Pairs Trading: Modeling Price Spread and Identifying Pairs

Beri Kohen Behar
Mathematical and Computational Science
Stanford University
bkohen@stanford.edu

Lucas Pauker
Physics
Stanford University
lpauker@stanford.edu

1 Introduction

Pairs trading is a difficult yet profitable market-neutral trading strategy. Pairs trading seeks to profit from the relative price movements of two stocks [1]. A trader looks to find volatile market conditions where two correlated stocks significantly diverge in price. The trader then takes a short position in one stock and a long position in the other. There are three main components to pairs trading: picking the pair of assets to trade, modeling the relation between the price of these assets, and executing a trading strategy. In this paper, we explore the first two components. In a real world setting, randomly selecting pairs of stocks is unlikely to produce good results, while exhaustively trying all pairs of stocks is practically impossible because of the high number of possible combinations and the computational power that would be required to train that many models (especially when using deep learning models). Thus, finding simple metrics that can be computed quickly and can indicate good model performance is valuable for of pairs trading. To model the relation between the price of two stocks we use Ornstein-Uhlenbeck modeling and long short-term memory (LSTM) networks. We then attempt to find time series statistics that are good indicators of model performance.

2 Methods

2.1 Stock Selection

We select stocks from three sectors: energy, healthcare, and financial services. These sectors are selected since they are large and generally move separately from each other. Within each sector, we choose five of the largest market cap stocks. All stocks chosen are traded on the New York Stock Exchange. Stocks with a large market cap tend to have the most volume and liquidity and are therefore easiest to trade and best for pairs trading. Figure 1 shows all the stocks we used in our analysis. We used daily close prices for these stocks from December 2016 to December 2021. This data comes from Yahoo Finance and is free to use.

2.2 Baseline Model

For our baseline model, we chose to use the one timestep lagged predictions. Therefore, for a spread at time t X_t, the baseline model predicts X_{t-1}. We chose this baseline for two reasons. First, this baseline only uses the past prediction, so comparing to this baseline shows how effectively our models can synthesize multiple past data points in a prediction. Second, this baseline is easy to implement.

2.3 Ornstein-Uhlenbeck Model

We construct the spread X_t between two stock prices A_t and B_t as

$$X_t = A_t - \beta B_t, \tag{1}$$

Sector	Ticker	Company Name
Energy	XOM	Exxon Mobil Corp
Energy	CVX	Chevron Corporation
Energy	RYDAF	Royal Dutch Shell Plc
Energy	PTR	PetroChina Company Limited
Energy	TTE	TotalEnergies SE
Healthcare	UNH	UnitedHealth Group Inc
Healthcare	CVS	CVS Health Corp
Healthcare	ANTM	Anthem Inc
Healthcare	HCA	HCA Healthcare Inc
Healthcare	MCK	McKesson Corporation
Financial Services	JPM	JPMorgan Chase & Co.
Financial Services	V	Visa Inc
Financial Services	BAC	Bank of America Corp
Financial Services	MA	Mastercard Inc
Financial Services	PYPL	Paypal Holdings Inc

Figure 1: Table of the stocks selected for our analysis.

where β is a scalar parameter. The spread of the two stock prices is assumed to be a mean-reverting time series process. An Ornstein-Uhlenbeck (OU) model is constructed using the following Stochastic Differential Equation:

$$dX_t = \mu(\theta - X_t)dt + \sigma dW_t, \tag{2}$$

where θ is the mean that the process converges to in the long term, μ is the speed of reversion, σ is the instantaneous volatility, and W_t is a Weiner process (one dimensional Brownian motion). In order to fit θ, μ, and σ, we use a least squares regression approach. First, we can write an exact solution of the differential equation [2]:

$$X_{t+1} = X_t e^{-\lambda\delta} + \mu\left(1 - e^{-\lambda\delta}\right) + \sigma\sqrt{\frac{1 - e^{-2\lambda\delta}}{2\lambda}}\mathcal{N}(0,1). \tag{3}$$

δ is the time step between subsequent observations; we use $\delta = 1$ since we are modeling the process with daily stock prices. We can see that this is an AR(1) process with drift. Therefore, we can fit an AR(1) process to the data to extract the parameters θ, μ, and σ.

