STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

FA 691 – Deep Learning for Finance

Spring 2023

Spread Prediction in Pairs Trading

Final Project

Sandeep Ranjan

Overview

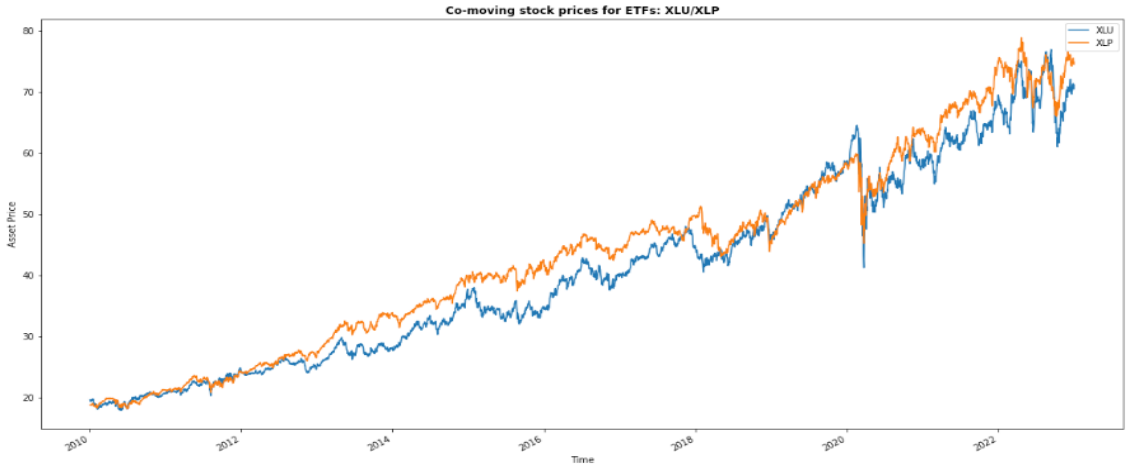
In this project, we use various Deep Learning models to forecast the ***Spread in a Pairs Trading Strategy*** and compare that with more traditional statistical techniques such as *ARIMA* model. Before we talk about any of these models, we first briefly discuss what is Pairs Trading and Statistical Arbitrage

Statistical Arbitrage *refers to trading strategies that employ some statistical model or method to take advantage of what appears to be relative mispricing of assets while maintaining a level of market neutrality.*

Pairs Trading *is a Statistical Arbitrage trading strategy where the goal is to find two (or more) assets whose prices have historically moved together, track the spread (the difference between their prices), and, once the spread widens, buy the asset that has dropped below the common trend and short the other stock. If the relationship persists, the long and/or the short leg will deliver profits as prices converge and the positions are closed.*

The mechanism of Pairs Trading is as follows: First a pair(s) of stocks with similar trends is identified. Then regression analysis such as Ordinary Least Squares (OLS) is used to calculate the spread of these stocks. Finally, if the spread hits preset boundaries, we take a long position on the undervalued stock and short the overvalued stock. Subsequently, as the Spread reverts to its mean we close out the positions thereby making a profit.

Below is a representation of how a pair stock prices can move relative to each other giving traders an opportunity to execute a trading strategy using the Spread between asset prices.



Data Collection

For the purpose of this project, we chose our universe of assets as the 11 Sector ETFs and then tried to find the best possible pair out of these.

1. XLE: Energy Select Sector SPDR Fund
2. XLF: Financial Select Sector SPDR Fund
3. XLU: Utilities Select Sector SPDR Fund
4. XLI: Industrial Select Sector SPDR Fund
5. XLK: Technology Select Sector SPDR Fund
6. XLV: HealthCare Select Sector SPDR Fund
7. XLY: Consumer Discretionary Select Sector SPDR Fund
8. XLP: Consumer Staples Select Sector SPDR Fund
9. XLB: Materials Select Sector SPDR Fund
10. XLRE: Real Estate Select Sector SPDR Fund
11. XLC: Communication Services Select Sector SPDR Fund

The asset prices were retrieved from ***yahoo finance*** for the period: January 1st, 2014 – December 31st, 2022.

