Spread prediction between stocks using time series algorithms with Implementation in Quantitative Trading Strategy

Aayush Patel[1-a], **Karson Cheng**[1-a], **Ren Gong**[1-a], **Junzhen Wang**[1-a], **and Sanketh Sarathy**[1-a]

[a]Rutgers University

In this we have tried to implement pairs trading strategy using time series algorithms like Auto Regressive Inegrated Moving Average (ARIMA) and Dynamic Mode Decomposition (DMD). The usage of algorithms to predict the spread between two integrated pairs and use them in trading strategy to exploit it for financial gain. Apart from that we have also used Bayesian optimization technique for large scale parameter optimization to find optimal parameters for trading strategy.

Quantitaive Finance | Pairs Trading | Time Series Analysis | ARIMA | Dynamic Mode Decomposition | Bayesian Optimization

1. Intoduction

Our project aims to address the complex and dynamic issue of predicting spreads in the financial markets. Accurate spread prediction is crucial for market participants, as it influences decision-making processes, risk management, and profitability. Recognizing the challenges presented by financial market volatilities and the need for advanced predictive capabilities, our approach seeks to harness the power of time series based algorithms for accuracy and reliability.

Quantitative investment strategies are used to exploit the financial markets that are not in equilibrium. Specifically, in this project we are building pairs trading strategy based on some advance level time-series algorithms to predict spread in advance so that we can increase the performance of trades.

To tackle this business problem, we will utilize a comprehensive dataset that encompasses a wide range of market indicators, including price levels, trading volumes, and historical spread data. This dataset has been meticulously curated to ensure that it captures the nuances and intricacies of market behaviors, providing a solid foundation for our analysis.

By employing statistical techniques and exploratory data analysis, we will identify several key features that show potential predictive value. Also different attempts will be made to use different time series algorithms ranging from classical methods to most recent frameworks which includes ARIMA & Dynamic Mode Decomposition. The algorithms will be applied to build a pairs trading strategy and back-test them using previous years stocks data.

2. Dataset

We retrieve the dataset from the Yahoo Finance API. The timeframe for our dataset is per day, spanning from March 29, 2017, to March 29, 2024.

A separate custom class named "YahooDataSource" has been created for data collection. Since the source data was not in a tabular format, we employed a pivoting methodology to transform it into a tabular format. This class is capable of handling large datasets containing different tickers and offers filtering options to retrieve data such as high, low, close, volume, etc.

Furthermore, it includes specific methods to perform data pre processing. This preprocessing is necessary as pairs trading demands that the data of the two securities be consistent. By consistency, we mean that the date and time of every feed of both securities should match exactly, and it also involves handling null values.

The original dataset comprises 1762 data feeds of close prices, which are utilized for the analysis and implementation of trading strategies. The detailed breakdown of the data will be discussed in later sections. It encompasses data from approximately 70 tickers to identify optimal pairs of co integrated assets.

3. Model Development

As outlined in (1), constructing a pairs trading algorithm involves three key phases: pair selection, spread modeling, and the development of trading rules.

Pair selection is the initial step, focusing on identifying assets that exhibit co-movement in their returns and possess a mean-reverting spread. This process is critical for establishing a foundation of assets that can potentially generate profits through the pairs trading strategy.

Spread modeling follows, aiming to replicate the spreads observed in the selected pairs in a manner that optimizes profit potential while maintaining market neutrality. This phase involves intricate simulations and analysis to ensure that the modeled spreads align with the expected behavior of the market.

Finally, trading rules development comes into play. In addition to defining entry and liquidation rules, this phase involves establishing supplementary regulations to manage short-term losses effectively. These additional rules serve to enhance the robustness of the trading strategy and mitigate risk during volatile market conditions.

Here are the brief of steps that used for implementing Pairs trading strategy for our case.

- **Step 1:** Extracting the Data form of some predefined financial instruments or asset classes to form pairs for trading
- **Step 2:** Conducting co-integration tests to assess the long term relationship between the selected pairs of financial instruments.

