# Profitable Pairs Selection Framework and Pairs Trading Strategy: Based on CMS Open Payments Data

## 1 Background and Questions

The Center for Medicare and Medicaid services (CMS) was founded in 1965 to "provide access to high quality care and improved health at lower costs". The open payments databases of the CMS are collected and published to provide the public with a more transparent health care system. In academia, there have been various researches and investigations conducted based on this public dataset, ranging from physician targeting [1] to compliance risks identification [5] and COVID-19 impact [10].

In this report, we are going to explore the following interesting questions based on general payments, research payments, and ownership/investment interests data for 2020(Refresh publication) and 2021(Initial publication).

- *From investors' point of view, do the payments to entities and individuals(such as physicians) contain hidden information not fully captured by any other public data source and thus the markets? For example, is similar payment behavior a good indicator of comparability between companies? Does payment behavior give a clue to the potential growth of the company?*
- *Based on the CMS Open Payments database, is it possible to detect comparable companies using data analysis and machine learning algorithms? Furthermore, is it profitable to adopt pairs trading among 'similar' companies that we find?*

## 2 Executive Summary

The whole research is about how to learn the behaviors of medical industry participants and select stocks for pairs trading.

First of all, we collect, understand and clean the data. We remove all the irrelevant information from the raw data and encode the context information to numerical values. We do some statistical analysis to get intuition. This inspires us to consider building stock pools for pairs trading.

Secondly, we utilize a customized Relief-based feature extraction model to reduce the number of features. We treat the payments records as time series and combine them with associated daily stock returns.

Thirdly, we consider the information in general payments data and research payments data, and apply Density-Based Spatial Clustering of Applications with Noise method (DBSCAN) to do unsupervised clustering. For each year, we have several stock pools containing candidate stocks for pairs trading.

Lastly, we select stocks from the stock pool and deploy trading strategies to test our results. For stocks in the same pool, we focus on the statistical arbitrage opportunities for each pair. We forecast next day's price spread using ARMA and LSTM, with the backtested strategy showing a 72% winning rate in out-of-sample performance.

# 3 Technical Exposition

## 3.1 Data Understanding

To achieve our goal of identifying companies with similar payment behaviors, selecting pairs and trading accordingly, we focus on the most relevant subset of the provided CMS Open Payments data and transform it for a better use. Also, we collect the stock prices of the target companies, chosen by the proposed pairs selection framework, to deploy our trading strategy and test our results.

### 3.1.1 CMS Open Payments Data

#### 3.1.1.1 Introduction and Motivation

Open Payments is a statutorily-required, national disclosure program that promotes transparency and accountability by making information about the financial relationships between **reporting entities** and **covered recipients** available to the public. Open payments data can be directly downloaded from CMS website[1].

- **Reporting Entities** refers to pharmaceutical and medical device manufacturers and their distributors who are required to report payments and other transfers of value to Open Payments; also referred to as Applicable Manufacturers and Applicable Group Purchasing Organizations (AM/GPOs).
- **Covered Recipients** refers to physicians, teaching hospitals, physician assistants, nurse practitioners, clinical nurse specialists, certified registered nurse anesthetists/anesthesiologists and certified nurse-midwifes receiving payments or other transfers of value from Applicable Manufacturers and/ GPOs.

In other words, CMS Open Payments database covers various kinds of payments made to physicians, teaching hospitals, or non-physician practitioners by industry.

Pharmaceutical companies continuously explore means to identify and reach out to Key Opinion Leaders (KOLs) to market new products and conduct clinical experiments. **Therefore, the payment records of a company contain rich information about the R&D progress, marketing and expansion strategies, business conditions and even the future growth potential of the company**.

The timeline for open payments publication each year is as follows.

- **Initial publication** contains data submitted and attested to by the Submission End Date (usually the end of March), reviewed, edited and re-attested by the Correction End Date (usually the end of May) and then published(usually the end of June).
- **Refresh publication** is a refresh of all records in initial publication. Data published is the latest attested-to data as of the end of December. The final publication date is usually in the early of next year.

**Therefore, the latest refresh publication is of Year 2020, released on Jan.21, 2022 while the latest initial publication is of Year 2021, released on June 30, 2022.**

Though the final version of open payments data is always published more than one year later than the recorded period of time, payments tend to happen when the new products just come out and are in high demand of promotion. It is highly possible that the revenue generated by the promotion will not be fully included in the financial statements until the following year(or even later). **The market may act behind the schedule and receive weak signals about the hidden growth drivers of the company if not looking at the open payments data.**

[1] https://www.cms.gov/OpenPayments

The idea mentioned above is the key driver of our research. To formulate this idea, we actually make following assumptions:

- **Information in payments data, such as R&D progress and marketing strategies, is not (fully) reflected in other public information, such as financial statements and earnings calls that year.** For example, for open payments data of Year 2020, all transactions happened in 2020, yet (partially) unknown to the public as private information in 2020.
- **Stock value is determined by DCF(Discounted Cash Flow) model, which highly relies on market's expectation for future growth.** Under this assumption, once the related information set of the market expands, the expectation may be adjusted accordingly, altering stock value.
- **The market is generally efficient, in the sense that once the stock value changes, the stock price will move toward its underlying value eventually.** This assumption is different from Efficient Market Hypotheses in the following ways. (1) EMH only categorizes information as public and private. However, in reality, not all public information is priced in immediately. One piece of public information is priced in if and only if it is added into the *related information set*, i.e., the market realizes it is related to stock valuation. This makes our assumption much weaker than EMH. (2) EMH focuses on investment performance and determines that the only opportunity investors have to gain higher returns on their investments is through purely speculative investments that pose a substantial risk. But the market may react slowly. It takes time for the market to realize and determine one piece of information is useful or not. Thus there are opportunities out there for investors to react faster than the market and wait the market to catch up. *Our assumption states that the market will always catch up.*
- **Payments data may be one of those data sources that have not been added into the related information set.** *This assumption also serves as our research motivation.*
- **The operating cycle for the companies we study is one calendar year.** This assumption is similar to that of financial statements.

#### 3.1.1.2 Natures of Payment

The natures of payment[2] can be categorized into three main types: general payments, research payments and ownership and investment information.The basic introduction to these data is as follows:

**General payments data.** Payments or other transfers of value made that are not in connection with research agreements or research protocol. General payments may include but are not limited to gifts, meals, honoraria, and travel.

**Research payments data.** Payments or other transfers of value made in connection with a research agreement or research protocol.

**Ownership and investment data.** Information about physicians who have an ownership or investment interest in the reporting entity or who have an immediate family member holding such interest.

The original payments data is exhaustive. As illustrated in Table 4, we further divide data elements into five groups and drop some information irrelevant to our problem, such as descriptions about Principal Investigators in the research payments data. We also drop some columns that contain either too many NANs (provide little information) or too detailed information, such as Name, ID and Address Line of Recipients. Neither of the case are suitable for feature construction. After that, we preprocess the data, treating INF as NAN.

The processed data containing the information of 1896 companies. We can see that almost every company have general payments in processed data and general payments account for a quarter of total payments. Only half of the processed data have research payments but research payments account for over 70% of the total

[2] For more details, see https://www.cms.gov/OpenPayments/Natures-of-Payment

payments. Only a few specific companies have Ownership and Investment. We use Relief-Based Algorithm and DBSCAN Clustering to select the companies and after selecting 59 companies remain. After selecting,all the companies have payments and the selected data preserves the characteristics of the processed data very well.

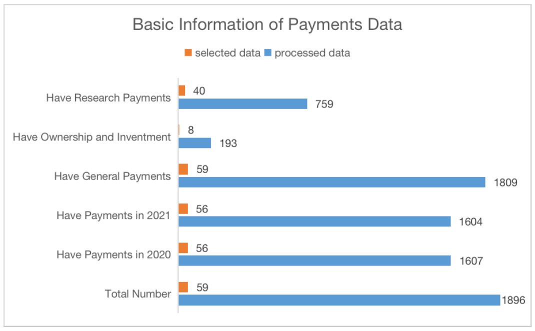


Figure 1: Basic Information of Payments Data

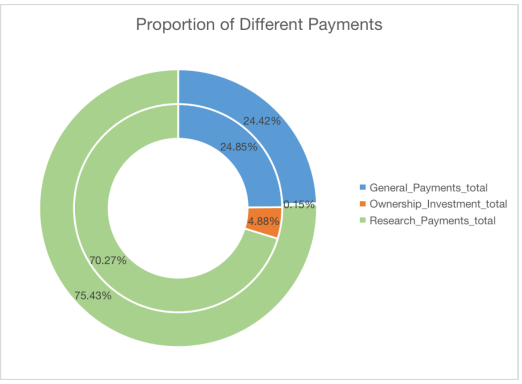


Figure 2: Proportion of Different Payments

### 3.1.2 Stock Prices Data

In order to further implement pairs trading strategy, we need the daily adjusted close prices and daily return rates(calculated from adjusted close prices). All data can be downloaded from Yahoo Finance and WRDS(Wharton Research Data Services).

#### About the Companies

We only focus on those listed on NYSE, NASDAQ and AMEX in the United States. We select 58 stocks of the companies, including 63 parent companies and subsidiaries. The majority of the companies we selected have a primary business in Healthcare (58), while others are primarily engaged in Financial Services (1), Industrials (1) ,Medical Equipment & Devices (2),Pharmaceuticals & Biotech (1) and Technology (2).

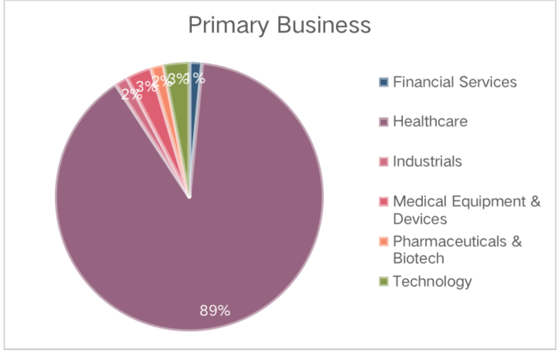


Figure 3: Primary Business of Selected Companies

#### About the Prices

The data we used starts on January 2, 2020 and ends on June 30, 2022,containing 629 data points for 58 stocks. 7 of the stocks had missing data points, meaning that they appeared after January 2, 2020. We calculate the average daily return,the standard deviation of daily return and use sharp ratio as an assessment of return stability.

The distribution of the mean daily return and the top 10 and last 10 stock dailty return are shown in Figure

2. The mean return of the 58 stock is 0.15%,and most of the stocks returns are between -0.03% and 0.15%. The stock MYO and APVO are outliers with positive return 4.53% and 2.41% and the stock DERM and UTRS is outlier with negative return -0.77% and -0.67%.

