	0.835	0.983
	MDD	-0.486	-0.545	-0.046	-0.461	-0.554	-0.112	-0.429	-0.523	-0.056	-0.502	-0.503
1.5 tstd	MR	0.240	0.073	0.167	0.219	0.059	0.161	0.233	0.067	0.166	0.126	0.239
	SR	1.373	0.498	2.607	1.374	0.431	1.835	1.459	0.483	3.063	0.835	0.983
	MDD	-0.488	-0.541	-0.061	-0.458	-0.549	-0.123	-0.425	-0.520	-0.059	-0.502	-0.503
10%	MR	0.249	0.072	0.177	0.255	0.004	0.251	0.216	0.089	0.127	0.126	0.239
	SR	1.300	0.456	2.249	1.394	0.028	1.997	1.308	0.604	2.808	0.835	0.983
	MDD	-0.534	-0.591	-0.068	-0.514	-0.754	-0.197	-0.469	-0.546	-0.052	-0.502	-0.503
5%	MR	0.252	0.077	0.175	0.264	-0.019	0.284	0.215	0.092	0.122	0.126	0.239
	SR	1.256	0.461	2.068	1.313	-0.126	1.861	1.287	0.620	2.682	0.835	0.983
	MDD	-0.556	-0.597	-0.082	-0.596	-0.858	-0.311	-0.471	-0.545	-0.058	-0.502	-0.503
PCA90	MR	0.279	-0.021	0.300	0.265	-0.002	0.267	0.282	0.037	0.245	0.126	0.239
	SR	1.487	-0.142	2.283	1.414	-0.014	2.083	1.472	0.235	2.432	0.835	0.983
	MDD	-0.440	-0.797	-0.149	-0.517	-0.877	-0.153	-0.446	-0.599	-0.224	-0.502	-0.503
PCA95	MR	0.271	0.058	0.213	0.263	0.011	0.252	0.278	0.038	0.240	0.126	0.239
	SR	1.307	0.351	2.259	1.466	0.075	2.001	1.472	0.241	2.288	0.835	0.983
	MDD	-0.495	-0.587	-0.156	-0.491	-0.773	-0.150	-0.445	-0.567	-0.231	-0.502	-0.503
PCA100	MR	0.272	0.054	0.218	0.258	0.003	0.256	0.280	0.032	0.248	0.126	0.239
	SR	1.325	0.327	2.316	1.392	0.016	2.018	1.478	0.203	2.643	0.835	0.983
	MDD	-0.491	-0.581	-0.141	-0.486	-0.850	-0.154	-0.449	-0.584	-0.096	-0.502	-0.503

impact. Table 8 (row '10 bps' and '20 bps') presents the portfolios' risk-return metrics after taking transaction costs into account. The transaction cost is assumed to be either 10 or 20 basis points (bps) per side of the transaction.

After factoring in 10 bps of transaction costs, the portfolio constructed via agglomerative clustering earns an annualized mean return of 20.4% and a Sharpe ratio of 2.23. DBSCAN and *k*-means clustering earn annualized mean returns of 17.2% and 21.3% and annualized Sharpe ratios of 1.87 and 1.70, respectively, which are still promising and higher than those of the benchmarks. Even with the conservatively assumed transaction cost of 20 bps, agglomerative clustering performs impressively, attaining an annualized mean return of 15.9% and a Sharpe ratio of 1.73. The *k*-means clustering and DBSCAN earn annualized mean returns of 12.8% and 16.7% and Sharpe ratios of 1.39 and 1.34, respectively, and outperform the benchmarks.

In an unreported analysis, we also examine the performance after transaction costs in the recent periods. As the performance is attenuated post 2000, the after-cost performance also becomes weaker, but the strategies still generate significant performances. For instance, the Sharpe ratio of the agglomerative clustering with 20 bps transaction cost is 1.78, 1.24, 0.56, and 1.60, respectively in the sub-periods of 2000–2006, 2007–2009, 2010–2019, and 2020. The *k*-means and DBSCAN produce the Sharpe ratio of 1.54, 1.07, 0.54, and -0.48, and 2.20, 0.47, 1.21, and 1.76 for the same sub-periods. These results suggest that our strategies are likely to remain profitable even when the transaction costs are relatively high.