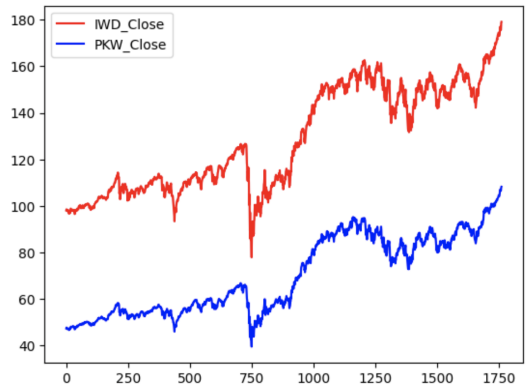

Fig. 1. Close prices of historical time series data of IWD and PKW tickers

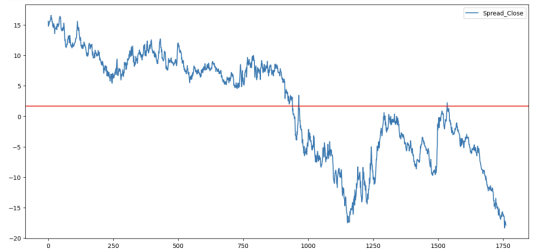

Fig. 2. Residual Spread between IWD and PKW tickers of OLS regression

- **Step 3:** Calculating the cointegration coefficient to quantify the strength of the long term relationship between the selected pairs of financial instruments.
- **Step 4:** Implementing ARIMA and DMD models to forecast the spread between the co-integrated pairs of financial instruments.
- **Step 5:** Hyperparameter tuning for the trading strategy.
- **Step 6:** Implement the strategy with Optimal Hyper Parameters

A. Pair Selection. In the domain of pairs trading, the pivotal determinant of success resides in the identification of co-moving pairs. Irrespective of the intricacy of spread modeling and trading regulations, the effectiveness and profitability of the algorithm are contingent upon the quality of these pairs.

Within our project, the pair selection process employs co-integration tests for each combination of selected tickers, filtering them based on p-values below 0.05, indicative of co-integration between pairs. Consequently, the algorithm identifies the most favorable pair from this subset by selecting the pair with the lowest p-value. Hence, from the pool of provided stocks, we have identified "IWD" and "PKW" as the optimal pair for analysis.

After finding optimal pair, ordinary least square regression is performed to identify the relation between pairs. The results of the linear regression coefficient and p-values of co integrated p-values are shown in the table 1

The plot of co integrated pair is shown in the figure 1. It seems that two pairs behave well by looking at the historical data. Also, the spread between two tickers are shown in the figure 2

Table 1. Statistics of co-integrated pairs

Tickers	IWD	PKW	p-value
coefficient	1.0	-1.807	0.00220

Table 2. Augmented Dickey–Fuller test for residual spread with one order differencing

ADF test	adf-stats	p-value
residual (1-order seasonal differencing)	-43.978	0.000005

B. Spread Modeling. Now we investigate the residual term from the above spread model. We will use the Ornstein-Uhlenbeck (2) process to model the residual term because the O-U process is a stochastic process such that the object modeled by the process always drifts towards its long-term mean. The residual term, namely the spread, has a very similar property according to the assumption of pairs trading. The residual term X_t from the above spread model satisfies the following stochastic differential equation:

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t \quad [1]$$

where θ, μ, and σ are the parameters we want to estimate later using linear regression. W_t denotes the Wiener process, which suggests that the probability distribution of W_t is a normal distribution with mean 0 and variance t.

By integrating Equation (2), we have

$$X_{t+1} = a + bX_t + \epsilon_{t+1} \quad [2]$$

$$a = (1 - e^{-\theta\Delta t})\mu \quad [3]$$

$$b = e^{-\theta\Delta t} \quad [4]$$

$$\mathrm{Var}(\epsilon) = \frac{\sigma^2}{1 - e^{-2\theta\Delta t}}/(2\theta) \quad [5]$$

In summary, future prediction of spread is given by above fundamental equations. So to predict spread, two time series forecasting methods namely ARIMA and DMD are used.