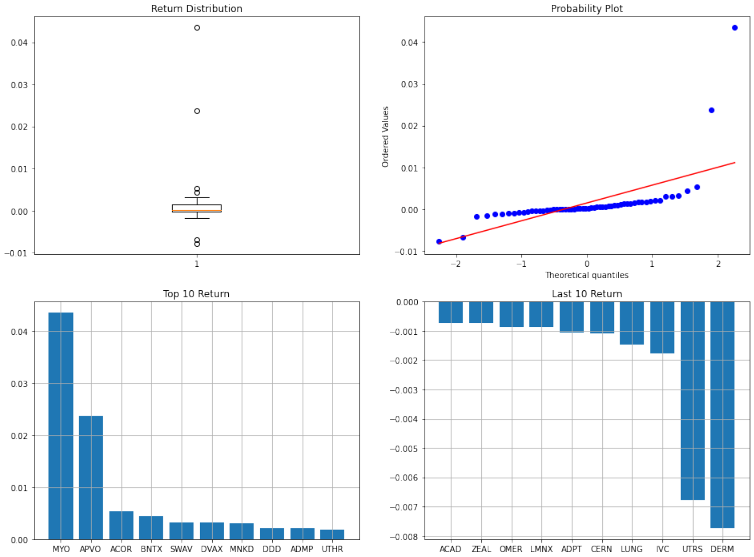


Figure 4: Average Daily Return

The distribution of the mean annual sharp ratio and the top 10 and last 10 stocks annual sharp ratio are shown in Figure 3. The mean annual sharp ratio of the 58 stocks is 0.15, and most of the stocks sharp ratio are between -0.12 and 0.46. The stock UTHR has the highest sharp ratio 1.23 and the stock URTS and DERM is is outlier with negative sharp ratio -2.39 and -1.37.

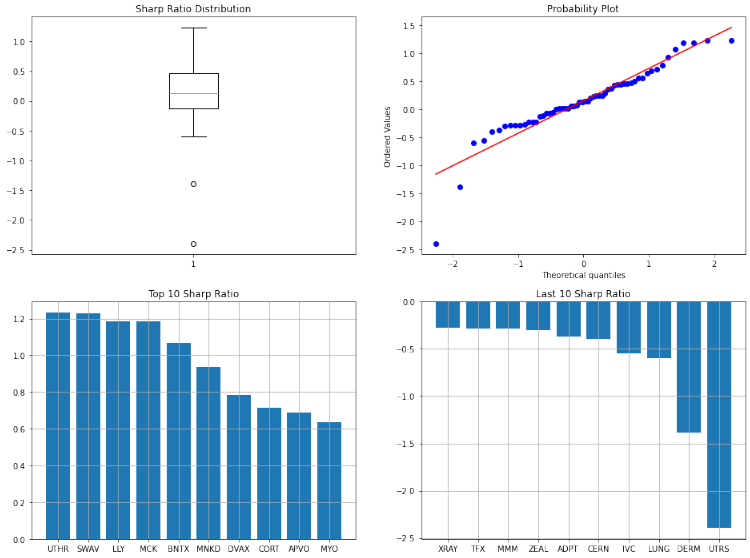


Figure 5: Annual Sharp Ratio

The above analysis shows that if we buy stocks with equal weight and hold them, we can hardly make a significant return. If our pairs trading strategies can make a higher return with higher sharp ratio, we can we can confirm that the strategy works.

## 3.2 Pairs Selection Framework

In this section, we introduce out pairs selection framework for pairs trading. For illustrative purposes, we first apply the time-lagged cross correlation method to assess the similarity of companies within a certain product category. Then we introduce our pairs selection framework.

### 3.2.1 Motivation

To find the companies with similarity, we first focus on the companies with common product categories. Notice that in selecting the target companies, we consider not only the companies with high payment ratio but also the companies with high absolute annual payments.

$$\text{payment ratio} = \frac{\text{annual payments with specific product categories}}{\text{annual payments}}.$$

We set the ratio threshold to be 0.5, but some giant companies might operate with several product lines and each product category only takes a relatively small fraction in total payments. Therefore we add the companies with top 10 total payments amount into the target companies list. The advantage of classifying companies by product categories instead of specific products is mainly the feasibility to group company without looking into the specific products.

We carefully examined the top 10 companies with total general payment and found that there are three companies - Stryker Corporation, Zimmer Biomet Holdings, Inc., DePuy Synthes- focus on products related to spine. In some sense, these three companies are competitive, so we want to explore whether there is some correlation among them.

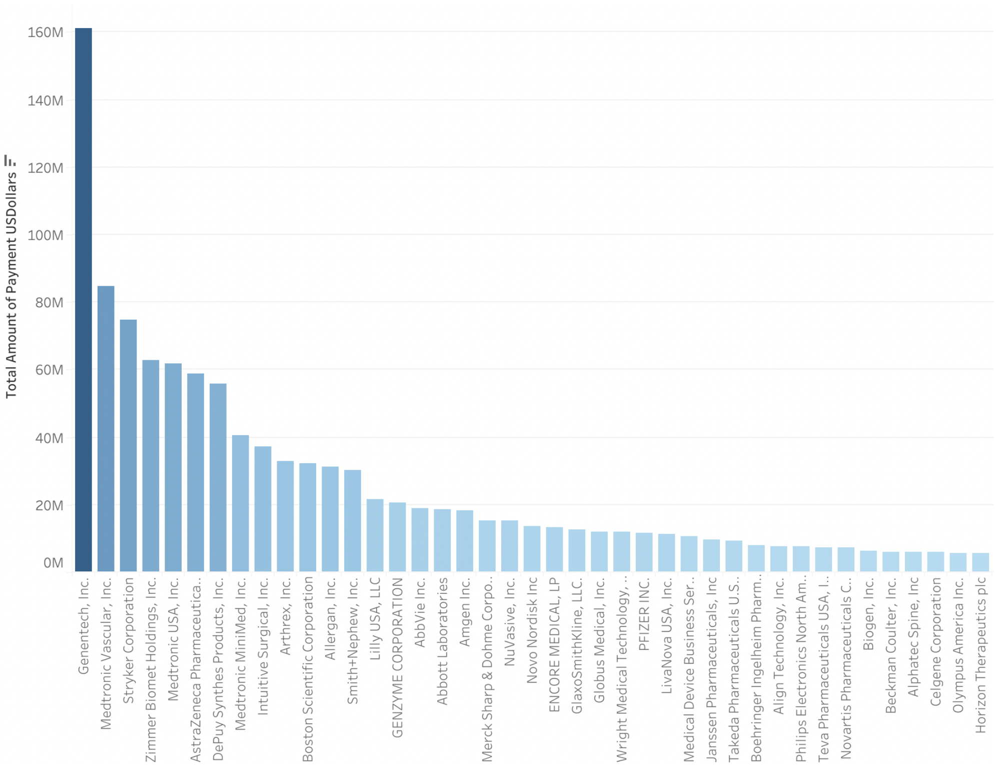


Figure 6: Companies ranked by total general payments

Surprisingly, the weekly time series of the general payment of these three companies are very similar. However, the weekly time series of another company (Pfizr, Inc.) producing different products are of quite different trend. (See Figure 7.) Thus, we would imagine that there is some correlation among the payment behavior of competitive companies. The first problem is how to define the competitive companies. We adopt the following strategies to identify the **competitive companies** on a given category of products:

- if the company's payment on such category of products accounts for more than 50% of the total payment; or
- if the company's payment on such category of products is in the top ten.

Such definition of competitive companies fully reflects the companies' own structure and market power.

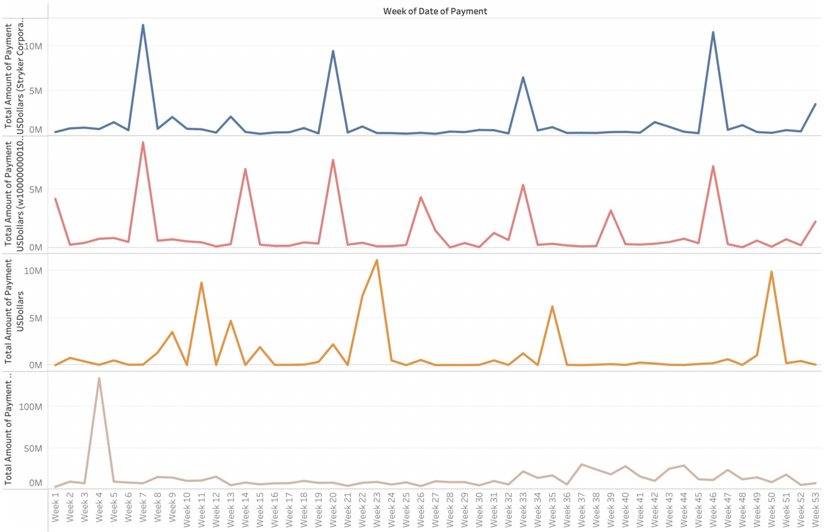


Figure 7: Weekly payments of four companies: Stryker Corporation, Zimmer Biomet Holdings, Inc., DePuy Synthes Products, Inc., and Pfizr, Inc., respectively

Next, we apply the Time-Lagged Cross Correlation (TLCC) [6] to assess the similarity between different companies payment time series. Time-Lagged Cross Covariance and Correlation can be useful to identify the leader-follower relationship between different time series. With TLCC, we can interpret the high payment-behaviors-similarity as the competitors behaviors.

**Timelagged cross correlation measurement** According to Figure 7, the time series of the first three companies are quite similar. We choose time-lagged cross correlation to measure the correlation between weekly total payment time series of similar companies. For two weekly total payment time series $G = (g_1, \cdots, g_l)$ and $H = (h_1, \cdots, h_l)$, considering the shift between $G$ and $H$ along the time axis. We keeps vector $H$ static and makes $G$ slide over $H$ to calculate the correlation coefficient for each shift $s$ of $G$ (where $s \in (k, k)$, $k$ is the maximum step allowed to shift). Different from the method used in Ya Su et al.[8], we adopt the following periodic continuation method to construct $G_s$:

$$G_s = \begin{cases} (\overbrace{g_{l-s+1}, \cdots, g_l}^{|s|}, g_1, \cdots, g_{l-s}) & \text{for } s \geq 0, \\ (g_{1-s}, \cdots, g_l, \underbrace{g_1, \cdots, g_{|s|}}_{|s|}) & \text{for } s < 0. \end{cases}$$

Intuitively, if $s \geq 0$, then $G$ is ahead of $H$; if $s < 0$, then $G$ lags behind $H$. The cross-correlation coefficient of $G_s$ and $H$ can be defined as

$$\rho\left(G_s, H\right) = \frac{\langle G_s, H\rangle}{\sqrt{\langle G, G\rangle \times \langle H, H\rangle}},$$

where $\langle G_s, H \rangle = \sum_{i=1}^{l} G_s[i] \times H[i]$ is the inner product of $G_s$ and $H$. Considering the minimum and maximum of the cross-correlation coefficient $\rho_s$ over $s \in (-k, k)$,

$$\min\rho = \min_s \left(\rho\left(G_s, H\right)\right) \quad \text{and} \quad \max\rho = \max_s \left(\rho\left(G_s, H\right)\right),$$

we define the time-lagged cross correlation of $G$ and $H$ as a 2-tuple TLCC :