5.5. Trading threshold

Previously, we use one standard deviation of the past one-month return difference as a threshold to trigger pairs trading.

Here, we test different thresholds and examine their impacts on portfolio performance. A larger threshold selects pairs that have diverged more significantly in the previous month and trades fewer stocks. Table 8 reports the results from different methods to set the threshold.

When the threshold is set to be 0.5 times the standard deviation (row '0.5 std'), the stocks traded are doubled, and the returns become lower while the Sharpe ratios change little. Raising the threshold to 1.5 times the standard deviation (row '1.5 std') increases the return but lowers the Sharpe ratio, which can be attributed to the reduced diversification effect. Still, our strategies outperform the benchmarks in both cases.

Instead of the cross-sectional standard deviation, we can also compare the divergence against the time-series standard deviation of the return difference of a pair. In the table, '*x* tstd' denotes the results when the trading threshold is set to be *x* times the time-series standard deviation. We find that using the time-series standard deviation improves the performance compared to using the cross-sectional standard deviation. This is perhaps because the threshold is based on the historical return difference of the pair and varies across the pairs, whereas the threshold based on the cross-sectional standard deviation is the same across all pairs. Remarkably, increasing the threshold increases both the mean return and the Sharpe ratio. When the threshold is set to be the time-series standard deviation, more pairs are selected (the number of stocks traded are 788, 244, and 460, respectively for the *k*-means, DBSCAN, and Agglomerative), and the portfolio appears to be sufficiently diversified even when the threshold is 1.5 times the time-series standard deviation.

We also test a crude approach that simply chooses a certain percentile of the stocks from the top and the bottom. When the portfolio is formed by going long on the 10% lowest return stocks and short on the 10% highest return stocks within each cluster (row '10%'), it obtains a high Sharpe ratio, but the return tends to be lower, with the exception of DBSCAN, which earns a similar level of return: the Sharpe ratios of *k*-means, DBSCAN, and agglomerative clustering are respectively 2.25, 2.00, and 2.81, and the mean returns are 17.7%, 25.1%, and 12.7%. Choosing more extreme return stocks with a 5% threshold (row '5%') renders similar results.

Choosing only a few pairs that have diverged significantly can yield a higher return, but the Sharpe ratio can deteriorate if the portfolio is not well diversified. A balance between the return and the volatility needs to be found. The result above suggests that the strategies perform robustly for a range of the threshold. It also suggests that determining the threshold dynamically based on the cross-sectional dispersion of the previous one-month return performs better than selecting a fixed proportion of the stocks.

5.6. Number of principal components

Table 8 (row 'PCA90', 'PCA95', and 'PCA100') reports the sensitivity of the clustering methods to the number of principal components. We choose the number of principal components so that they can explain 90%, 95%, or 100% of the variation. We use 100% since the objective of the PCA in our application is not to reduce the dimension but to transform the feature space so that the principal components rather than the raw features receive the same weight in the distance metric. On average, 50 out of 126 principal components are selected at 90%. 63 principal components at 95%, and 86 principal components at 99%.

The results suggest that the pairs trading strategies are robust to the change of the number of principal components. Most risk-return metrics change only slightly with the number of principal components. Clustering algorithms are notoriously slow for a large set of high-dimensional data, and choosing the principal components at a lower variance level, *e.g.*, 95%, may save the computational costs significantly without sacrificing portfolio performance.

6. Conclusion

In this paper, we develop pairs trading strategies via unsupervised learning. We employ three clustering methods, *i.e.*, *k*-means clustering, DBSCAN, and agglomerative clustering, and utilise the information from firm characteristics as well as past returns.