	XLE	XLP	XLF	XLU	XLI	XLK	XLV	XLY	XLB	XLC	XLRE
2014-01-02	61.163265	33.471817	14.838815	27.761885	43.406315	31.156721	47.676102	59.570732	38.213570	0.000000	0.000000
2014-01-03	60.938938	33.408737	14.941195	27.672781	43.524235	31.007050	47.788521	59.409542	38.121895	0.000000	0.000000
2014-01-06	61.023075	33.282578	14.954854	27.717340	43.271534	30.954229	47.589619	59.042393	37.913551	0.000000	0.000000
2014-01-07	61.485741	33.463924	14.961674	27.962351	43.541088	31.235950	48.091190	59.400593	37.846863	0.000000	0.000000
2014-01-08	61.058105	33.211613	15.009457	27.813860	43.465271	31.235950	48.514954	59.239407	38.071892	0.000000	0.000000
...	...	...	...	...	...	...	...	...	...	...	...
2022-12-23	87.059998	75.180000	33.950001	70.910004	98.389999	124.599998	136.100006	129.429993	78.519997	47.799999	37.080002
2022-12-27	87.989998	75.540001	33.939999	71.349998	98.690002	123.419998	135.750000	127.330002	78.589996	47.369999	37.049999
2022-12-28	86.019997	74.599998	33.820000	70.669998	97.400002	121.430000	134.869995	126.260002	77.389999	46.680000	36.459999
2022-12-29	86.919998	74.919998	34.290001	71.180000	98.599998	124.610001	136.339996	129.509995	78.230003	47.970001	37.270000
2022-12-30	87.470001	74.550003	34.200001	70.500000	98.209999	124.440002	135.850006	129.160004	77.680000	47.990002	36.930000

2266 rows × 11 columns

Pair Selection

There are different methodologies available for selecting the right pair – one of them being ***Cointegration***. This approach relies on an econometric model of a long-term relationship among two or more assets and allows for statistical tests that promise more reliability than other available methods. We use the Engle-Granger procedure to identify the best pair of assets among the 11 Sector ETFs.

Using the Engle-Granger test, the pair having lowest p-value (of less than 0.05) was selected. Further, Augmented Dickey Fuller (ADF) tests were executed on the spread of this pair to ensure that its stationary and doesn't have unit root. Stationarity of a time series implies that it doesn't have a trend or seasonal effects. After running these tests, the following Pair of ETFs was chosen since it had the lowest p-value. *A lower p-value simply means that there is stronger evidence in favor of the alternate hypothesis (i.e., the series is stationary), thereby rejecting the null hypothesis (non-stationarity of the series)*

XLU: Utilities Select Sector SPDR Fund, **XLP**: Consumer Staples Select Sector SPDR Fund

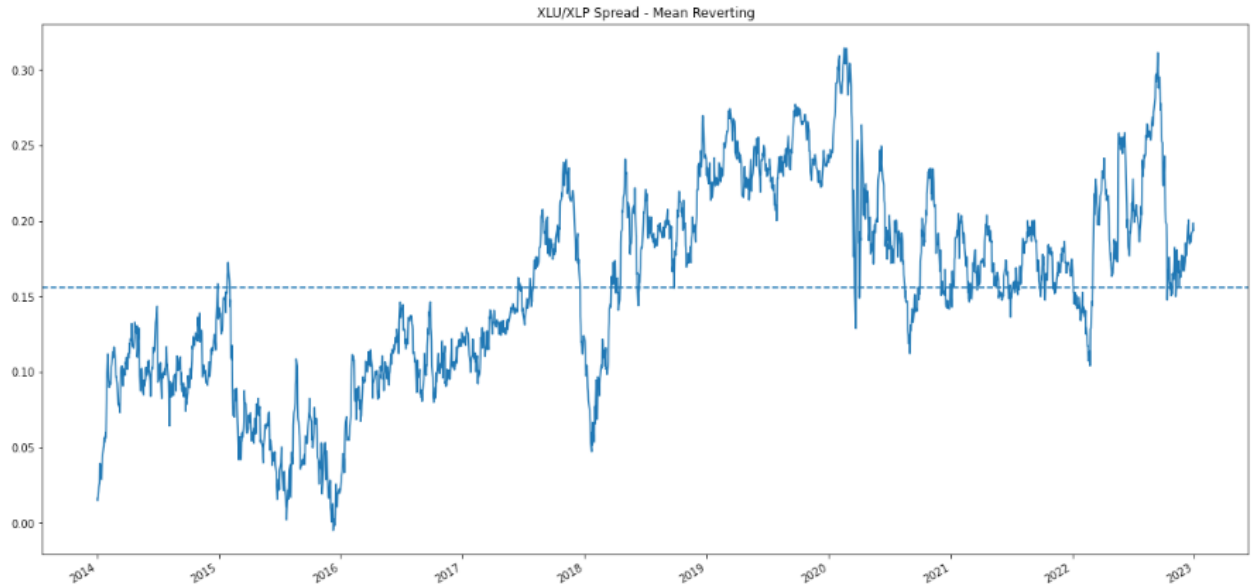
```
****** Running ADF unit root test on spread for pair:XLP,XLU ******
ADF test statistic: -4.06
        1%: -3.433
        5%: -2.863
        10%: -2.567
p-value: 0.001
The Spread for Pair:XLP,XLU is stationary
---------------------
****** Running ADF unit root test on spread for pair:XLP,XLV ******
ADF test statistic: -3.86
        1%: -3.433
        5%: -2.863
        10%: -2.567
p-value: 0.002
The Spread for Pair:XLP,XLV is stationary
---------------------
****** Running ADF unit root test on spread for pair:XLU,XLV ******
ADF test statistic: -3.41
        1%: -3.433
        5%: -2.863
        10%: -2.567
p-value: 0.011
```