One interesting property of the OU model is that the further away the spread is from its long term mean, the faster it reverts to it. One practical problem with this model is that it assumes that the stock prices are co integrated. This means that there exists a scalar β such that the spread X_t is stationary. We ran the ADF test on different pairs of stocks, with various values of β, and found that in most cases two pairs of stocks are not co-integrated over a long period, even if the companies have similar businesses. However, it is reasonable to assume that a pair is co-integrated for a shorter time period. Therefore, we decide to use a rolling window to calculate the spread. To calculate the spread at time t, we use the data from d previous days. We use linear regression to calculate the the optimal β_t that minimizes the following expression:

$$\sum_{i=t-d}^{t-1} (A_i - \beta_t B_i)^2. \tag{4}$$

Using the β_ts, we calculate the new spread, and fit the OU equation to this new spread. For our experiments we used the value $d = 10$.

2.4 Deep Learning with LSTMs

As an alternative to OU Modeling, we model the price difference of two stocks using LSTM-based neural networks. This has been explored in previous research with moderate success [3]. When using LSTMs, less data processing is needed. Specifically, we do not have a coefficient β that we try to calculate because there are no assumptions relating to stationary. Therefore, we can use the raw spread X'_t for stock prices A_t and B_t

$$X'_t = A_t - B_t \tag{5}$$

as input to the model. One advantage of LSTMs (and deep learning in general) is that it can uncover complex non-linear relationships that traditional statistical models cannot. One downside is that is that the parameters and the predictions of the model are often not easy to explain. Also, there are many hyper-parameters related to the architecture and the training process that need tuning, which requires a certain level of expertise in the field.

LSTMs are a version of standard neural networks that work better with sequential data. They are more sophisticated than RNNs (Recurrent Neural Networks), because they have to ability to carry information from much earlier data in a sequence. It's not clear whether long sequences are needed to predict the price difference of two stocks, but because of our limited time we decided to go along with using LSTMs instead of RNNs. For each data point, we build sequences of length 51, where the first 50 elements are the price differences from the previous 50 days. Even if the 50 day period we chose is unnecessarily long, we would expect LSTM "forget gate" weights to learn this during training, so we don't need to spend time optimizing the sequence length. Figure 2 shows a diagram of an LSTM cell.

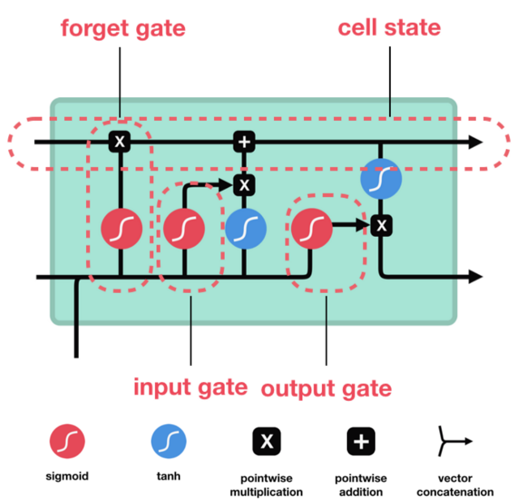

Figure 2: Figure visualizing how an LSTM cell works.

Our model we use includes two layers of LSTMs. This means that the sequence is fed through an LSTM, and the output of the first LSTM is fed through another LSTM. The output of the second LSTM passes through a dense layer that transforms a vector in to a scalar, which is our final prediction of the price difference. We used two layers because it performed better than a single layer network and using three layers significantly increased training time while not improving model performance. To avoid overfitting, we used dropout for each LSTM layer, monitored the validation set loss and used early stopping.

Ideally, we would want to tune the hyper-parameters for each pair spread model separately for optimal model performance. This is not practical in our case, however, since even a small set of 15 stocks results in 210 pairs. Therefore, we tuned the hyper-parameters for the spread of a single pair that was performing well, and then trained separate models for all pairs using the same architecture and hyper-parameters. This challenge that we faced further supports the significance of developing pair selection methods. For example, if we had a method of predicting the top 5 pairs out of a 210 using a simple statistic, we could tune their models separately.

2.5 Methods for Picking Pairs

We calculate four metrics for each pair of stocks: cumulative distance, Augmented Dickey-Fuller (ADF) test p-value, standard deviation of the spread with no beta adjustment, and standard deviation of the spread with beta adjustment.

The first metric we use to compare each pair of stocks is the distance between the normalized cumulative returns of two stocks. This method has been used in previous literature on pairs trading [1]. Let $x(t)$ and $y(t)$ be the prices of two stocks at time t. This distance is given by the following formula:

$$\sum_{i=0}^{N}(C_{x(t)} - C_{y(t)})^2 \tag{6}$$

where $C_{x(t)} = \dfrac{x(t) - x(0)}{x(0)}$, and $C_{y(t)} = \dfrac{y(t) - y(0)}{y(0)}$.