We test our strategies on the US stock market during the period of January 1980 to December 2020 and find all three strategies significantly outperform both the market and the short-term reversal benchmarks. The agglomerative clustering-based strategy appears to be the best performing strategy with an annualised mean excess return of 23.8% and an annualised Sharpe ratio of 2.69. It also has a low maximum drawdown of 12.3%, which lasts for only two months. Factor models cannot explain the significant profits of the strategies. The strategies perform robustly during financial crises, such as the 2007 financial crisis and the 2020 market crash.

Firm characteristics are revealed to be an important source of information in identifying pairs. Clustering stocks based on their characteristics as well as past returns significantly reduces the volatility and downside risk of the strategies and improves performance.

The clustering methods select stocks mainly from the manufacturing and financial sectors and from mid- to large-cap deciles, making the strategies highly scalable. Each cluster consists of only one or few sectors, which implies that stocks in the same industry tend to behave in a similar manner. It appears that the profitability of the strategies are not only resulted from the divergence of the stocks in the same industry, but also from the divergence between different industries.

A sub-period analysis and robustness tests reveal that the strategies perform robustly under different conditions and are not sensitive to changes of model parameter values.

In conclusion, clustering stocks incorporating the information from firm characteristics improves the identification of stock pairs and the performance of the pairs trading strategy.

We applied unsupervised learning for identifying pairs among stocks. It will be interesting to apply the same method to other asset classes such as fixed income securities or cryptocurrencies. Bianchi et al. (2020) apply the methodology of Gu et al. (2020) to the US Treasury bond market and find machine learning-based strategies are profitable also in the bond market. Based on their findings, it is expected that our method can also generate a profitable pairs trading strategy in the fixed income market. Cryptocurrencies are extremely volatile and a market-neutral strategy is more important than in any other markets. Cryptocurrencies of the same kind, *e.g.*, NFT coins or DeFi coins, tend to move in tandem and these characteristics could be utilized to construct a successful market-neutral pairs trading strategy. We leave this for future research.

Another direction of future research could be a combination of supervised learning and unsupervised learning. In this paper, undervalued and overvalued stocks are identified based on their previous one-month return. Instead, one could predict stocks' (relative) future returns using supervised learning, *e.g.*, as done in Han (2021), and use them to identify pairs within a cluster.

Appendix A. Firm characteristics

Acronym	Firm characteristic	Acronym	Firm characteristic
absacc	Absolute accruals	invest	Capital expenditures and inventory
acc	Working capital accruals	IPO	New equity issue
aeavol	Abnormal earnings announcement volume	lev	Leverage
age	# years since first Compustat coverage	lgr	Growth in long-term debt
agr	Asset growth	maxret	Maximum daily return
baspread	Bid-ask spread	ms	Financial statement score
beta	Beta	mve	Size
betasq	Beta squared	mve_ia	Industry-adjusted size
bm	Book-to-market	nincr	Number of earnings increases
bm_ia	Industry-adjusted book to market	operprof	Operating profitability
cash	Cash holdings	pchcapx_ia	Industry adjusted % change in capital expenditures
cashdebt	Cash flow to debt	pchcurrat	% change in current ratio
cashpr	Cash productivity	pchdepr	% change in depreciation
cfp	Cash flow to price ratio	pchgm_pchsale	% change in gross margin - % change in sales
cfp_ia	Industry-adjusted cash flow to price ratio	pchquick	% change in quick ratio
chatoia	Industry-adjusted change in asset turnover	pchsale_pchrect	% change in sales - % change in A/R
chcsho	Change in shares outstanding	pctacc	Percent accruals
chempia	Industry-adjusted change in employees	pricedelay	Price delay
chinv	Change in inventory	ps	Financial statements score
chmom	Change in 6-month momentum	quick	Quick ratio
chpmia	Industry-adjusted change in profit margin	rd	R&D increase
chtx	Change in tax expense	retvol	Return volatility
cinvest	Corporate investment	roaq	Return on assets
convind	Convertible debt indicator	roeq	Return on equity
currat	Current ratio	roic	Return on invested capital
depr	Depreciation / PP&E	rsup	Revenue surprise
divi	Dividend initiation	salecash	Sales to cash
divo	Dividend omission	salerec	Sales to receivables
dolvol	Dollar trading volume	securedind	Secured debt indicator
dy	Dividend to price	sgr	Sales growth
ear	Earnings announcement return	sin	Sin stocks
egr	Growth in common shareholder equity	SP	Sales to price
ep	Earnings to price	std_dolvol	Volatility of liquidity (dollar trading volume)
gma	Gross profitability	std_turn	Volatility of liquidity (share turnover)
herf	Industry sales concentration	sue	Unexpected quarterly earnings
hire	Employee growth rate	tang	Debt capacity/firm tangibility
idiovol	Idiosyncratic return volatility	tb	Tax income to book income
ill	Illiquidity	turn	Share turnover
indmom	Industry momentum	zerotrade	Zero trading days

