Statistical Methods In Pairs Trading: An Introduction

Name: Hongyue Li
SUNet ID: lhy
Department of Computer Science
Stanford University
lhy@stanford.edu

1 Introduction

In the field of financial time series analysis, the ability to accurately predict future security returns stands at the core of the industry's success. The formulation of trading strategies hinges on a nuanced understanding of the financial market's forthcoming dynamics. This domain encompasses two primary methodologies: fundamental analysis and quantitative trading. While the former relies on subjective assessments of an industry or company's future trajectory, drawing from public information such as market news and financial statements, the latter leverages mathematical models to minimize the influence of human subjectivity and emotion.

Traditionally, quantitative strategies have leaned on established tools like linear regressions, ARIMA, and GARCH models to distill time series features and capture volatility's stochastic nature. While effective in previous market regimes, shifts in the financial industry landscape have rendered these models less potent. As the industry evolves, the quantitative trading landscape has transitioned into the 'deep learning era', harnessing the formidable capabilities of neural networks in modern financial analysis.

In this project, we aim to predict stock returns utilizing advanced deep learning models, specifically the LSTM architecture. Unlike conventional financial time series prediction methods that rely solely on price and volume data, our approach incorporates additional critical factors, including market momentum and moving average statistics, which wield significant influence over future price movements. The model's output is a prediction of the spread of the chosen pair of stocks for the following day.

Subsequently, armed with these predictions, we will construct a simple pairs trading strategy. This strategy's performance will be rigorously compared against the broader market, offering valuable insights into the potential efficacy of our combined approach. By melding established statistical methods with cutting-edge deep learning techniques, this project seeks to redefine the boundaries of pairs trading, presenting a comprehensive framework for investors to navigate the complexities of today's dynamic financial markets.

2 Related Work

In the realm of financial time series analysis and predictive modeling, numerous studies have explored diverse methodologies and techniques. This section reviews relevant literature and research conducted in the broader domains of quantitative finance, deep learning applications in finance, and pairs trading strategies.

2.1 Quantitative Trading Strategies

Traditional quantitative trading strategies have extensively employed statistical models such as linear regression, Autoregressive Integrated Moving Average (ARIMA), and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models. These models have been valuable tools for capturing trends, seasonality, and volatility patterns in financial time series data. Research by [1]

demonstrated the effectiveness of linear regression models in predicting stock returns based on historical data.

However, recent shifts in market dynamics and the growing complexity of financial instruments have prompted researchers to explore more sophisticated approaches, leading to the emergence of machine learning techniques in quantitative finance.

2.2 Deep Learning in Financial Time Series Analysis

The integration of deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, has gained prominence in predicting financial time series. LSTMs, a type of recurrent neural network (RNN), are well-suited for capturing sequential dependencies in time series data. Studies by [2] and [3] demonstrated the superiority of LSTM models in predicting stock prices and volatility compared to traditional methods.

2.3 Pairs Trading Strategies

Pairs trading is a well-established strategy in quantitative finance that involves exploiting relative price movements between two correlated assets. Research by [4] demonstrated the effectiveness of pairs trading in generating consistent profits. However, traditional pairs trading models often rely on statistical arbitrage and may lack adaptability in dynamic market conditions.

2.4 Integration of Statistical Methods and Deep Learning

The convergence of established statistical methods with deep learning techniques in financial modeling has been explored by researchers to enhance prediction accuracy and robustness. Studies by [5] and [6] showcased the synergies of combining traditional statistical models with deep learning architectures for improved forecasting performance.

3 Data

We use historical exchange-traded funds(ETFs) data downloaded from YahooFinance(yfinance) to test our trading strategy. We identify cointegrated pairs of ETFs based on their close price as potential candidates for pairs trading.

Definition 3.1 (Order of Integration)**.** *A time series X_t is integrated of order d, denoted I(d), if $(1-B)^d X_t$ is a (weakly) stationary process, where B is the backshift operator.*

Definition 3.2 (Cointegration)**.** *For a collection of order d time series $\{X_{it} | \forall i \in I, X_{it} \in I(d)\}$, if $\exists \beta \in \mathbb{R}^{|I|}$ such that $\beta^T X_{it}$ is integrated of order less than d, then X_{it} are cointegrated.*

To calculate the cointegration score of two time series, we use the Engle-Granger Two-Step Method described below.