Establish Baseline for the Spread

Before we dive into the various Machine learning methods, we need to establish a baseline for calculating the spread between the pair of chosen ETFs. The actual Spread was estimated using ***Ordinary Least Squares (OLS)*** which is a common technique for estimating the coefficients of linear regression between two or more variables. In Pairs trading, this coefficient needs to be estimated that will be used to combine the prices of the two assets which would then render the resulting time series stationary. This resultant series will be the Spread and the coefficient is called the Hedge Ratio. It is the # of shares of one asset to buy or sell for every 1 share of the other asset.

The Spread is defined as: *log (Asset 1 price) – n * log (Asset 2 price),* where n is the hedge ratio.

Below plot shows the Mean reverting nature of the Spread between XLU and XLP.



For comparison purposes, we also used ***ARIMA*** model to forecast the spread.

ARIMA (Auto-Regressive Integrated Moving Average) is a way of modelling time series data for forecasting (i.e., for predicting future points in the series), in such a way that:

- a pattern of growth/decline in the data is accounted for (hence the term “auto regressive”)

- the rate of change of the growth/decline in the data is accounted for (hence the term “integrated”)
- noise between consecutive time points is accounted for (hence the term “moving average”)

To predict the spread, we used ARIMA (1,0,0).

Train/Validation/Test Data set

The dataset was split as follows –

Train: 80% - Jan 2nd, 2014 – March 15th, 2021

Validation: 10% - March 16th, 2021 – Feb 3rd, 2022

Test: 10% - Feb 4th, 2022 – Dec 30th, 2022

Machine Learning Models

Three different Machine learning models were used on this data set to predict the spread. *Root Mean Squared Error (RMSE)* was used as the primary metric to compare the performance of the 3 models. Since the dataset in question is a time series, we chose the following models:

- Feed Forward Neural Network: These are Artificial Neural Networks in which the nodes do not form loops. This type of neural network is also known as Multi-layer Neural Network as all the information is only passed forward. The input nodes receive data, which then travel through the hidden layers and finally exit through the output nodes.
- Recurrent Neural Network (RNN): *They are typically suitable to solve problems with sequential input data such as time series.* A Recurrent Neural Network is a type of neural network that contains loops, allowing information to be stored within the network. In short, Recurrent Neural Networks use their reasoning from previous experiences to inform the upcoming events.
- Long Short-Term Memory (LSTM): Long Short-Term Memory Networks is an advanced RNN, a sequential network, that allows information to persist.

Hyperparameter Tuning

Different values for the following Hyperparameters were experimented with across all the 3 models chosen.