Note: This table lists the 78 firm characteristics used in the study. Readers are referred to Green et al. (2017) and the references therein for the exact definitions.

Appendix B. Clustering algorithms

Algorithm 1: *k*-means clustering.

Input: X: All sample points in month t
K: Number of Clusters
MaxIter: Maximum number of iterations
Tol: tolerance for convergence
1 Initialize centroids μ_k's at random points
2 **for** *i in MaxIter* **do**
3   **for** *each sample point x in X* **do**
4   Assign x to its nearest centroid
5   Update the centroids μ_k's
6   **if** *the l_2 norm of the difference in the cluster centroids of two consecutive iterations is smaller than Tol* **then**
7   Break

Algorithm 2: Identification of outliers in *k*-means clustering.

Input: X: All sample points in month t
α: Percentile to calculate ϵ, the maximum distance to the centroid for a sample point to remain in its cluster
μ_x: The corresponding centroid of sample point x where $x \in X$
1 Compute the $l2$ distances of every point to its corresponding centroid
2 ϵ = the α percentile of the $l2$ distances
3 **for** *each sample point x in X* **do**
4   **if** $||x - \mu_x||_2 > \epsilon$ **then**
5   Mark x as outlier

Algorithm 3: DBSCAN.

Input: X: All sample points in month t
MinPts: Minimum number of sample points per cluster
ϵ: Maximum distance between two samples for one to be considered the neighbourhood of the other
1 $C = 0$
2 **for** *each unvisited sample point x in X* **do**
3   Mark x as visited
4   *NeighborPts* = all points within ϵ from x
5   **if** *size of NeighborPts < MinPts* **then**
6   Mark x as NOISE
7   **else**
8   C = next cluster
9   Add x to C
10   **for** *each point x' in NeighborPts* **do**
11   *NeighborPts'* = all points within ϵ from x'
12   **if** *size of NeighborPts' $\geq$ MinPts* **then**
13   *NeighborPts = NeighborPts $\cup$ NeighborPts'*

Algorithm 4: Agglomerative clustering.

Input: X: All sample points in month t
ϵ: Maximum distance between two clusters for them to be merged
1 Initialization: Treat all sample points as individual clusters
2 **while** *minimum distance between clusters < ϵ* **do**
3   Compute the distances between all clusters
4   Merge the pair of clusters that has the minimum distance into a single cluster

---

References

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory* (pp. 420–434). Springer.

Avellaneda, M., & Lee, J.-H. (2010). Statistical arbitrage in the us equities market. *Quantitative Finance, 10*(7), 761–782.

Babu, M. S., Geethanjali, N., & Satyanarayana, B. (2012). Clustering approach to stock market prediction. *International Journal of Advanced Networking and Applications, 3*(4), 1281.

Baker, S. R., Bloom, N., Davis, S. J., Kost, K. J., Sammon, M. C., & Viratyosin, T. (2020). The unprecedented stock market impact of COVID-19. *Technical Report*. National Bureau of Economic Research.