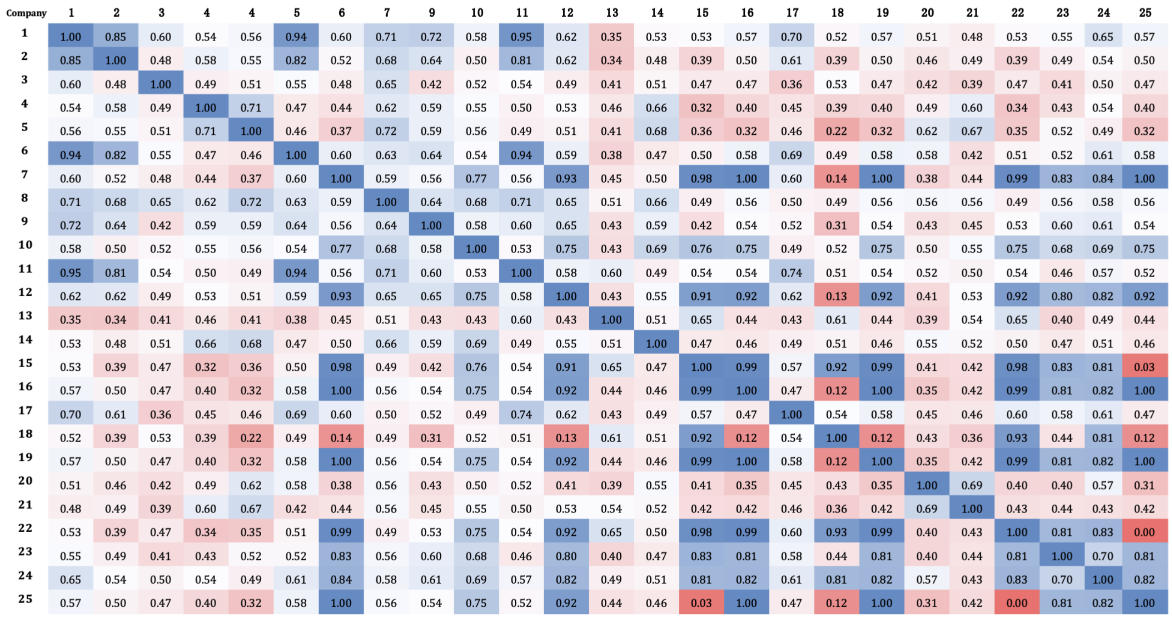
$$\text{TLCC}(G, H) = \begin{cases} (\min\rho, s_1), & \text{for } |\max\rho| < |\min\rho| \\ (\max\rho, s_2), & \text{for } |\max\rho| \geq |\min\rho| \end{cases} \tag{1}$$

where

$$s_1 = \arg\min_s \left\{ |s| : s \in \arg\min_s \left(\rho\left(G_s, H\right)\right) \right\} \quad \text{and} \quad s_2 = \arg\min_s \left\{ |s| : s \in \arg\max_s \left(\rho\left(G_s, H\right)\right) \right\}.$$

It is clear that TLCC $\in [-1, 1]$. The closer to 1 or $-1$ TLCC lies, the stronger the correlation between $G$ and $H$. Moreover, TLCC $> 0$ indicates that $G$ and $H$ move in the same direction.

As an example, we use the above method to analyze the spine industry. The details of the companies in Figure 8 are listed in Figure 9.



| Company | 1 | 2 | 3 | 4 | 4 | 5 | 6 | 7 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.00 | 0.85 | 0.60 | 0.54 | 0.56 | 0.94 | 0.60 | 0.71 | 0.72 | 0.58 | 0.95 | 0.62 | 0.35 | 0.53 | 0.53 | 0.57 | 0.70 | 0.52 | 0.57 | 0.51 | 0.48 | 0.53 | 0.55 | 0.65 | 0.57 |
| 2 | 0.85 | 1.00 | 0.48 | 0.58 | 0.55 | 0.82 | 0.52 | 0.68 | 0.64 | 0.50 | 0.81 | 0.62 | 0.34 | 0.48 | 0.39 | 0.50 | 0.61 | 0.39 | 0.50 | 0.46 | 0.49 | 0.39 | 0.49 | 0.54 | 0.50 |
| 3 | 0.60 | 0.48 | 1.00 | 0.49 | 0.51 | 0.55 | 0.48 | 0.65 | 0.42 | 0.52 | 0.54 | 0.49 | 0.41 | 0.51 | 0.47 | 0.47 | 0.36 | 0.53 | 0.47 | 0.42 | 0.39 | 0.47 | 0.41 | 0.50 | 0.47 |
| 4 | 0.54 | 0.58 | 0.49 | 1.00 | 0.71 | 0.47 | 0.44 | 0.62 | 0.59 | 0.55 | 0.50 | 0.53 | 0.46 | 0.66 | 0.32 | 0.40 | 0.45 | 0.39 | 0.40 | 0.49 | 0.60 | 0.34 | 0.43 | 0.54 | 0.40 |
| 5 | 0.56 | 0.55 | 0.51 | 0.71 | 1.00 | 0.46 | 0.37 | 0.72 | 0.59 | 0.56 | 0.49 | 0.51 | 0.41 | 0.68 | 0.36 | 0.32 | 0.46 | 0.22 | 0.32 | 0.62 | 0.67 | 0.35 | 0.52 | 0.49 | 0.32 |
| 6 | 0.94 | 0.82 | 0.55 | 0.47 | 0.46 | 1.00 | 0.60 | 0.63 | 0.64 | 0.54 | 0.94 | 0.59 | 0.38 | 0.47 | 0.50 | 0.58 | 0.69 | 0.49 | 0.58 | 0.58 | 0.42 | 0.51 | 0.52 | 0.61 | 0.58 |
| 7 | 0.60 | 0.52 | 0.48 | 0.44 | 0.37 | 0.60 | 1.00 | 0.59 | 0.56 | 0.77 | 0.56 | 0.93 | 0.45 | 0.50 | 0.98 | 1.00 | 0.60 | 0.14 | 1.00 | 0.38 | 0.44 | 0.99 | 0.83 | 0.84 | 1.00 |
| 8 | 0.71 | 0.68 | 0.65 | 0.62 | 0.72 | 0.63 | 0.59 | 1.00 | 0.64 | 0.68 | 0.71 | 0.65 | 0.51 | 0.66 | 0.49 | 0.56 | 0.50 | 0.49 | 0.56 | 0.56 | 0.56 | 0.49 | 0.56 | 0.58 | 0.56 |
| 9 | 0.72 | 0.64 | 0.42 | 0.59 | 0.59 | 0.64 | 0.56 | 0.64 | 1.00 | 0.58 | 0.60 | 0.65 | 0.43 | 0.59 | 0.42 | 0.54 | 0.52 | 0.31 | 0.54 | 0.43 | 0.45 | 0.53 | 0.60 | 0.61 | 0.54 |
| 10 | 0.58 | 0.50 | 0.52 | 0.55 | 0.56 | 0.54 | 0.77 | 0.68 | 0.58 | 1.00 | 0.53 | 0.75 | 0.43 | 0.69 | 0.76 | 0.75 | 0.49 | 0.52 | 0.75 | 0.50 | 0.55 | 0.75 | 0.68 | 0.69 | 0.75 |
| 11 | 0.95 | 0.81 | 0.54 | 0.50 | 0.49 | 0.94 | 0.56 | 0.71 | 0.60 | 0.53 | 1.00 | 0.58 | 0.60 | 0.49 | 0.54 | 0.54 | 0.74 | 0.51 | 0.54 | 0.52 | 0.50 | 0.54 | 0.46 | 0.57 | 0.52 |
| 12 | 0.62 | 0.62 | 0.49 | 0.53 | 0.51 | 0.59 | 0.93 | 0.65 | 0.65 | 0.75 | 0.58 | 1.00 | 0.43 | 0.55 | 0.91 | 0.92 | 0.62 | 0.13 | 0.92 | 0.41 | 0.53 | 0.92 | 0.80 | 0.82 | 0.92 |
| 13 | 0.35 | 0.34 | 0.41 | 0.46 | 0.41 | 0.38 | 0.45 | 0.51 | 0.43 | 0.43 | 0.60 | 0.43 | 1.00 | 0.51 | 0.65 | 0.44 | 0.43 | 0.61 | 0.44 | 0.39 | 0.54 | 0.65 | 0.40 | 0.49 | 0.44 |
| 14 | 0.53 | 0.48 | 0.51 | 0.66 | 0.68 | 0.47 | 0.50 | 0.66 | 0.59 | 0.69 | 0.49 | 0.55 | 0.51 | 1.00 | 0.47 | 0.46 | 0.49 | 0.51 | 0.46 | 0.55 | 0.52 | 0.50 | 0.47 | 0.51 | 0.46 |
| 15 | 0.53 | 0.39 | 0.47 | 0.32 | 0.36 | 0.50 | 0.98 | 0.49 | 0.42 | 0.76 | 0.54 | 0.91 | 0.65 | 0.47 | 1.00 | 0.99 | 0.57 | 0.92 | 0.99 | 0.41 | 0.42 | 0.98 | 0.83 | 0.81 | 0.03 |
| 16 | 0.57 | 0.50 | 0.47 | 0.40 | 0.32 | 0.58 | 1.00 | 0.56 | 0.54 | 0.75 | 0.54 | 0.92 | 0.44 | 0.46 | 0.99 | 1.00 | 0.47 | 0.12 | 1.00 | 0.35 | 0.42 | 0.99 | 0.81 | 0.82 | 1.00 |
| 17 | 0.70 | 0.61 | 0.36 | 0.45 | 0.46 | 0.69 | 0.60 | 0.50 | 0.52 | 0.49 | 0.74 | 0.62 | 0.43 | 0.49 | 0.57 | 0.47 | 1.00 | 0.54 | 0.58 | 0.45 | 0.46 | 0.60 | 0.58 | 0.61 | 0.47 |
| 18 | 0.52 | 0.39 | 0.53 | 0.39 | 0.22 | 0.49 | 0.14 | 0.49 | 0.31 | 0.52 | 0.51 | 0.13 | 0.61 | 0.51 | 0.92 | 0.12 | 0.54 | 1.00 | 0.12 | 0.43 | 0.36 | 0.93 | 0.44 | 0.81 | 0.12 |
| 19 | 0.57 | 0.50 | 0.47 | 0.40 | 0.32 | 0.58 | 1.00 | 0.56 | 0.54 | 0.75 | 0.54 | 0.92 | 0.44 | 0.46 | 0.99 | 1.00 | 0.58 | 0.12 | 1.00 | 0.35 | 0.42 | 0.99 | 0.81 | 0.82 | 1.00 |
| 20 | 0.51 | 0.46 | 0.42 | 0.49 | 0.62 | 0.58 | 0.38 | 0.56 | 0.43 | 0.50 | 0.52 | 0.41 | 0.39 | 0.55 | 0.41 | 0.35 | 0.45 | 0.43 | 0.35 | 1.00 | 0.69 | 0.40 | 0.40 | 0.57 | 0.31 |
| 21 | 0.48 | 0.49 | 0.39 | 0.60 | 0.67 | 0.42 | 0.44 | 0.56 | 0.45 | 0.55 | 0.50 | 0.53 | 0.54 | 0.52 | 0.42 | 0.42 | 0.46 | 0.36 | 0.42 | 0.69 | 1.00 | 0.43 | 0.44 | 0.43 | 0.42 |
| 22 | 0.53 | 0.39 | 0.47 | 0.34 | 0.35 | 0.51 | 0.99 | 0.49 | 0.53 | 0.75 | 0.54 | 0.92 | 0.65 | 0.50 | 0.98 | 0.99 | 0.60 | 0.93 | 0.99 | 0.40 | 0.43 | 1.00 | 0.81 | 0.83 | 0.00 |
| 23 | 0.55 | 0.49 | 0.41 | 0.43 | 0.52 | 0.52 | 0.83 | 0.56 | 0.60 | 0.68 | 0.46 | 0.80 | 0.40 | 0.47 | 0.83 | 0.81 | 0.58 | 0.44 | 0.81 | 0.40 | 0.44 | 0.81 | 1.00 | 0.70 | 0.81 |
| 24 | 0.65 | 0.54 | 0.50 | 0.54 | 0.49 | 0.61 | 0.84 | 0.58 | 0.61 | 0.69 | 0.57 | 0.82 | 0.49 | 0.51 | 0.81 | 0.82 | 0.61 | 0.81 | 0.82 | 0.57 | 0.43 | 0.83 | 0.70 | 1.00 | 0.82 |
| 25 | 0.57 | 0.50 | 0.47 | 0.40 | 0.32 | 0.58 | 1.00 | 0.56 | 0.54 | 0.75 | 0.52 | 0.92 | 0.44 | 0.46 | 0.03 | 1.00 | 0.47 | 0.12 | 1.00 | 0.31 | 0.42 | 0.00 | 0.81 | 0.82 | 1.00 |