Step 1: Regression

1. **Run Regression:** Regress one time series on the other using a simple linear regression model:

$$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t$$

   where Y_t and X_t are the two time series, β_0 and β_1 are coefficients to be estimated, and ε_t is the error term.

2. **Test for Stationarity of Residuals:** Check the residuals (ε_t) for stationarity using a unit root test, such as the Augmented Dickey-Fuller (ADF) test. The null hypothesis is that the series has a unit root (non-stationary). If the null hypothesis is rejected, it suggests the presence of cointegration.

Step 2: Cointegration Test

1. **Perform Cointegration Test:** Once you've established the cointegration relationship in step 1, perform a cointegration test, such as the Johansen test or the Engle-Granger test, to confirm the cointegration relationship.

    - **Johansen Test:** This is a likelihood ratio test that tests for the number of cointegrating vectors. It is often used for more than two time series.
    - **Engle-Granger Test:** This is a simpler test that involves a t-statistic on the estimated coefficient of the lagged residual from the regression in step 1. If the t-statistic is significant, it supports the presence of cointegration.

If the cointegration test is significant, it suggests that the two time series are cointegrated. This implies that there is a long-term relationship between the two series that can be exploited in pairs trading strategies. However, cointegration does not imply causation, and it's crucial to carefully interpret the results.

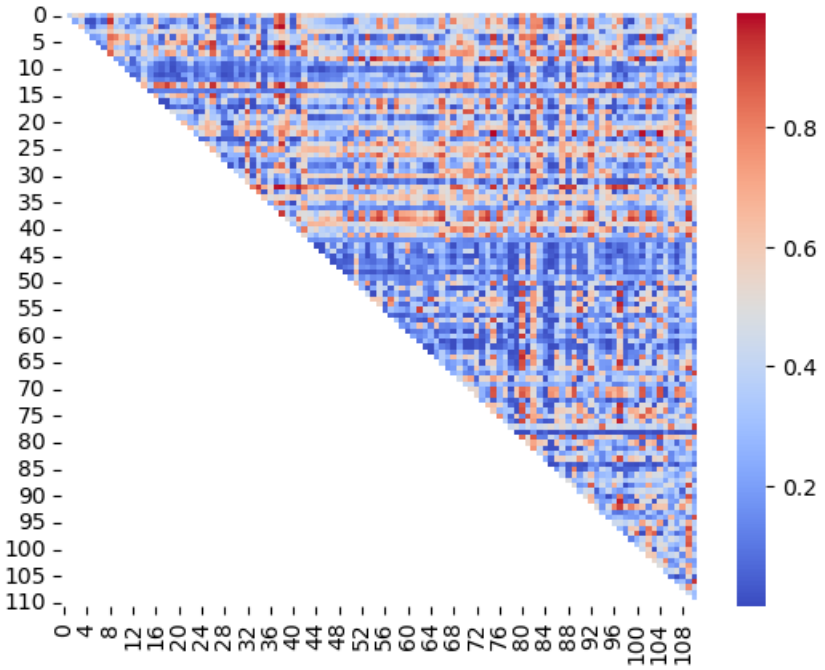

Figure 1: p-values of cointegrated pairs

For each pair of stocks, we calculate the p-value of the cointegration, and we select the pairs with small p-values. Here we included the top 10 pairs with the smallest p-values in the table below.

Pair 1	Pair 2	Value
PDP	XLF	1.027e-06
PDP	VFH	1.125e-06
PDP	PRF	1.147e-06
IWO	KRE	4.376e-06
IVV	SPY	7.271e-06
IWO	KBE	7.561e-06
PDP	RSP	2.913e-05
IWP	IYF	3.410e-05
IWP	XLF	3.879e-05
IWP	VFH	5.560e-05

Table 1: Cointegration Scores

Based on the p-values of the cointegration, we selected the pair IWP(iShares Russell Mid-Cap Growth ETF) and VFH(Vanguard Financials Index Fund ETF) as our candidates. The reason that we discard other pairs with smaller p-values is that those pairs actually contain almost the same underlying portfolio and we do not expect that their prices will diverge even with such a high cointegration score.