B.1. ARIMA. ARIMA (Auto Regressive Integrated Moving Average) model (3) is a widely used method for time series forecasting. It combines auto regressive, differencing, and moving average components to model and predict future values based on past observations.

To identify the parameters for spread predictions, we begin by creating the auto-correlation and partial-auto correlation plots of the given spreads, as depicted in the accompanying figure 3 and 4. Upon examination, it becomes evident that the correlation plot exhibits a lack of decay, signaling the necessity for differencing the series. Following this adjustment, both the autocorrelation and partial autocorrelation plots now demonstrate tailing at the first-order lag, indicating a potential fit as seen in figures 5 and 6

To further assess the adequacy of our model, we employ the augmented Dickey–Fuller test (ADF), a statistical tool utilized to ascertain whether a given process is stationary. The results of this analysis are presented in the subsequent table 2, providing crucial insights into the stationary nature of the process under consideration.

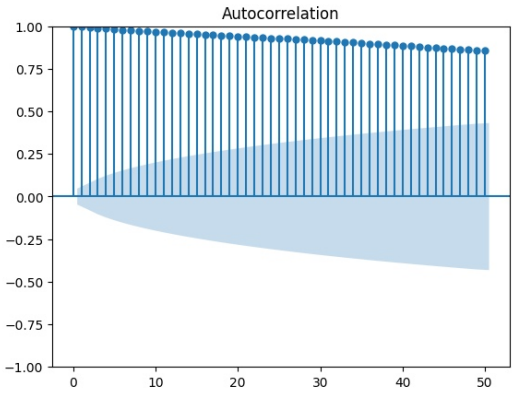

Fig. 3. Autocorrelation plot of residual spread data

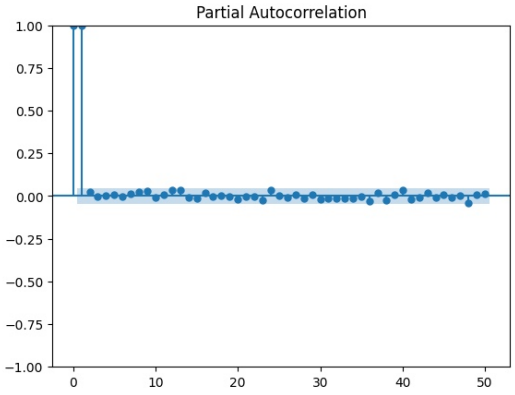

Fig. 4. Partial Autocorrelation plot of residual spread data

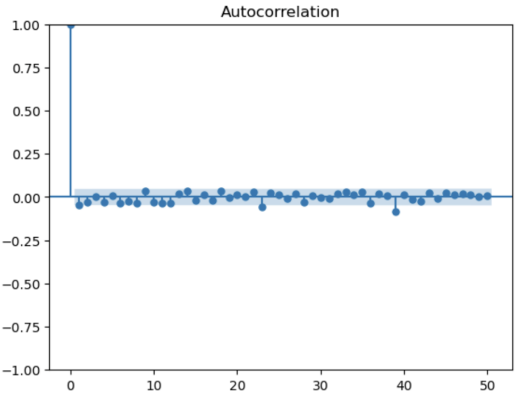

Fig. 5. Autocorrelation plot of residual spread data with one order seasonal differencing

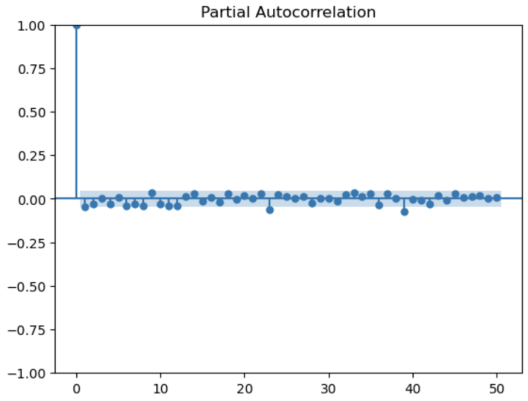

Fig. 6. Partial Autocorrelation plot of residual spread data with one order seasonal differencing

Table 3. Comparison of Mean Squared Error of two methods

Type	ARIMA	DMD
Mean Squared Error	0.6579	1.5353

B.2. Dynamic Mode Decomposition (DMD). This method is basically model order reduction technique for high dimensional nonlinear dynamical system (4). It can able to identify spatio-temporal coherent structures effectively into best fit linear approximation. This DMD method has lots of extension available for improving robustness of the algorithm.