Figure 8: Correlation coefficients of main participants in spine industry

It can be seen that these competitors are highly correlated. The Time lag in Figure 9 reflects a type of leader-follower relationship among competitive companies.

| Order | Company Name | Time Lag (weeks) | Total Payment (US $) |
|---|---|---|---|
| 1 | Stryker Corporation | 0 | 81566380.09 |
| 2 | Zimmer Biomet Holdings, Inc. | 0 | 67781100.93 |
| 3 | DePuy Synthes Products, Inc. | -4 | 65882538.1 |
| 4 | Smith+Nephew, Inc. | -11 | 39944515.5 |
| 5 | Medical Device Business Services, Inc. | 5 | 27796374.86 |
| 6 | NuVasive, Inc. | -1 | 17029423.5 |
| 7 | Globus Medical, Inc. | 7 | 12390578.4 |
| 8 | Janssen Scientific Affairs, LLC | -6 | 9446188.7 |
| 9 | Medacta USA, Inc. | 1 | 5884921.21 |
| 10 | Ethicon Inc. | 8 | 5155289.97 |
| 11 | SEASPINE ORTHOPEDICS | 5 | 2745063.12 |
| 12 | Aesculap Implant Systems, LLC | -5 | 1850263.8 |
| 13 | The Institute of Musculoskeletal Science and Education | -7 | 1120151.79 |
| 14 | Centinel Spine, LLC | 11 | 1084559.98 |
| 15 | Biogennix, LLC | 2 | 389232.12 |
| 16 | CELTIC BIODEVICES LLC | -11 | 324417.5 |
| 17 | MiRus, LLC | 1 | 286330.75 |
| 18 | Atlas Spine, Inc. | 6 | 169377.64 |
| 19 | IKON SPINE LLC | -7 | 120224.83 |
| 20 | Nutech Spine, Inc. | 11 | 113577.24 |
| 21 | HydroCision, Inc. | -8 | 69589.23 |
| 22 | HT Medical, LLC | 10 | 12765.01 |
| 23 | Wenzel Spine, Inc. | -8 | 7611.36 |
| 24 | Amplify Surgical, Inc. | 6 | 4641.85 |
| 25 | SpineGuard | -12 | 955 |

Figure 9: Details of the corresponding companies in Figure 8. The time lag in row $i$ measures the time lag of company $i$ relative to company $i-1$. The total payment contains both general payment and research payment.

### 3.2.2 Framework Diagram

**Proposed Framework.** We summarize the whole framework of our research in the following Figure 10.

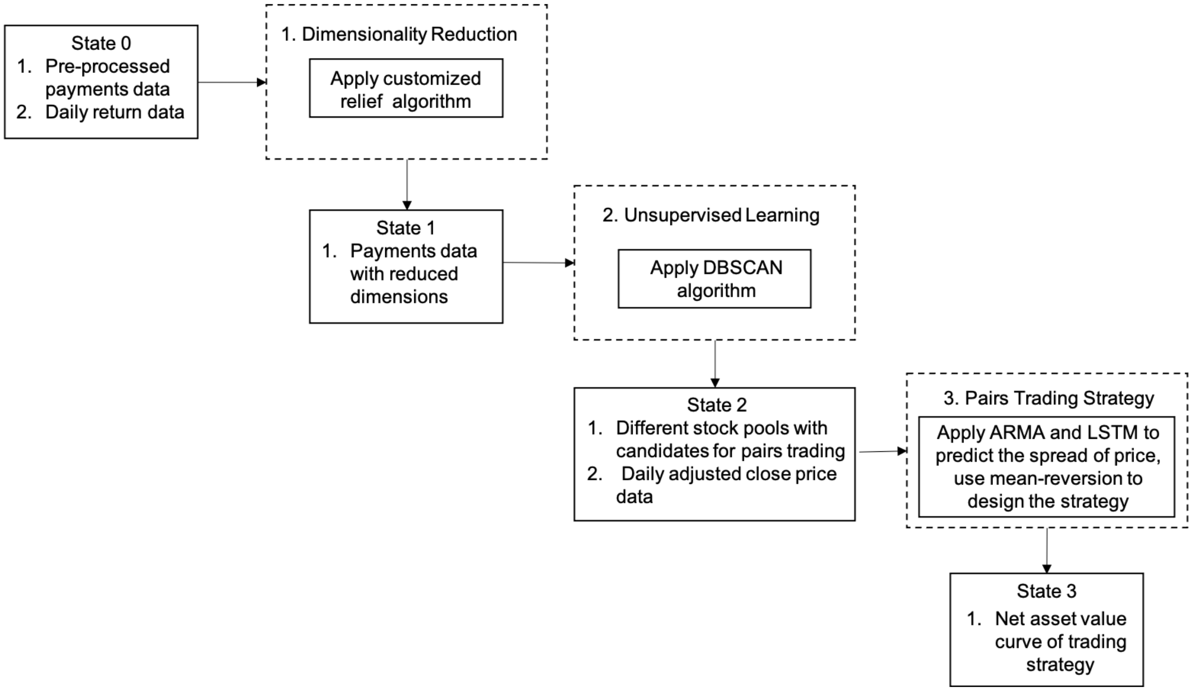

State 0
1. Pre-processed payments data
2. Daily return data
1. Dimensionality Reduction
Apply customized relief algorithm
State 1
1. Payments data with reduced dimensions
2. Unsupervised Learning
Apply DBSCAN algorithm
State 2
1. Different stock pools with candidates for pairs trading
2. Daily adjusted close price data
3. Pairs Trading Strategy
Apply ARMA and LSTM to predict the spread of price, use mean-reversion to design the strategy
State 3
1. Net asset value curve of trading strategy

Figure 10: Proposed Framework

### 3.2.3 Selecting Features with a Relief-Based Algorithm

We want to take a step further to measure the similarity between companies in a more comprehensive way, taking into account the nature of payments and the distribution of recipients' location and speciality, along with product category, payment amount and payment ratio mentioned above. To avoid using the irrelevant information, we utilize a customised Relief-based algorithm to select the top 5 features relevant to the target from General Payments data and Research Payments data, respectively.

**Relief Algorithm Review.** Kira and Rendell [3, 4] formulated the original Relief algorithm in 1992. As an individual evaluation filtering feature selection method, Relief calculates a proxy statistic for each feature that can be used to estimate feature relevance to the target. These feature statistics are referred to as feature weights that can range from -1 (worst) to +1 (best). [9] We can rank the feature weights and select top X features that we want.

**Motivation.** With many strengths that the Relief algorithm has, such as being non-myopic and non-parametric with less time complexity [9], the main reason we turn to it is that Relief considers interactions between features. This feature makes Relief-based algorithm better at dealing with datasets containing highly correlated columns. In our payments datasets, features, especially categorical features, are closely related to each other. For example, 'Type_Ind', 'Product_Category_or_Therapeutic_Area' and 'Product_Name' all provide the product information of the company and are highly related to each other. We choose not to directly drop the information inside these correlated categorical features but take advantage of the special ability of Relief to mine the information and reduce the dimension of the datasets.

**Algorithm.** The original Relief algorithm is restricted to binary classification problems. However, in our case, we need to deal with multi-classification and continuous values at the same time. We thus modify the original Relief algorithm to get a Relief-based algorithm[3] for a better use. Here, we represent the pseudo-code for our algorithm, where the $dif(\cdot)$ gives the absolute distance(L1 norm) with range normalization.

**Algorithm 1: Pseudo-code for the Relief-based algorithm**

```
Require: for each training instance a vector of feature values and the class value
  $n \leftarrow$ number of training instances
  $a \leftarrow$ number of features
  Parameter: $m \leftarrow$ number of random training instances out of $n$ used to update $W$

Initialize all feature weights $W[A] := 0.0$
for $i := 1$ to $m$ do
    randomly select a 'target' instance $R_i$
    find a nearest hit '$H$' and nearest miss '$M$' (instances)
    for $A := 1$ to $a$ do
      $W[A] := W[A] - dif(A, R_i, H)/m + dif(A, R_i, M)/m$
    end for
  end for
return the vector $W$ of feature scores that estimate the quality of features
```

**Dataset Setup.**

- For Features: We aggregate general payments and research payments for each company as features. We need to apply z-score normalization to each feature before the selection process begins. This is because the Relief-based algorithm will find k-nearest neighbours for each feature.

- For Target(Label): For payments data of Year 2020, published on Jan.21, 2022, we use daily return rates in 2021 as target. For payments data of Year 2021, published on June 30, 2022, we use daily return rates from Jan.1, 2022 to June 30, 2022 as target.

[3] For more details: see our code rrelieff.py

The reason why we choose target in this way is based on the last assumption in 3.1.1.1. At the same time, we avoid using price data after payments data publication date.