The second metric we look at is the p-value of the ADF test, which is used to test if a time series is stationary. When applying this metric to a pair of stocks, we first normalize the prices for each stock by subtracting the mean and dividing by the standard deviation. Then we take the difference of the two normalized time-series, and apply the ADF test.

The third metric we use is the standard deviation of the spread between the two stocks. The spread for the price series for two stock prices A_t and B_t is simply $A_t - B_t$ as described in Section 2.4. Before taking the difference, we normalize the prices for each stock by subtracting the mean and dividing by the standard deviation.

The fourth metric we use is the standard deviation of the β-adjusted spread between the two stocks. The β-adjusted spread for the price series for two stocks is described by Equation 4.

3 Experiments

3.1 Metrics Results

We calculate the metrics from Section 2.5 for all pairs of stocks. Figure 3 includes a heatmap for each metric calculated for all pairs of stocks. The stocks are ordered by sector; stocks in the same sector are adjacent. For each metric, we expect the 5x5 blocks on the diagonals to be darker (lower distance) since these correspond to stocks in the same sector. For the cumulative distance heatmap, the top left 5x5 block is dark, meaning the distance between the stock prices of energy companies is quite low. This is likely because they all heavily depend on oil prices and generally move together. In contrast, the financial services sector tends to have higher distances than the energy sector. This is likely because this sector contains many different types of businesses such as Bank of America (BAC) and PayPal (PYPL). Furthermore, we see some similarities between the ADF p-value heatmap and the cumulative distance heatmap. Notably, the pairs in the energy sector in the top left 5x5 block have low p-values. From the heatmap we also see that standard deviation of the spread is low for energy stocks and high for financial stocks, similar to the heatmaps for cumulative distance and p-value. Lastly, from the figure we see that the standard deviation of the β-adjusted spread is also low for energy stocks and higher for financial stocks.

3.2 OU Modeling Results

We fit an OU model using the methodology described in Section 2.3 to each pair of stocks. Figure 4 shows an OU model fit to spread between two tickers. We can see that the OU predictions look similar to one step lagged predictions. However, the model notably tends to the mean more than one step ahead predictions would. This is a feature of the OU model.

For each pair of stocks, we calculate the mean squared error (MSE) for the OU process. We then compare the OU MSE to the MSE of the baseline model described in Section 2.2. We compute the ratio of OU vs. baseline MSE by dividing the OU MSE by the baseline MSE. A ratio less than one