Here, we used Higher Order Dynamic Mode Decomposition (HODMD) (5) which is specifically catered for low dimensional time series predictions which using time delay embedding to transfer analysis algorithm to predictive one.

There are certain hyper parameters for the DMD the most crucial one is SVD rank, which specifies the number of eigen modes to reconstruct the decomposed data and used it future prediction. We have used the PyDMD library to facilitate the DMD for our case and we set SVD rank to zero which automatically identifies the eigen modes.

B.3. Predictive Accuracies. To check the above models performance we have used two metrics to identify its applicability. One of the metric was to calculate the mean squared error between the predicted forecast values and actual forecast values. The results of that metric is given in the following table 3.

Another performance measure was to identify the directional accuracy of two models. The accuracy of the given classes increase and decrease are given in the table 4.

so from the above results, both methods are good for forecasting the spreads with slightly higher directional accuracy than the fifty percent. But in both of the time series methods ARIMA beats DMD in terms of predictive performance.

Table 4. Comparison of directional accuracies for each class with actual class

Type	ARIMA	DMD
Directional accuracies	0.53	1.53

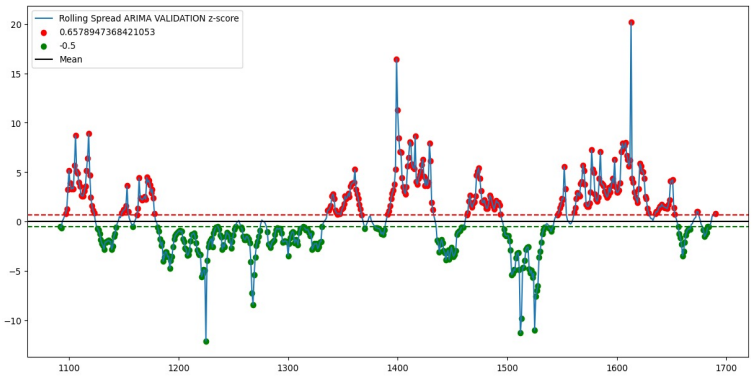

Fig. 7. Trade signals generated on validation data and found hyperparameters using ARIMA.

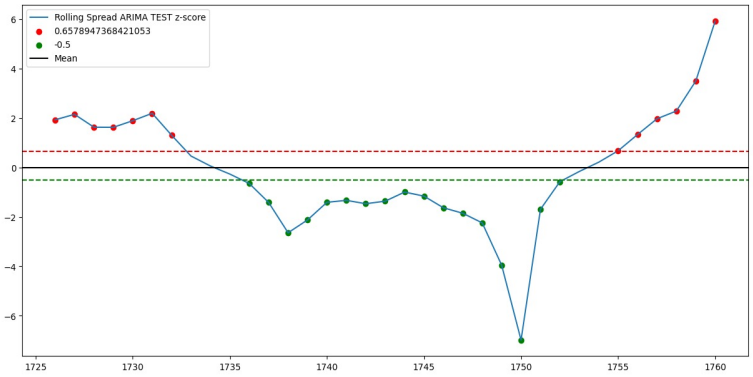

Fig. 8. Trade signals generated on test data and found hyperparameters using ARIMA

C. Trading Strategy. The evaluation of trading strategy requires some processing steps to implement it in best possible way. It requires the normalization of spread data and hyper parameter optimization. They are described in this section with actual algorithm steps.