Also, only general payments and research payments can be organized as time-series data by 'Date_of_Payments' and aligned with daily return rates series afterwards. Therefore, we only use general payments and research payments data to construct features.

**Result Interpretation.** For each year of the data, we first treat the data company-wisely. For each company we have 10 features, 5 from general payments data and 5 from research payments data, from which we want to choose the top 5. Although, generally speaking, 10 is not a large number of features, but we have to consider the following scenarios that one company may only have research or general payments. The relief scores for missing features are always zero.

Then we get a vector of feature scores for each company. After that, we take the equally weighted average among companies to get an overall feature scores for the year. By taking average, if a feature score is very much close to zero, then this indicates that there may be enough companies missing this feature and thus not desirable for the clustering step.

Finally, we choose the top 5 as our final features. The results are as follows.

| | **2020** | **2021** |
|---|---|---|
| **Covered_Recipient_Type_General** | 0.0064 | 0.0020 |
| **Covered_Recipient_Type_Research** | 0.0110 | -0.0025 |
| **Form_of_Payment_General** | 0.0089 | 0.0140 |
| **Form_of_Payment_Research** | 0.0000 | 0.0000 |
| **Location_General** | 0.0085 | 0.0255 |
| **Location_Research** | 0.0048 | -0.0070 |
| **Payments_General** | 0.0359 | 0.0267 |
| **Payments_Research** | 0.0065 | 0.0034 |
| **Type_Ind_General** | -0.0001 | 0.0001 |
| **Type_Ind_Research** | 0.0000 | -0.0002 |

Table 1: Feature scores for each year: results of Relief-based algorithm

### 3.2.4 DBSCAN Clustering and Pairs Selection

Having transformed each time series into a smaller set of features, an unsupervised learning technique can be effectively applied to cluster the records. In our work, we utilize the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) method.

**DBSCAN Review.** Density-Based Spatial Clustering of Applications with Noise (DBSCAN), proposed by Ester et al.[2], is one of the most important algorithms among all density-based clustering methods. DBSCAN interprets the clustering space as an open set in the Euclidean space and, according to the author, can easily detect clusters because there is a typical density of points within each cluster which is considerably higher than outside of the cluster. Comparing to other clustering methods, such as partitioning clustering and hierarchical clustering, density-based clustering has many benefits that make it suitable for our context. [7] First, clusters do not need to have gaussianity assumptions regarding the shape of the data. Secondly, it is naturally robust to outliers. Lastly, it requires no specification of the number of clusters. Therefore we proceed to apply this method to cluster the payments records.

**Implementation and Result Interpretation.** After the feature extraction procedure, we have all the payments data with reduced features. We then utilize DBSCAN algorithm to cluster all the records. We first concatenate all the company-wise data together. Then, for each year, we apply DBSCAN respectively. For both situations, we implement the following settings of the model: $\epsilon = 0.5$, $min\_samples = 2$.

For payments data of Year 2020, the number of estimated number of clusters is 6 and the number of estimated number of noise points is 5. For payments data of Year 2021, the number of estimated number of clusters is 3 and the number of estimated number of noise points is 6.

For stocks in each cluster, they are considered similar to each other by our pairs selection framework. We call each cluster as 'stock pool'. By setting $min\_samples = 2$, we ensure that, in each stock pool, there would be at least one pair of stocks to trade.

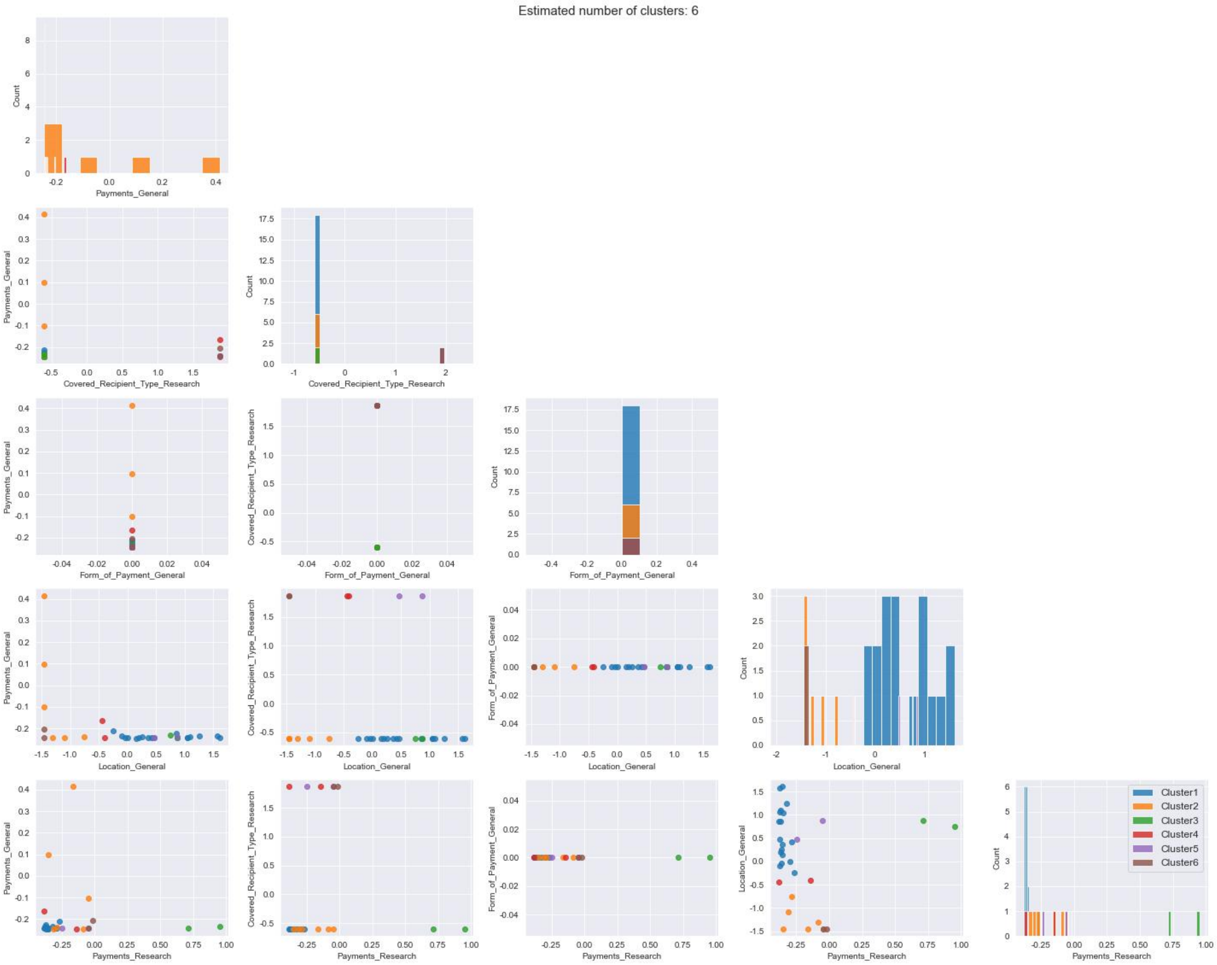


Figure 11: DBSCAN clustering results for 2020: plot matrix for all 2D projections

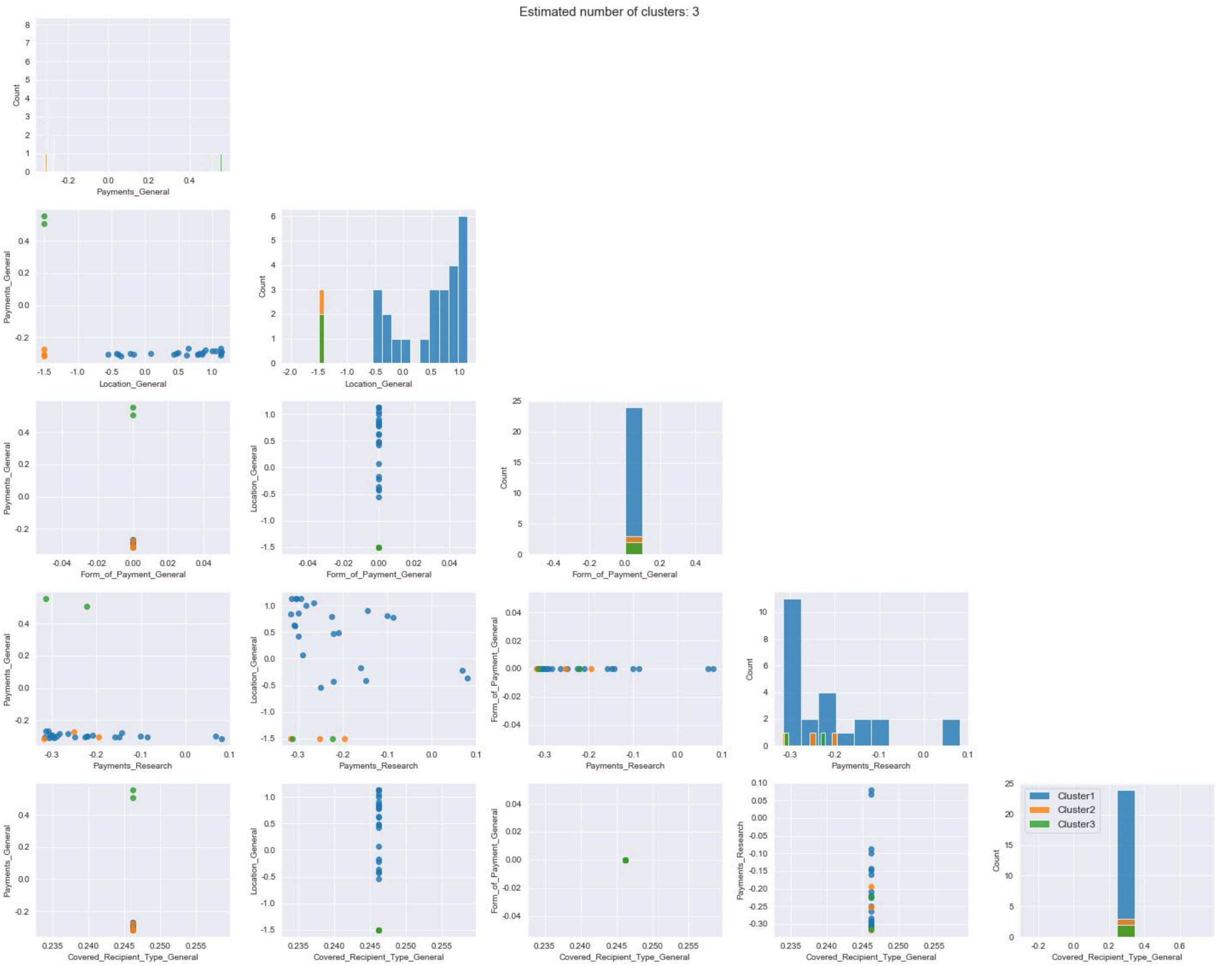


Figure 12: DBSCAN clustering results for 2021: plot matrix for all 2D projections

## 3.3 Pairs Trading

After obtaining the stock pools, we deploy trading strategies to test our results. For stocks in the same pool, we focus on the statistical arbitrage opportunities for each pair.