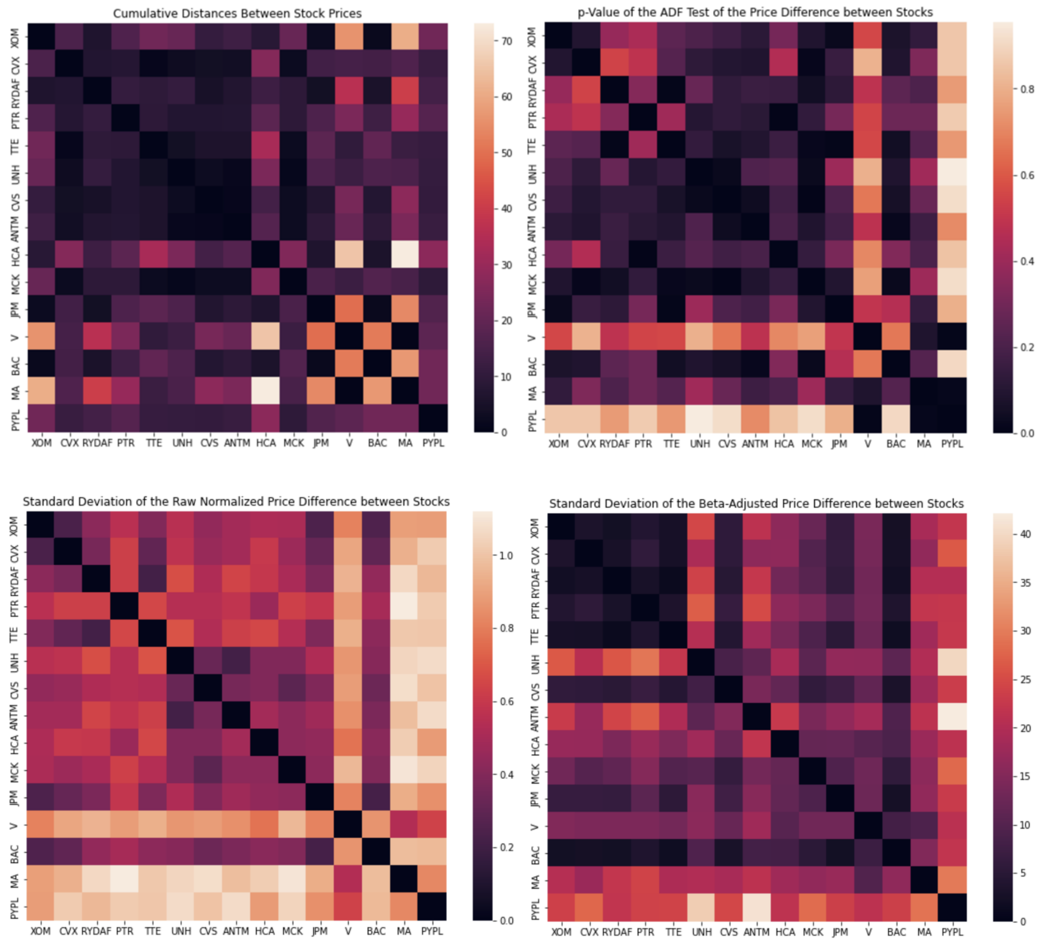

Figure 3: *Top Left*: Heatmap of the distances between stocks using the difference between normalized cumulative returns of stocks. *Top Right*: Heatmap of the p-value of the ADF test applied on the difference of normalized stock prices. *Bottom Left*: Heatmap of the standard deviation of the normalized price difference between stocks. *Bottom Right*: Heatmap of the standard deviation of the β-adjusted price difference between stocks.

indicates that the OU model has lower MSE than the baseline model and a ratio greater than one indicates that the OU model has a higher MSE. This ratio is an indicator of how good the OU model is compared to the baseline. We create a heatmap of this ratio for each pair of stocks in Figure 5. We see that the best OU baseline MSE ratio is for pairs of stocks in the energy sector. This means that the spread for pairs of stocks in the energy sector are best modeled by an OU process compared to stocks in other sectors.

We also compare the metrics described in Section 2.5 to the OU baseline MSE ratio. Intuitively, we would like to see if the metrics described are good predictors of OU MSE fit. Figure 6 shows the raw (not β-adjusted) standard deviation metric compared to the OU baseline MSE ratio. We see that there is a positive correlation between these two variables: lower standard deviation generally corresponds to lower MSE and vice versa. This makes sense because spreads with lower variance should result in better OU fits. Although there is not a perfect relationship between standard deviation and OU baseline MSE ratio, we can use the standard deviation to eliminate pairs that perform poorly with OU modeling. We can see that the best fits (i.e. lowest OU baseline MSE ratio) occur when the standard deviation is less than 0.5. Therefore, when we are choosing which pairs to use for modeling the spread, we can eliminate pairs with a standard deviation above 0.5 since they are likely to have poor OU fits. This is useful because around half of the pairs have a standard deviation above 0.5, so by