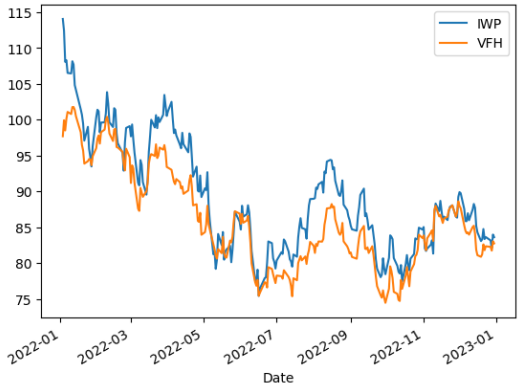

Figure 2: close prices of IWP and VFH

Call the two time-series IWP and VFH x_t, y_t respectively. We perform a linear regression to determine β such that $y_t = \beta x_t + \epsilon_t$, where ϵ_t is a stationary process. We check if the residual is stationary using the Augmented Dick-Fuller (ADF) test.

We extracted the close, open, high, low, and volume traded as features and augmented the data with the following extracted features using the Technical Analysis library(ta): relative strength index, money flow index, accumulation/distribution indicator, volume price trend, average true range, Bollinger Bands, average directional index, exponential moving average, moving average convergence/divergence, daily log return. We also added the spread $y_t - \beta x_t$ of the two stocks of their close, open, high, and low prices.

Technical Indicator	Type of Indicator	Description
Relative Strength Index	Momentum (t=14)	$RSI = 100 - \frac{100}{1+RS}$ $whr\ RS = \frac{Average\ of\ t-day's\ up\ closes}{Average\ of\ t-day's\ down\ closes}$
Money Flow Index	Momentum (t=14)	$MFI = 100 - \frac{100}{MFR}$ $whr\ MFR = \frac{t-period\ Positive\ Money\ Flow}{t-period\ Negative\ Money\ Flow}$
Accumulation / Distribution Index	Volume	$ADL = ADL_{prev} + (MFM * Volume)$ $whr\ MFM = \frac{[(Close-Low)-(High-Close)]}{High-Low}$
Volume – Price Trend	Volume	$VPT = VPT_{prev} + (Volume * \frac{Close_{today} - Close_{prev}}{Close_{prev}})$
Average True Range	Volatility (n=14)	$ATR_t = \frac{ATR_{t-1}*(n-1)+TR_t}{n}$ $where\ TR_t = \max(High - Low, abs(High - Close_{prev}), abs(Low - Close_{prev}))$
Bollinger Bands	Volatility (n=20)	$BB = n-day\ Simple\ Moving\ Average$
Average Directional Movement Index	Trend (n=14)	$ADMI = Average\ of + ve\ Directional\ Movement\ and - ve\ Directional\ Movement$
Exponential Moving Average	Trend (n=14)	$EMA = [Close - EMA_{prev}] * \frac{2}{n+1} + EMA_{prev}$
Moving Average Convergence Divergence	Trend (n_fast = 14, n_slow = 30)	$MACD = EMA_{n\_fast} - EMA_{n\_slow}$
Daily Log Return	Other	$Log\ Return = ln\frac{Close_i}{Close_{i-1}}$

Figure 3: Momentum and Moving Average Features

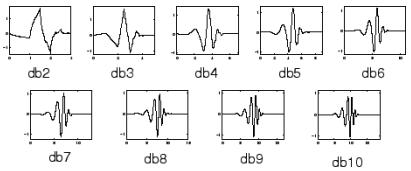

Figure 4: Daubechies Wavelets Basis

Then we use discrete wavelet transform via 'pywt' package in python to denoise each feature column. We chose Daubechies Wavelets 8('db8') for its ability to capture both smooth and oscillatory patterns in data. We discard coefficients that are below 1/8 standard deviation of the coefficient array.

4 Methods

In order to perform pairs trading, we need to predict the spread of the two stocks. Here we consider the simplest case that given all data up to time 't', we aim to predict the spread of the pair at the time 't+1'. In this section, we compare the performance of five methods to predict the spread.