### 3.3.1 Predicting Model

Time series forecasting models can be divided into two major classes, parametric and non-parametric models. For parametric modelling, we use **autoregressive moving-average(ARMA)**.For non-parametric modelling, deep learning models can capture more complicated data patterns. We choose **Long Short-Term Memory(LSTM)**.

**Autoregressive Moving-average(ARMA) Review.** The ARMA model describes a stationary stochastic process as the composition of two polynomials. The first polynomial, the autoregression (AR), intends to regress the variable at time $t$ on its own lagged values. The second polynomial, the moving average (MA), models the prediction error as a linear combination of lagged error terms and the time series expected value. We proceed to describe in a formal way how each of the polynomials models a time series $X_t$ . The $AR(p)$ model is described as

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t$$

where $p$ is the polynomial order, $\phi_1, \phi_2, ..., \phi_p$ are the model parameters, $c$ is a constant, and $\varepsilon_t$ is a random variable representing white noise. The $MA(q)$ model is described as

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

where $q$ is the polynomial order, $\theta_1, \theta_2, ..., \theta_q$ represent the model parameters and $\mu$ represents the mean value of $X_t$. The variables $\varepsilon_t, \varepsilon_{t-1}, ..., \varepsilon_{t-q}$ correspond to white noise error terms in the corresponding time instants. Finally, the $ARMA(p, q)$ model can be represented as

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

**Long Short-Term Memory(LSTM) Review.**The structure of RNN, from which LSTM derives, is composed by an input layer, in which a sequence of data ($x_1$,...,$x_\tau$) (in general a multivariate time series) enters the network one step at a time, one or more hidden layers for each time step ($h_1$,...,$h_\tau$), and an output layer that is chosen appropriately with respect to the problem. Our study considers a single classification output $y_\tau$ at the end of the sequence, but in general RNN can deal with sequence-to-sequence predictions as well. Thus, for the present case, the RNN is trying to model the following relation:

$$y_\tau = f(x_1, x_2, ..., x_\tau)$$

The key feature of RNN, as opposed to others Artificial Neural Networks, is that the hidden layers have an autoregressive nature, namely

$$h_t = \Phi(h_{t-1}, x_t; \theta)$$

where the vector $\theta$ summarizes the parameters defining the family of functions $\Phi$. For the “vanilla” RNN with one hidden layer and classification output, the model can be summarized by the following system of equations:

$$h_t = tanh(Wh_{t-1} + Ux_t + b^h), \ \ t = 1, 2, ..., \tau$$

$$\hat{y_t} = softmax(Vh_t + b^y)$$

where $(W, U, V, b^h, b^y)$ are the parameters to be calibrated from data, "tanh" is the hidden layer activation function (that is intended to be applied, as usual, component-wise), and

$$softmax(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

is needed to output probabilities in the classification case, effectively reducing to a logit in the binary case. The LSTM model introduces three gated units: a forget gate f, an external input gate $g$ and an output gate $o$, together with another internal state $s$:

$$f_t = \sigma(W^f h_{t-1} + U^f x_t + b^f) \quad t = 1, 2, ..., \tau$$

$$g_t = \sigma(W^g h_{t-1} + U^g x_t + b^g) \quad t = 1, 2, ..., \tau$$

$$o_t = \sigma(W^o h_{t-1} + U^o x_t + b^o) \quad t = 1, 2, ..., \tau$$

$$s_t = f_t \otimes s_{t-1} + g_t \otimes tanh(W^s h_{t-1} + U^s x_t + b^s) \quad t = 1, 2, ..., \tau$$

$$h_t = tanh(s_t) \otimes o_t \quad t = 1, 2, ..., \tau$$

$$\hat{y_t} = softmax(V h_t + b^y)$$

where $\otimes$ denotes the element-wise multiplication operator and $\sigma$ the sigmoid activation function:

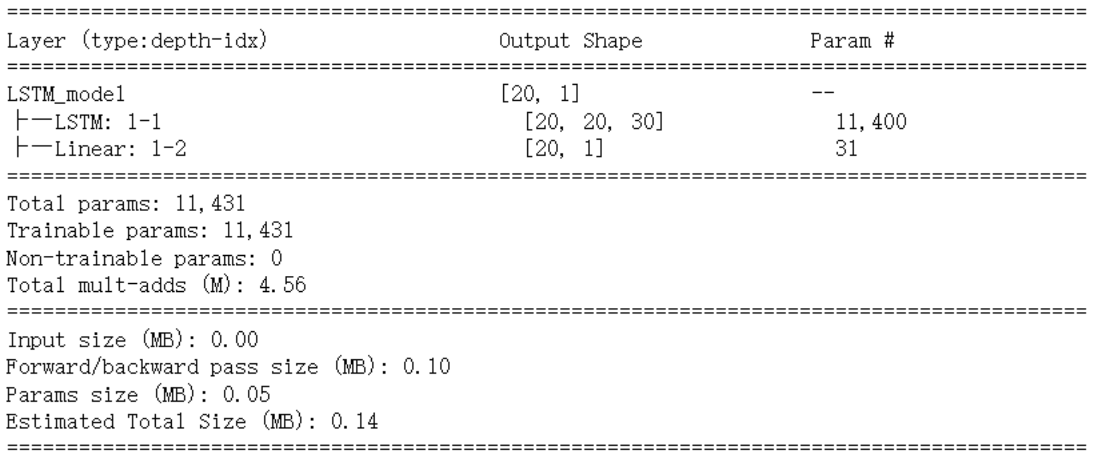
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The idea behind LSTM networks is to use gated units to control the flow of information coming from past states in order to avoid the well known problem of standard RNN in learning from long sequences due to the vanishing/exploding gradient problem. The key point in this respect is the (conditional) selfloop in the internal state, where the information from step $t - 1$ is directly pushed to step $t$, conditionally only on the activation of the forget gate $f$. Specifically, the forget gate $f$ determines how much of the past state $s_{t-1}$ is evolved directly to $s_t$; the external input gate $g$ determines the way in which the past hidden layer contributes to $s$; while the output gate $o$ accounts for the on/off state of the current hidden layer $h_t$. Thus, the information is easily propagated back in time along the sequence from state $s_t$ to $s_{t1}$ through gate $f$, without the issue of 'losing the memory' by successive multiplication of small gradients.

### 3.3.2 Model Training

The total number of observations for each stock is 629. The data will be split into training(377), validation(149) and testing(103). The training set is data from 2020-01-01 to 2021-06-30. The validation set is data from 2021-07-01 to 2022-01-31. The testing set is data from 2022-02-01 to 2022-06-30.

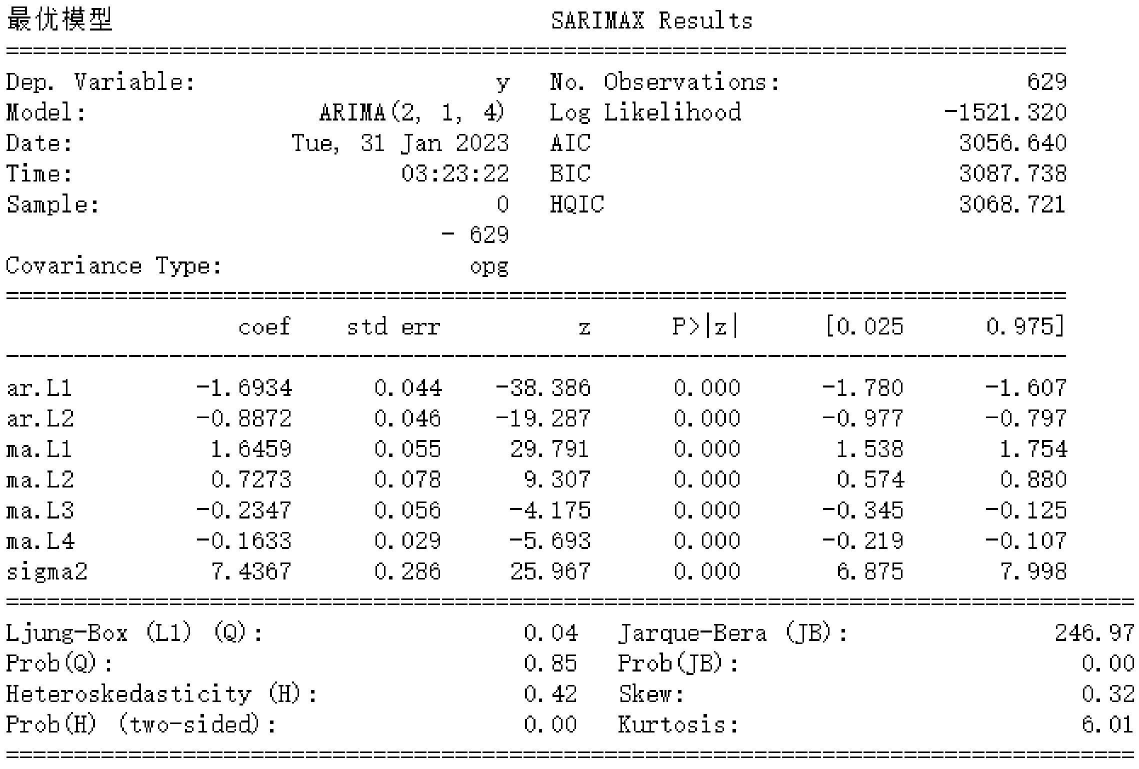
In our LSTM model, we choose hidden size to be 30 with two layers. We choose the "Adam" optimizer because although it's a one-to-one sequence prediction model, we needed a stochastic optimizer to check all possible outcomes and learn the pattern. The batch size is 20. We set learning rate to be 0.001. The Epoch was set to 200 to allow for the inner layer to learn the pattern enough to give an acceptable result. And we use validation data to choose the most suitable Epoch to save our model, during which we can avoid overfitting.



```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
LSTM_model                               [20, 1]                   --
├─LSTM: 1-1                                [20, 20, 30]              11,400
├─Linear: 1-2                              [20, 1]                   31
==========================================================================================
Total params: 11,431
Trainable params: 11,431
Non-trainable params: 0
Total mult-adds (M): 4.56
==========================================================================================
Input size (MB): 0.00
Forward/backward pass size (MB): 0.10
Params size (MB): 0.05
Estimated Total Size (MB): 0.14
==========================================================================================
```

Figure 13: Our LSTM Structure

In our ARIMA model, the parameters $p$ and $q$ both ranges from 0 to 5, and the number of differences, $d$, from 0 to 2. We choose the best parameter combination according to AIC(Akaike information criterion).



```
最优模型                                SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  629
Model:                 ARIMA(2, 1, 4)   Log Likelihood               -1521.320
Date:                Tue, 31 Jan 2023   AIC                           3056.640
Time:                        03:23:22   BIC                           3087.738
Sample:                             0   HQIC                          3068.721
                                - 629
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.6934      0.044    -38.386      0.000      -1.780      -1.607
ar.L2         -0.8872      0.046    -19.287      0.000      -0.977      -0.797
ma.L1          1.6459      0.055     29.791      0.000       1.538       1.754
ma.L2          0.7273      0.078      9.307      0.000       0.574       0.880
ma.L3         -0.2347      0.056     -4.175      0.000      -0.345      -0.125
ma.L4         -0.1633      0.029     -5.693      0.000      -0.219      -0.107
sigma2         7.4367      0.286     25.967      0.000       6.875       7.998
===================================================================================
Ljung-Box (L1) (Q):                   0.04   Jarque-Bera (JB):               246.97
Prob(Q):                              0.85   Prob(JB):                         0.00
Heteroskedasticity (H):               0.42   Skew:                             0.32
Prob(H) (two-sided):                  0.00   Kurtosis:                         6.01
===================================================================================
```

Figure 14: Our ARIMA Structure

### 3.3.3 Trading Strategy

Using prediction model, we can predict the next day's price spread, $S_1$, at the end of every trading day. Also, we can already observe today's price spread, $S$.