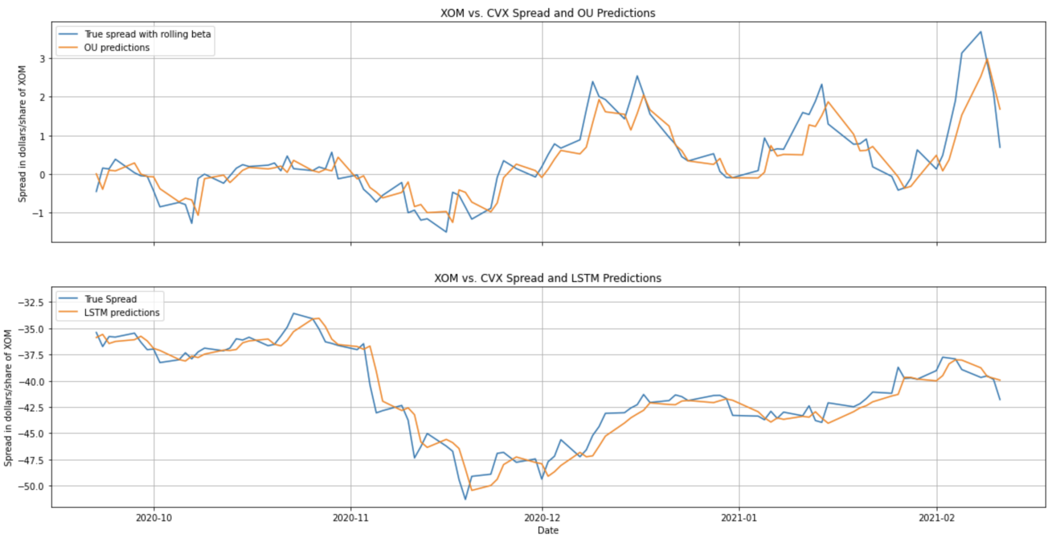

Figure 4: *Upper*: Time series of the spread between XOM and CVX and the OU predictions for several months of data. *Lower*: Time series of the spread between XOM and CVX and the LSTM predictions for several months of data. Note that the true spreads are different for the two graphs since the upper graph spread is adjusted with a rolling β while the lower graph is the raw spread (β=1).

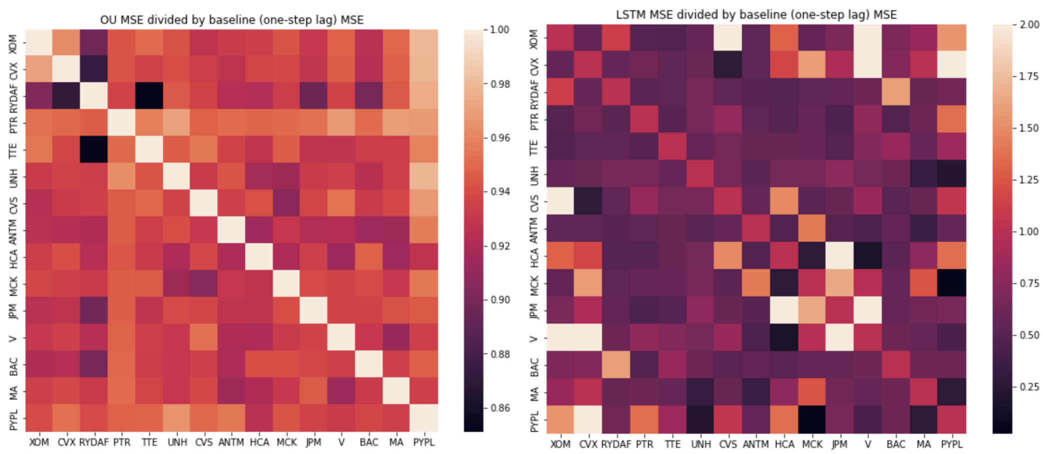

Figure 5: *Left*: Heatmap of OU MSE divided by baseline MSE for all stock pairs. *Right*: Heatmap of LSTM MSE divided by baseline MSE for all stock pairs. The values in this heatmap are clipped to the range [0,2].

eliminating the pairs with high standard deviation spreads, we only have to consider half the total pairs to find the most effective OU models.

3.3 LSTM Results

We fit an LSTM model using the methodology described in Section 2.4 to each pair of stocks. Figure 4 shows an LSTM model fit to spread between two tickers. Similar to the OU model, we can see that the LSTM predictions look similar to one step lagged predictions. However, the model is more smooth than simple one step ahead predictions.

For each pair of stocks, we calculate the MSE for the LSTM process. We then compare the LSTM MSE to the MSE of the baseline model described in Section 2.2. We compute the ratio of LSTM vs. baseline MSE by dividing the LSTM MSE by the baseline MSE. As before, a ratio less than one

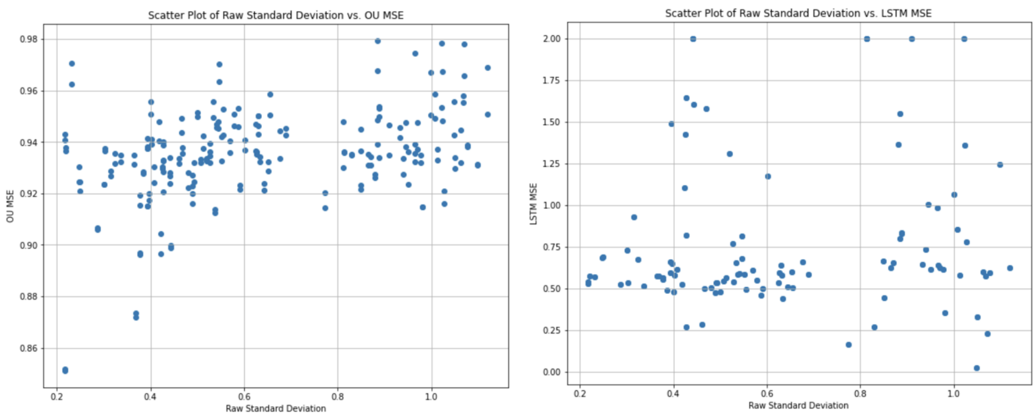

Figure 6: *Left*: Scatter plot comparing OU MSE divided by the baseline MSE of a stock pair to the standard deviation of the spread between the stock prices with no beta adjustment. Each point on the scatter plot represents a stock pair. *Right*: Scatter plot comparing LSTM MSE divided by the baseline MSE of a stock pair to the standard deviation of the spread between the stock prices with no beta adjustment.