C.1. z-score. As seen from the spread graph shown., spreads are not mean reverting. So to get the mean reverting behaviour the normalization is performed over spread using two moving averages one at higher window and one lower window. The standardization is done via standard deviation of higher window. so z-score is calculated via below formula.

$$z = \frac{\mu_1 - \mu_2}{\sigma}$$

μ_1 = moving average of spread having window1 size

μ_2 = moving average of spread having window2 size

σ = standard deviation of spread having window2 size

C.2. Threshold Setting. To generate the buy and sell signals, threshold is required to optimize trading strategy. Moreover, it is important to set threshold to square off the position we hold. For this case, following are the set of hyper parameters are used for threshold setting of trading strategy and calculation of z-score which are described in 5

So considering all these parameters, approximately we have to compute for 1.2 million parameter search to get optimal results. Such a large scale parameter optimization is time consuming. So to reduce the time of obtaining optimal parameters we used Bayesian Optimization technique for this large scale optimization. To implement Bayesian Optimization we have used mango-arm package which has in-built parallelization and easy to customize according to our requirements. For evaluation of optimal parameters using Bayesian technique we are limiting our evaluation to 150 iterations. The complete steps of algorithm is shown below.

Table 5. Sets of hyperparameters for trading strategy

Parameter	start	end	gap	no. of parameters
window1	10	100	2	45
window2	1	7	1	7
Sell Threshold	0.5	1	0.025	20
Buy Threshold	-0.5	-1	0.025	20
Square off Threshold	0	0.5	0.05	10

D. Performance of Trading Strategy. To evaluate the trading strategy first we divide the raw data into three categories say training, validation and test data. The use of training data to one step ahead forecast using two different time series algorithms that we have discussed so far. Then from the validation data set we actually tune hyper parameters and on the test data we actually implemented the strategy to see overall money gain.

The figure signifies the trading signal generation using ARIMA method on validation data (hyper parameter tuning) and test data for which results of monetary gain is reported. Similarly same analysis is done using for DMD and actual foretasted spread.

The table 3 summarizes the performance of money gain with different time-series algorithms and it is compared with the actual one day step ahead spread values

Algorithm 1 Trading Strategy

if window1 == 0 or window2 == 0 **then**
  return 0
Calculate z-score of spread using window1 and window2
Initialize money, countS1, and countS2 with zero
for each data point in the spread **do**
  if z-score > sell_threshold **then**
    Sell short S1 and buy S2 * spread.
    Update money, countS1, and countS2.
  else if z-score < buy_threshold **then**
    Buy long S1 and sell S2 * spread.
    Update money, countS1, and countS2.
  else if abs(z-score) < clear_threshold **then**
    Clear positions by selling S1 and buying S2.
    Update money, countS1, and countS2 to zero
return Total money after trading + including the value of remaining open positions.

Table 6. Comparison of Money gain with different time series algorithms

Time series method	Money on validaion data	Money on test data
ARIMA	7332.769	1200.463
DMD	9252.262	47.089
Actual	8447.9713	8447.971

4. Conclusion

The following article summarizes the implementation of pairs trading strategy using proposed time series alogrithms. Also, for the trading strategy optimal hyperparameter tuning is done via Bayesian Optimization. And results shows that ARIMA works better than the DMD in terms of money gain and it is evident from it's higher accuracy of forecasting values.

Supporting Information (SI). The code available for this project can be found on following repository code.

ACKNOWLEDGMENTS. We all want to thank Professor Dr. Xiaodong Lin for teaching us Time series course in great depth.

1. Vidyamurthy G (2004) *Pairs Trading: quantitative methods and analysis.* (John Wiley & Sons) Vol. 217.
2. Maller RA, Müller G, Szimayer A (2009) Ornstein–uhlenbeck processes and extensions. *Handbook of financial time series* pp. 421–437.
3. Tsay RS (2005) *Analysis of financial time series.* (John wiley & sons).
4. Schmid PJ (2022) Dynamic mode decomposition and its variants. *Annual Review of Fluid Mechanics* 54:225–254.
5. Le Clainche S, Vega JM (2017) Higher order dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems* 16(2):882–925.