On every trading day, we can take a look back at several days ago(a time window), and observe the true price spread series, calculate the mid(50th quantile), down(10th quantile) and up(90th quantile), where 10 and 90 are hyperparameters to be decided in validation set. Also we define $\Delta = (q_{90} - q_{10})/2$.

For every stock pair $A$ and $B$, in trading strategy, there are three positions: 0(no position), 1(A long, B short), -1(A short, B long). At the end of every trading day, according to the size relationship among $S$, $S_1$ and some threshold values above, we may change the position and generate profit or loss in the next day.

This strategy is based on the mean-reversion of price spread. If the spread has exceeded 90th quantile, for example, of history data, we can expect it will start to revert to median. The prediction data $S_1$ can help us to look a step forward. If we notice that $S > high$ but $S_1 > S$, then the spread will probably continue to rise before it reaches the top, so we choose not to take the position immediately. In this way we can earn more from the mean-reversion process, as long as the prediction is accurate.

**Algorithm 2: Trading Strategy**

```
if position == 0:
  if $S > up$ and $S_1 < S$:
    sell A and buy B
    position = -1
  if $S < down$ and $S_1 > S$:
    buy A and sell B
    position = 1
if position == -1:
  if $S_1 - S > \Delta$:
    close the position, then buy A and sell B
    position = 1
  if $S < mid$ and $S_1 > S$:
    close the position
    position = 0
if position == 1:    if $S - S_1 > \Delta$:
    close the position, then sell A and buy B
    position = -1
  if $S > mid$ and $S_1 < S$:
    close the position
    position = 0
```

### 3.3.4 Prediction Evaluation

Before generating trading signals, we first estimate our prediction result for price spread. We choose two performance indicators: Pearson's Correlation and Spearman's Correlation. Because Spearman's Correlation is the correlation of sorting, if our strategy has a high Spearman's Correlation, then we can accurately find the time point of large price spread and the time point of small price spread, which means we can choose correct trading signals for pairs trading.

| | **Pearson's Correlation** | **Spearman's Correlation** |
|---|---|---|
| **LSTM_Train** | 0.9830 | 0.9746 |
| **LSTM_Validation** | 0.9297 | 0.9192 |
| **LSTM_Test** | 0.9254 | 0.9021 |
| **ARMA_Train** | 0.9544 | 0.9498 |
| **ARMA_Validation** | 0.9061 | 0.9030 |
| **ARMA_Test** | 0.8979 | 0.9015 |

Table 2: Evaluation of Price Spread Prediction

For LSTM: Pearson's Correlation and Spearman's Correlation of training data are beyond 95%; two correlations for both validation data and testing data are beyond 90%.

For ARMA: Pearson's Correlation and Spearman's Correlation of training data are approximately 95%; two correlations for both validation data and testing data are nearly beyond 90%.

The Correlation values of LSTM is larger than ARMA in all datasets, this is because LSTM is a more complicated model and probably has a better prediction ability.

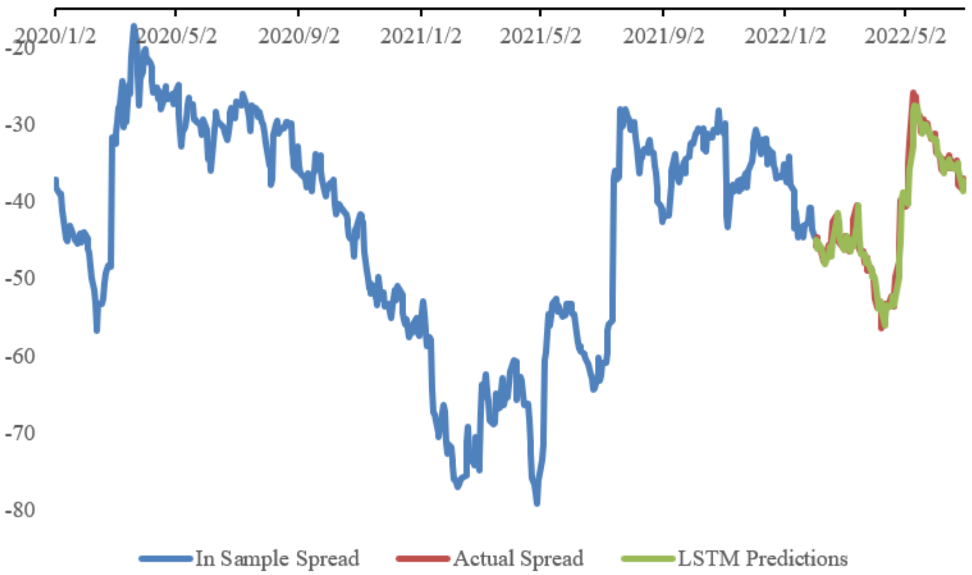


Figure 15: LSTM Prediction for AXGN-GKOS Spread

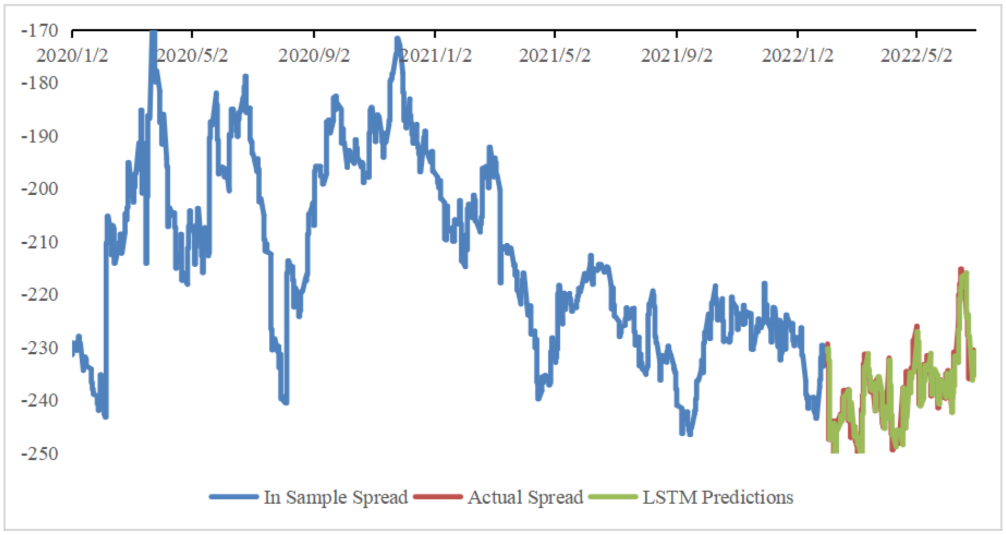


Figure 16: ARMA Prediction for ACAD-BDX Spread

### 3.3.5 Backtesting Results

Having trained the models, we use daily stock prices from 07/01/2021 to 01/31/2022 to do the in-sample test and those after publication date(02/01/2022-06/30/2022) to do the out-of-sample test.

Based on the output of clustering algorithm, with newly listed and delisted stocks removed, we have 150 pairs in total. We apply the trading strategy based on LSTM/ARMA to every pair and get 300 net value curves.

**In-sample Results.** Before the out-of-sample test, we need to choose the two hyperparameters, rolling window and quantile threshold, in the strategy. We set the selection set: window in 30, 45, 60 and quantile in 0.10, 0.15, 0.20, 0.25 and choose a parameter tuple to maximize the mean annualized return and Sharpe ratio of the 300 curves. The evaluation outcome is as follows:

| | **annualized return** | **Sharpe ratio** |
|---|---|---|
| **30, 0.10** | 0.2912 | 0.6284 |
| **30, 0.15** | 0.3414 | 0.7717 |
| **30, 0.20** | 0.5017 | 0.9054 |
| **30, 0.25** | 0.7286 | 1.1894 |
| **45, 0.10** | 0.1610 | 0.5624 |
| **45, 0.15** | 0.2479 | 0.5732 |
| **45, 0.20** | 0.3565 | 0.7559 |
| **45, 0.25** | 0.4706 | 0.9719 |
| **60, 0.10** | 0.0893 | 0.3673 |
| **60, 0.15** | 0.1489 | 0.4658 |
| **60, 0.20** | 0.2003 | 0.5598 |
| **60, 0.25** | 0.3560 | 0.7122 |

Table 3: Strategy Performance of Different Parameter Combinations

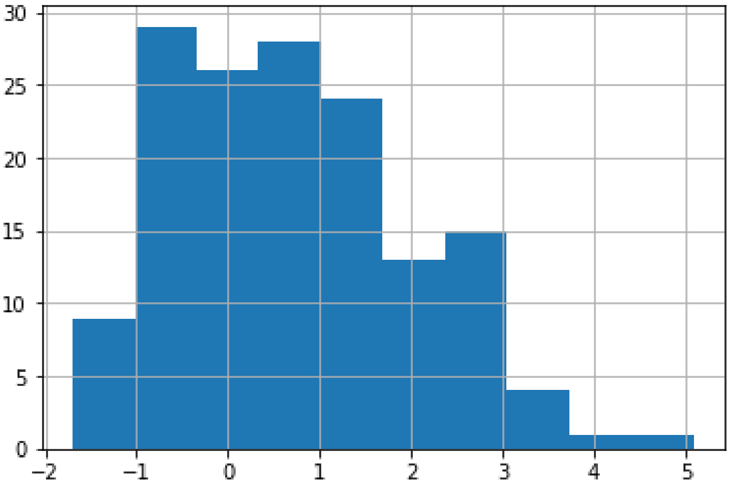


Figure 17: Hist of in-sample Sharpe ratio

**Out-of-sample Results.** For the best Parameter Combinations (30, 0.25), we provide the figure illustration of the out-of-sample test result. The overall performance of our out-of-sample results can be summarized as follows. **The best Sharpe is approximately 5.08, with winning rate[4] 92.7%, and the worst Sharpe is -1.70, while the average Sharpe is 0.82. The winning rate of our framework[5] is 73.5%.**

Between the two prediction models, LSTM is strongly better than ARMA, the average Sharpe of LSTM is 1.25 while that of ARMA is only 0.39.