indicates that the LSTM model has lower MSE than the baseline model and a ratio greater than one indicates that the LSTM model has a higher MSE. This ratio is an indicator of how good the LSTM model is compared to the baseline. We create a heatmap of this ratio for each pair of stocks in Figure 5. For the heatmap, we restrict the ratio to the range [0,2] for better visualization. In contrast to the OU baseline MSE ratio, the LSTM baseline MSE ratio has a higher range of values. The lowest value for the OU baseline MSE ratio is 0.85, while the highest value is 1. However, the lowest value for the LSTM baseline MSE ratio is 0.10, while the highest value is over 4. This shows that the model is extremely effective for some pairs and less effective for others. From the heatmap, we can see that the best LSTM MSE to baseline MSE ratio is not concentrated in specific sectors. This contrasts to the OU MSE to baseline MSE ratio, where the best ratios occurred for pairs of energy stocks.

We also compare the metrics described in Section 2.5 to the LSTM MSE to baseline MSE ratio. Figure 6 shows the raw (not β-adjusted) standard deviation metric compared to the LSTM baseline MSE ratio. We see that there is little correlation between these variables. Therefore, standard deviation is not a good indicator of LSTM fit. This is likely because LSTMs are nonlinear models and can model the spread of pairs of unrelated stocks well. Overall, we see that the standard deviation metric is more useful for predicting OU model performance compared to LSTM model performance.

4 Challenges and Future Work

4.1 Challenges

One challenge we encountered was comparing model performance between different pairs and different models. Comparing model performance between different pairs was difficult since each stock has a different price and therefore the spreads between different pairs have vastly different scales. To remedy this problem, we normalized the stocks before inputting them into our models. Comparing different models was difficult since the OU model and the LSTM model have different inputs. The OU model uses the β-adjusted spread, while the LSTM model uses the non-adjusted spread. Therefore, we could not compare the MSE or other statistics from the fitted models directly. Instead, we opted to compare each model separately to a baseline model. This let us compare the two models by comparing their improvement over the baseline.

4.2 Future Work

There are many directions that can be taken to extend the work done in this paper. First, more work can be done to tune the rolling window size for the β-adjusted spread. The value of d in Equation 4 determines the size of the rolling window. In our analysis, we used $d = 10$, however changing this value could improve model performance.

Second, the methods developed in this paper can be extended to more pairs of stocks and more sectors. The number of stocks considered in this paper (15) is low compared to the number of available stocks on the New York Stock Exchange (about 2400). Increasing the number of stocks is hard because training the models, specifically the LSTM model, is time-consuming. With more time and compute power, however, our analysis could be easily extended to more stocks and sectors, however.

Third, the experiments in this paper can be applied to more granular data (hour, minute, or even second level). The properties of the differenced time-series can significantly change with more granular data, and taking advantage of shorter divergences in price could be more profitable. Additionally, deep learning methods could prove even more useful because the number of data points for training would be much higher. One interesting research direction would be to apply modern time series deep learning algorithms such as S3 [4] to spread prediction.

Fourth, another way to assess the effectiveness of our models would be to conduct simulated trading using the models. There is existing literature that seeks to find the best trading signal with certain models [1][3][5][6], however optimizing a strategy with our framework and models would be an interesting extension of this work.

References

[1] Evan Gatev, William Goetzmann, and K. Rouwenhorst. Pairs trading: Performance of a relative value arbitrage rule. *Review of Financial Studies*, 19:797–827, 02 2006.

[2] Thijs van den Berg. Calibrating the ornstein-uhlenbeck (vasicek) model, 2011.

[3] 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.

[4] Anonymous. S3: Supervised self-supervised learning under label noise. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. under review.

[5] Tim Leung and Xin Li. Optimal mean reversion trading with transaction costs and stop-loss exit, 2015.

[6] Taewook Kim and Ha Kim. Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. *Complexity*, 2019:1–20, 11 2019.