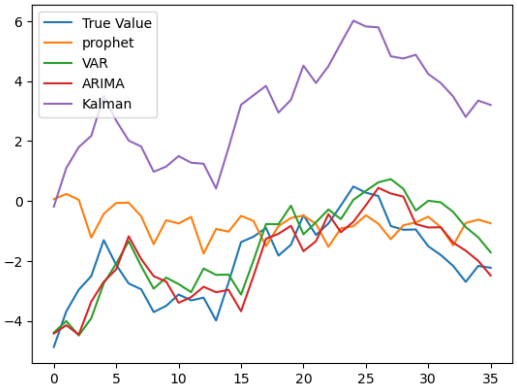

Figure 5: Predicted Spread of Various Models

4.1 ARIMA

ARIMA(Auto-Regressive Integrated Moving Average) models are frequently employed in stock price prediction due to their ability to effectively capture time series components, including trends and seasonality while handling non-stationarity through differencing. The autoregressive (AR) component captures the influence of past stock prices, reflecting serial correlation in financial markets, while the moving average (MA) component accounts for short-term fluctuations. The simplicity and interpretability of ARIMA models, along with their availability in popular statistical software, make them a common baseline for comparing more complex predictive models.

We use the spread as the only feature in the ARIMA model. The model is sensitive to hyperparameters so we have to determine these first. We first plot the Auto-correlation and Partial Auto-Correlation functions of the time series, together with AIC and BIC scores to determine the hyperparameters of the ARIMA model. We use the simple ARIMA(1,0,1) model to forecast the step 1 ahead spread given the history of the spread.

The Mean Squared Error (MSE) on the train set is 0.13, and on the test set is 0.83.

4.2 Kalman Filter

The Kalman Filter is a powerful tool in time series analysis, particularly for estimation and prediction in dynamic systems. We hope that the Kalman filter can estimate the underlying dynamics of the evolution of the two financial time series.

We only use the close price of the two stocks as features. We treated the two stocks x_t, y_t as a linear regression problem and computed the online update β_t such that $y_t = x_t\beta_t$ via Kalman filtering. The transition and observation matrices are learned using the Expectation-Maximization(EM) algorithm.

The Mean Squared Error (MSE) on the train set is 1.13, and on the test set is 4.23.

4.3 Vector Autoregressive Model(VAR)

Vector Autoregression (VAR) models find application in stock price prediction due to their ability to capture the interdependencies among multiple stocks simultaneously. By modeling the endogenous relationships and contemporaneous interactions among different variables, VAR models offer a framework to analyze and predict stock price movements in a comprehensive manner. Their incorporation of lagged values and flexibility in handling simultaneity makes them particularly suitable for capturing the dynamic and interconnected nature of financial markets.

We used all the 26 features and modeled it as a VAR(26) process. The MSE error on the train set is 0.24, and on the test set is 1.16.

4.4 Prophet

Prophet, a forecasting tool developed by Facebook, is employed for financial time series analysis due to its ability to handle inherent challenges such as irregularities, missing data, and seasonality. It incorporates a flexible additive model that accounts for daily, weekly, and yearly patterns, making it well-suited for capturing the complex seasonality often observed in financial data. Prophet is robust to outliers and offers an intuitive platform for incorporating domain knowledge through customizable holidays and special events. Its ease of use, automatic handling of data gaps, and adaptability to various data characteristics make it an attractive choice for analysts in the financial domain seeking accurate and interpretable predictions, particularly for time series with distinct patterns and irregularities.

We treated spread as the target variable and used the Prophet package to forecast the spread. The MSE error on the train set is 1.24, and on the test set is 3.56.

No	Method	No. Features	MSE Test Error
1	Kalman Filter	1	4.23
2	ARIMA	1	0.83
3	VAR	All	1.16
4	Prophet	1	3.56

Table 2: Experimental Results

4.5 Long-Short Term Memory(LSTM)

As we can see in the previous section, traditional statistical models often have restricted modeling ability and cannot capture much non-linearity.

Deep learning methods are considered advantageous for predicting stock returns due to their ability to capture intricate patterns, non-linear relationships, and high-dimensional features within financial time series data. Due to the expansive capacity of deep learning models, particularly neural networks, it can learn complex representations from data. Unlike traditional models that may struggle to capture nuanced dependencies, deep learning models, such as Long Short-Term Memory (LSTM) networks, can adapt to the evolving dynamics of financial markets and uncover hidden patterns.