Barucci, E., Bonollo, M., Poli, F., & Rroji, E. (2021). A machine learning algorithm for stock picking built on information based outliers. *Expert Systems with Applications, 184*, 115497.

Bianchi, D., Büchner, M., & Tamoni, A. (2020). Bond risk premiums with machine learning. *The Review of Financial Studies*.

Bini, B., & Mathew, T. (2016). Clustering and regression techniques for stock prediction. *Procedia Technology, 24*, 1248–1255.

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering, 60*(1), 208–221.

Chahn (2018). BarclayHedge survey: Majority of hedge fund pros use AI/machine learning in investment strategies. *Technical Note*.

Chen, S., Lach, J., Amft, O., Altini, M., & Penders, J. (2013). Unsupervised activity clustering to estimate energy expenditure with a single body sensor. In *2013 IEEE International conference on body sensor networks* (pp. 1–6). IEEE.

Chordia, T., Subrahmanyam, A., & Tong, Q. (2014). Have capital market anomalies attenuated in the recent era of high liquidity and trading activity? *Journal of Accounting and Economics, 58*(1), 41–58.

Clegg, M., & Krauss, C. (2018). Pairs trading with partial cointegration. *Quantitative Finance, 18*(1), 121–138.

Cummins, M., & Bucca, A. (2012). Quantitative spread trading on crude oil and refined products markets. *Quantitative Finance, 12*(12), 1857–1875.

Dempster, M. A., & Leemans, V. (2006). An automated FX trading system using adaptive reinforcement learning. *Expert Systems with Applications, 30*(3), 543–552.

Do, B., & Faff, R. (2010). Does simple pairs trading still work? *Financial Analysts Journal, 66*(4), 83–95.

Dose, C., & Cincotti, S. (2005). Clustering of financial time series with application to index and enhanced index tracking portfolio. *Physica A: Statistical Mechanics and its Applications, 355*(1), 145–151.

Elbatta, M. N. (2012). *An improvement for DBSCAN algorithm for best results in varied densities*. The Islamic University-Gaza Ph.D. thesis..

Elliott, R. J., Van Der Hoek, J., & Malcolm, W. P. (2005). Pairs trading. *Quantitative Finance, 5*(3), 271–276.

Engle, R. F., & Granger, C. W. (1987). Co-integration and error correction: Representation, estimation, and testing. *Econometrica: Journal of the Econometric Society*, 251–276.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). KDD'96: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (pp. 226–231). AAAI Press.

Fama, E. F., & French, K. R. (1996). Multifactor explanations of asset pricing anomalies. *The Journal of Finance, 51*(1), 55–84.

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics, 116*(1), 1–22.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669.

Fung, G. P. C., Yu, J. X., & Lam, W. (2002). News sensitive stock trend prediction. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 481–493). Springer.

Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies, 19*(3), 797–827.

Gowda, K. C., & Ravi, T. (1995). Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity. *Pattern recognition, 28*(8), 1277–1282.

Green, J., Hand, J. R. M., & Zhang, X. F. (2017). The characteristics that provide independent information about average U.S. monthly stock returns. *The Review of Financial Studies, 30*(12), 4389–4436.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies, 33*(5), 2223–2273.

Han, C. (2021). Bimodal characteristic returns and predictability enhancement via machine learning. *Management Science*.

Hautamäki, V., Cherednichenko, S., Kärkkäinen, I., Kinnunen, T., & Fränti, P. (2005). Improving k-means by outlier removal. In *Scandinavian conference on image analysis* (pp. 978–987). Springer.

Hou, K., Mo, H., Xue, C., & Zhang, L. (2020). An augmented q-factor model with expected growth. *Review of Finance*.

Huck, N. (2009). Pairs selection and outranking: An application to the S&P 100 index. *European Journal of Operational Research, 196*(2), 819–825.

Huck, N. (2010). Pairs trading and outranking: The multi-step-ahead forecasting case. *European Journal of Operational Research, 207*(3), 1702–1716.

Huck, N. (2019). Large data sets and machine learning: Applications to statistical arbitrage. *European Journal of Operational Research, 278*(1), 330–342.