- # of hidden layers
- # of Neurons
- Epochs
- Batch size
- Activation function
- Features (lag1, lag2, lag1 and lag2)

1. Feed Forward Neural Network:

Features	Layers	Neurons	Epochs	Batch Size	Activation function	RMSE Train	RMSE Validation	RMSE Test
lag1	5	10	100	32	sigmoid	0.068660	0.028276	0.074381
lag1	5	10	100	32	relu	0.068052	0.030653	0.075631
lag1	5	10	100	32	tanh	0.069109	0.018411	0.063905
lag1	10	20	300	64	sigmoid	0.069056	0.032055	0.078008
lag1	10	20	300	64	relu	0.068019	0.026478	0.072071
lag1	10	20	300	64	tanh	0.068523	0.022806	0.068940
lag2	5	10	100	32	sigmoid	0.068546	0.024825	0.070962
lag2	5	10	100	32	relu	0.067912	0.027238	0.071206
lag2	5	10	100	32	tanh	0.068439	0.027493	0.073549
lag2	10	20	300	64	sigmoid	0.070281	0.016074	0.059640
lag2	10	20	300	64	relu	0.067282	0.025472	0.070228
lag2	10	20	300	64	tanh	0.068407	0.023397	0.069516
lag1/lag2	5	10	100	32	sigmoid	0.068611	0.027480	0.073605
lag1/lag2	5	10	100	32	relu	0.068152	0.034438	0.077900

lag1/lag2	5	10	100	32	tanh	0.068933	0.018876	0.064524
lag1/lag2	10	20	300	64	sigmoid	0.069369	0.034069	0.079915
lag1/lag2	10	20	300	64	relu	0.066261	0.024678	0.070604
lag1/lag2	10	20	300	64	tanh	0.068376	0.026281	0.072434

2. Recurrent Neural Network:

Features	Layers	Neurons	Epochs	Batch Size	Activation function	RMSE Train	RMSE Validation	RMSE Test
lag1	5	10	100	32	sigmoid	0.069440	0.017730	0.062750
lag1	5	10	100	32	relu	0.068728	0.018067	0.062992
lag1	5	10	100	32	tanh	0.068473	0.026308	0.072423
lag1	10	20	300	64	sigmoid	0.068550	0.024422	0.070551
lag1	10	20	300	64	relu	0.067334	0.028727	0.074133
lag1	10	20	300	64	tanh	0.068739	0.020529	0.066505
lag2	5	10	100	32	sigmoid	0.068562	0.023916	0.070031
lag2	5	10	100	32	relu	0.068416	0.033583	0.077896
lag2	5	10	100	32	tanh	0.068378	0.023607	0.069864
lag2	10	20	300	64	sigmoid	0.069812	0.036382	0.082092
lag2	10	20	300	64	relu	0.067322	0.021005	0.067125
lag2	10	20	300	64	tanh	0.068365	0.026325	0.072562
lag1/lag2	5	10	100	32	sigmoid	0.068620	0.027638	0.073758
lag1/lag2	5	10	100	32	relu	0.066756	0.026028	0.072672
lag1/lag2	5	10	100	32	tanh	0.068369	0.025545	0.071733
lag1/lag2	10	20	300	64	sigmoid	0.069016	0.031763	0.077730
lag1/lag2	10	20	300	64	relu	0.064100	0.028528	0.076491
lag1/lag2	10	20	300	64	tanh	0.068281	0.025843	0.071990

3. Long Short-Term Memory:

Features	Layers	Neurons	Epochs	Batch Size	Activation function	RMSE Train	RMSE Validation	RMSE Test
lag1	5	10	100	32	sigmoid	0.068565	0.026346	0.072486
lag1	5	10	100	32	relu	0.068562	0.023887	0.070001
lag1	5	10	100	32	tanh	0.068591	0.023154	0.069237
lag1	10	20	300	64	sigmoid	0.069883	0.036719	0.082408
lag1	10	20	300	64	relu	0.068559	0.026139	0.072281
lag1	10	20	300	64	tanh	0.068767	0.021011	0.066908
lag2	5	10	100	32	sigmoid	0.068765	0.029565	0.075628
lag2	5	10	100	32	relu	0.068546	0.025206	0.071347
lag2	5	10	100	32	tanh	0.068549	0.025735	0.071883
lag2	10	20	300	64	sigmoid	0.068549	0.024540	0.070671
lag2	10	20	300	64	relu	0.068553	0.024271	0.070396
lag2	10	20	300	64	tanh	0.068787	0.020845	0.066720
lag1/lag2	5	10	100	32	sigmoid	0.069217	0.018496	0.063836
lag1/lag2	5	10	100	32	relu	0.068546	0.024247	0.070375
lag1/lag2	5	10	100	32	tanh	0.068553	0.025847	0.071989
lag1/lag2	10	20	300	64	sigmoid	0.068550	0.025638	0.071781
lag1/lag2	10	20	300	64	relu	0.068577	0.026710	0.072846
lag1/lag2	10	20	300	64	tanh	0.068788	0.029806	0.075860