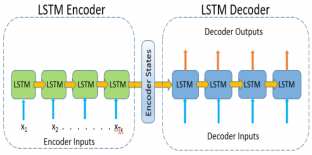

Figure 6: LSTM Model Structure

We implemented a two-layer LSTM model with hidden dim = 10 after embedding layers of dim = 64. Dropout probability of 0.3. The inputs are just the 26 features. We train it with Adam optimizer with a learning rate of 1e-3 for 1000 epochs, the MSE on the training set is 0.0006, and the MSE on the test set is 0.0156.

Unlike traditional statistical methods which can be trained in seconds, due to its sequential structure, the LSTM model took much longer time(about 1.5 hours) to train. For our dataset, this is still manageable. But for longer time series we can mitigate this issue by gradient clipping, learning rate scheduling, and other techniques.

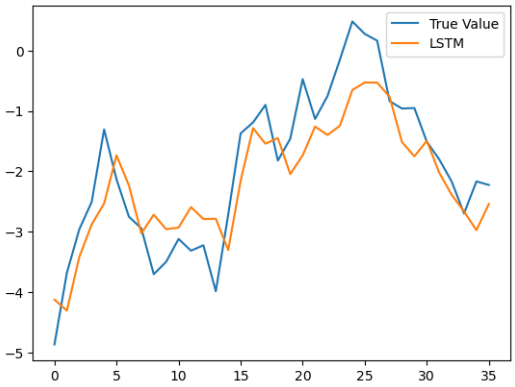

Figure 7: LSTM predicted spread

4.6 Pairs Trading Strategy

After we forecasted the spread using one of the methods described above, we calculated the Z-score of the spread, where we used a rolling window mean and standard deviation to approximate the true values. Now if the Z-score is greater than 1, we long the first stock and short the second stock. If the Z-score is less than 1, we short the first stock and long the second. If the Z-score is between -0.5 and 0.5, we clear our holdings.

Now the expected profits of the 5 methods are included below:

Method	Expected Profit
Kalman	238.31
ARIMA	1354.46
VAR	1024.13
Prophet	-183.80
LSTM	1427.08

5 Conclusion

In this paper, we studied the pairs trading strategy by identifying cointegrated pairs and forecasted the spread with statistical learning models. We performed pair selection using the cointegration approach. We added additional momentum and moving average features and smoothed the data using Wavelet transform.

Traditional statistical methods like ARIMA, Kalman Filter, VAR, and Prophet are fast to train, and less sensitive to hyperparameters but have less modeling ability. LSTM models are more sophisticated and are able to capture temporal dependencies and patterns that may not be adequately represented by traditional standalone numerical values but take a longer time to train. It was observed that the LSTM model achieved the lowest train and test MSE loss. With the LSTM model, we were able to predict the 1-step ahead spread with high accuracy. The prediction of 1-step ahead spread is performing well and achieved nearly optimal spread value.

Lastly, there are a few future directions to extend our work. We can explore the attention mechanism and incorporate it into our LSTM model. The model can be further improved by providing outside information other than stock prices into consideration. For example, incorporating the sentiment score of the current news and Twitter posts via NLP techniques may provide valuable information about the selected stocks.

6 Appendices

The code is available here: `https://github.com/ordinarylhy/CS229BProject`.

References

[1] Carlos Eduardo De Moura, Adrian Pizzinga, and Jorge Zubelli. A pairs trading strategy based on linear state space models and the kalman filter. Quantitative Finance, 16(10):1559–1573, 2016.

[2] Andrea Flori and Daniele Regoli. Revealing pairs-trading opportunities with long short-term memory networks. European Journal of Operational Research, 295(2):772–791, 2021.

[3] Victor Chang, Xiaowen Man, Qianwen Xu, and Ching-Hsien Hsu. Pairs trading on different portfolios based on machine learning. Expert Systems, 38(3):e12649, 2021.

[4] Evan Gatev, William N Goetzmann, and K Geert Rouwenhorst. Pairs trading: Performance of a relative-value arbitrage rule. The Review of Financial Studies, 19(3):797–827, 2006.

[5] Sang-Ho Kim, Deog-Yeong Park, and Ki-Hoon Lee. Hybrid deep reinforcement learning for pairs trading. Applied Sciences, 12(3):944, 2022.

[6] Lan Wu, Xin Zang, and Hongxin Zhao. Analytic value function for a pairs trading strategy with a lévy-driven ornstein–uhlenbeck process. Quantitative Finance, 20(8):1285–1306, 2020.