Huck, N., & Afawubo, K. (2015). Pairs trading and selection methods: Is cointegration superior? *Applied Economics, 47*(6), 599–613.

Jacobs, N., & Weber, M. (2015). On the determinants of pairs trading profitability. *Journal of Financial Markets, 23*, 75–97.

Jae Won Lee (2001). Stock price prediction using reinforcement learning. In *ISIE 2001. 2001 IEEE International symposium on industrial electronics proceedings (Cat. no.01th8570): vol. 1* (pp. 690–695).

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika, 32*(3), 241–254.

Kaucic, M. (2010). Investment using evolutionary learning methods and technical rules. *European Journal of Operational Research, 207*(3), 1717–1727.

Kim, T., & Kim, H. Y. (2019). Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. *Complexity, 2019*.

Krauss, C. (2017). Statistical arbitrage pairs trading strategies: Review and outlook. *Journal of Economic Surveys, 31*(2), 513–545.

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research, 259*(2), 689–702.

Ledoit, O., & Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. *Journal of Empirical Finance, 15*(5), 850–859.

Lim, B., Zohren, S., & Roberts, S. (2019). Enhancing time-series momentum strategies using deep neural networks. *The Journal of Financial Data Science, 1*(4), 19–38.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability: vol. 1* (pp. 281–297). Oakland, CA, USA

Messmer, M. (2017). Deep learning and the cross-section of expected returns. *Econometric Modeling: Capital Markets - Asset Pricing eJournal*.

Moritz, B., & Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. *Working Paper*.

Nanda, S., Mahanty, B., & Tiwari, M. (2010). Clustering indian stock market data for portfolio management. *Expert Systems with Applications, 37*(12), 8793–8798.

Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications, 115*, 635–655.

Pavlidis, N. G., Plagianakos, V. P., Tasoulis, D. K., & Vrahatis, M. N. (2006). Financial forecasting through unsupervised clustering and neural networks. *Operational Research, 6*(2), 103–127.

Rad, H., Low, R. K. Y., & Faff, R. (2016). The profitability of pairs trading strategies: Distance, cointegration and copula methods. *Quantitative Finance, 16*(10), 1541–1558.

Rahmah, N., & Sitanggang, I. S. (2016). Determination of optimal epsilon (Eps) value on DBSCAN algorithm to clustering data on peatland hotspots in sumatra. In *IOP conference series: Earth and environmental science: vol. 31* (p. 012012). IOP Publishing.

Sarmento, S. M., & Horta, N. (2020). Enhancing a pairs trading strategy with the application of machine learning. *Expert Systems with Applications, 158*, 113490.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS), 42*(3), 1–21.

Shumway, T., & Warther, V. A. (1999). The delisting bias in CRSP's NASDAQ data and its implications for the size effect. *The Journal of Finance, 54*(6), 2361–2379.

Siraj, M. M., Maarof, M. A., & Hashim, S. Z. M. (2009). Intelligent clustering with PCA and unsupervised learning algorithm in intrusion alert correlation. In *2009 fifth international conference on information assurance and security: vol. 1* (pp. 679–682). IEEE.

Ta, V.-D., & Liu, C.-M. (2016). Stock market analysis using clustering techniques: The impact of foreign ownership on stock volatility in vietnam. In *Proceedings of the seventh symposium on information and communication technology* (pp. 99–106).

Vidyamurthy, G. (2004). *Pairs trading: Quantitative methods and analysis*: vol. 217. John Wiley & Sons.

Xie, W., Liew, R. Q., Wu, Y., & Zou, X. (2016). Pairs trading with copulas. *The Journal of Trading, 11*(3), 41–52.

Yang, Y., Lian, B., Li, L., Chen, C., & Li, P. (2014). DBSCAN clustering algorithm applied to identify suspicious financial transactions. In *2014 International conference on cyber-enabled distributed computing and knowledge discovery* (pp. 60–65). IEEE.