Here are the performance of our strategy based on LSTM and ARMA. In each figure, *best* and *worst* are the pairs with highest/lowest return and *mean* is equal weight combination of all the 150 pairs, which can represent the performance of strategy and prediction model because it is an investment portfolio based on all stocks.

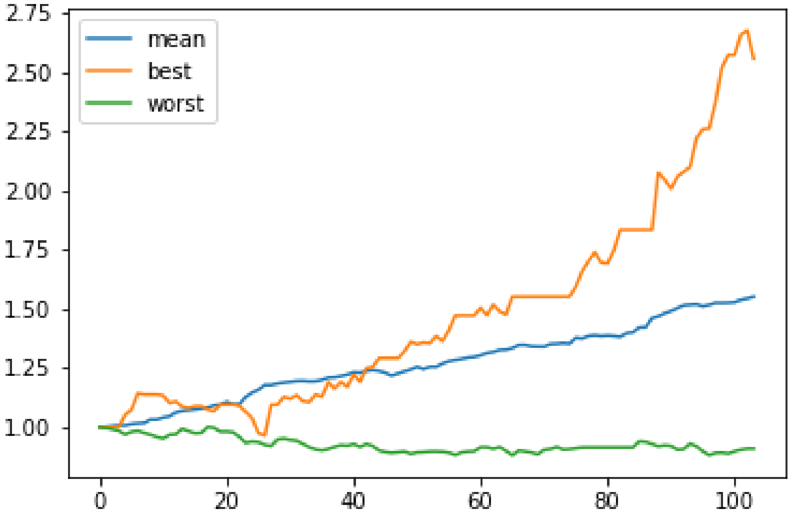


Figure 18: Out-of-sample net asset value curve with LSTM prediction

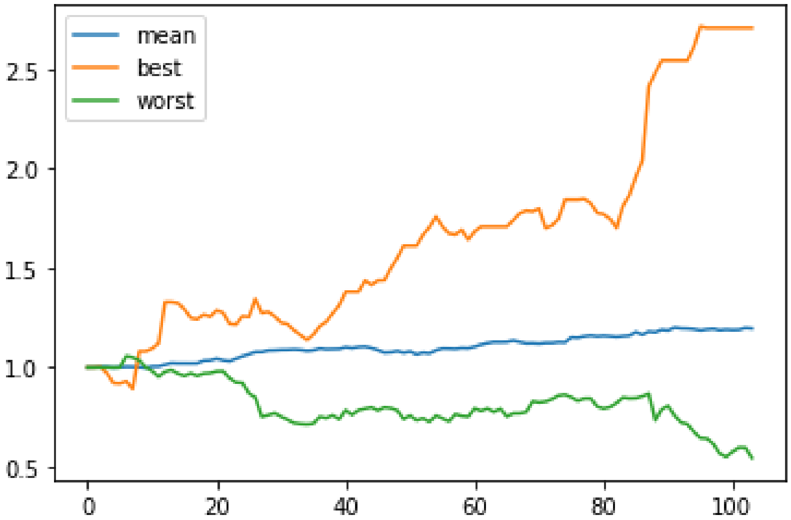


Figure 19: Out-of-sample net asset value curve with ARMA prediction

[4] The winning rate of a strategy is defined as $\frac{count(pnl>0)}{count(pnl\neq 0)}$

[5] The winning rate of our framework is defined as $\frac{\text{number of profitable pairs}}{\text{number of all pairs}}$

**Discussion on Backtesting Results.** We find interesting facts through comparing in-sample results with out-of-sample results.

- Firstly, the out-of-sample results are better than the in-sample result, with not only higher best Sharpe, but a better average performance as well.
- Secondly, the in-sample test is conducted within the observation period of the payments while the out-of-sample test is conducted after the publication date, more than one year later than the in-sample time window.

These facts give us some valuable insights into the market and also strongly defend our investment logic.

- Firstly, the comparison results provide strong arguments about the fact that **the payments data contains business information uncovered by financial statements or any other public news in the reported year.** This is because our arbitrage strategy have 100% winning rate in the in-sample test.
- Secondly, **the open payments data can be a rich source of detecting comparability between companies**, for the simple reason that our pairs trading strategy yields much higher Sharpe than the market both in in-sample test and out-of-sample test.
- Thirdly, **the market has not realized that the statistical arbitrage opportunities come from the Open Payments Database.** Our strategy provides even better performance after the data is published.
- Finally, one reason that our out-of-sample test results are better than those of in-sample test is that **the indicating power of the payments data may delay in time**.

### 3.4 Drawbacks and Future Work

Firstly, many of the reporting entities are not publicly listed at all, listed in other countries or listed in small exchanges in the U.S.. This phenomenon reduces data accessibility thus the size of our stock pools. In the future, we could at least manage to cover all listed companies in the U.S.. Next, we could include more companies listed in other markets after carefully examining the possibility for pairs trading across markets.

Secondly, we only apply DBSCAN clustering method, without heavy parameter tuning to avoid overfitting, for our application. Chances are that other clustering methods, without heavy tuning as well, perform better than DBSCAN in our case. In the following work, we could consider apply more clustering methods, validate among them and choose the best model.

Thirdly, in the trading strategy, in order to simplify our model and highlight the focus of our study, we assume the training process in computer can be done immediately so we can change the positions at the end of trading day, but in fact it takes time. Further work will include the transaction slip point to model the difference, and algorithm acceleration is also a valuable research field.

## 4 Conclusions

In this work, we mine the CMS Open Payments Data to learn the behaviors of Medical industry participants. We propose a pairs selection framework and a pairs trading strategy. We also build a backtesting platform for evaluation. Finally, we get the best Sharpe ratio 5.08 from LSTM prediction, with winning rate 92.7%. 73.5% of our selected pairs make profits in the out-of-sample backtesting.

# References

[1] R Anand et al. "Leveraging CMS open payments data to identify channel preferences and gather competitive intelligence, thereby improving HCP targeting". In: *Journal of the Pharmaceutical Management Science Association* (2017), pp. 59–65.

[2] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[3] Rendell Kira. "A practical approach to feature selection." In: *Proceedings of the ninth international workshop on Machine learning*. 1992, pp. 249–256.

[4] Rendell Kira. "The feature selection problem: Traditional methods and a new algorithm". In: *AAAI. Vol. 2*. 1992, pp. 129–134.

[5] Deborah C Marshall, Madeleine E Jackson, and Jona A Hattangadi-Gluth. "Disclosure of industry payments to physicians: an epidemiologic analysis of early data from the open payments program". In: *Mayo Clinic Proceedings*. Vol. 91. 1. Elsevier. 2016, pp. 84–96.

[6] Chenhua Shen. "Analysis of detrended time-lagged cross-correlation between two nonstationary time series". In: *Physics Letters A* 379.7 (2015), pp. 680–687.

[7] Nuno Horta Simão Moraes Sarmento. *A Machine Learning based Pairs Trading Investment Strategy. SpringerBriefs in Applied Sciences and Technology*. Springer, 2021.

[8] Ya Su et al. "CoFlux: robustly correlating KPIs by fluctuations for service troubleshooting". In: *Proceedings of the International Symposium on Quality of Service*. 2019, pp. 1–10.

[9] Ryan J Urbanowicz et al. "Relief-based feature selection: Introduction and review". In: *Journal of biomedical informatics* 85 (2018), pp. 189–203.

[10] Stephen Zuckerman, Laura Skopec, and Joshua Aarons. "Medicaid Physician Fees Remained Substantially Below Fees Paid By Medicare In 2019: Study compares Medicaid physicians fees to Medicare physician fees." In: *Health Affairs* 40.2 (2021), pp. 343–348.

# 5 Appendix A

In this section, we list which columns we use and which columns we remove in our research. The variables can be found in following table. This works for both 2020 and 2021 data.

| | General Payments | Research Payments | Ownership and Investment |
|---|---|---|---|
| **Recipient Information** | Recipient_Postal_Code/Zip_Code | Recipient_Postal_Code/Zip_Code | Recipient_Postal_Code/Zip_Code |
| | Physician_Primary_Type/Specialty/Ownership_Ind | Physician_Primary_Type/Specialty/Ownership_Ind | Physician_Primary_Type/Specialty |
| | Teaching_Hospital_ID/CNN/Name | Teaching_Hospital_ID/CNN/Name | Physician_ID/Name |
| | Physician_ID/Name | Physician_ID/Name | Recipient Address/City/Province/State/Country |
| | Recipient Address/City/Province/State/Country | Recipient Address/City/Province/State/Country | |
| | Physician_License_State_code | Physician_License_State_code/Noncovered_Recipient_Entity | |
| **Company Information** | Manufacturer_or_GPO_ID/Name | Manufacturer_or_GPO_ID/Name | Manufacturer_or_GPO_ID/Name |
| | Manufacturer_or_GPO_State/Country | Manufacturer_or_GPO_State/Country | Manufacturer_or_GPO_State/Country |
| **Payment Information** | Total_Amount_of_Payment/Number_of_Payments | Total_Amount_of_Payment/Number_of_Payments | Total_Amount_Invested |
| | Date_of_Payment | Date_of_Payment | Value_of_Interest |
| | Form/Nature_of_Payment | Form_of_Payment | Terms_of_Interest |
| | Third_party_Payment_Recipient_Ind/Name/Charity_Ind/<br>Third_Party_Equals_Covered_Recipient_Ind | Expenditure_Category | Interest_Held_by_Physician_or_Family |
| **Product Information** | Related_Product_Ind/Covered_or_Noncovered_Ind | Related_Product_Ind/Covered_or_Noncovered_Ind | |
| | Type_Ind/Product_Category_or_Therapeutic_Area/Product_Name | Type_Ind/Product_Category_or_Therapeutic_Area/Product_Name | |
| | Associated_Drug_or_Biological_NDC | Associated_Drug_or_Biological_NDC/Name_of_Study | |
| **Others** | Payment_Publication_Date/Delay_in_Publication_Ind/Record_ID | Payment_Publication_Date/Record_ID | Payment_Publication_Date/Record_ID |
| | Contextual_Information | Change_Type | Change_Type |
| | City/State/Country_of_Travel | ClinicalTrials_Gov_Identifier | |
| | Change_Type | Research_Information_Link/Context_of_Research | |
| | | Principal_Investigators_ID/Type/Specialty/License_State_code | |

Table 4: Description of dataset, texts in blue are the information that we put in our dataset while texts in black are those we discard.