Results/Spread Comparison:

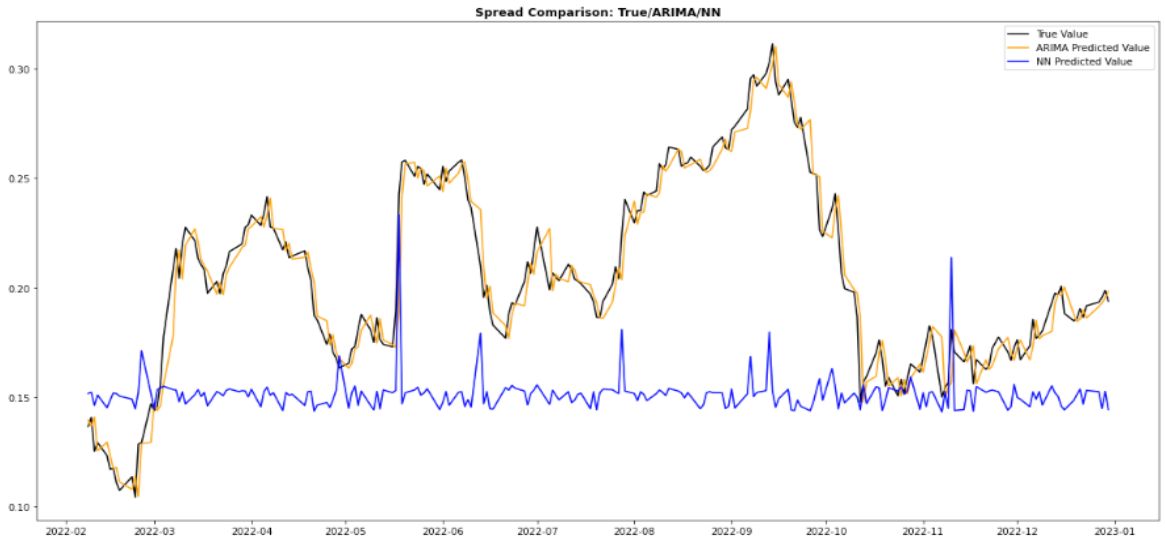
From the above data, the following set of hyper parameters seems to have lower RMSE for Feed Forward Neural Network

Features	Layers	Neurons	Epochs	Batch Size	Activation function	RMSE Train	RMSE Validation	RMSE Test
lag2	10	20	300	64	sigmoid	0.070281	0.016074	0.059640

Using this set of hyperparameters, the *Feed Forward Neural Network* model was again trained on 90% train and 10% test data. Following results were observed –

RMSE (Train data): 0.06461136
RMSE (test data): 0.06910987

ARIMA model still performed better than the Feed Forward NN. RMSE (ARIMA) = **0.0083438**



Next Steps:

As the next steps, there are 2 different work streams that can be explored.

1. Create a robust process (using Machine learning models) to identify Co-integrated pairs out of all the available stocks/ETFs.
2. Once the pair(s) is identified, we create a trading strategy to generate buy/sell signals for the pair and use Sharpe Ratio (or other measures) to evaluate the Portfolio Performance.

References:

- Machine Learning for Algorithmic Trading – Stefan Jansen
- https://hudsonthames.org/definitive-guide-to-pairs-trading/
- https://robotwealth.com/practical-pairs-trading/
- https://doi.org/10.1155/2019/3582516 Optimizing the Pairs-Trading Strategy Using Deep Reinforcement Learning with Trading and Stop-loss Boundaries.