Strategic Thinking CA1

Conor Sheridan

December 21, 2023

Contents

1 Abstract

This report presents a comprehensive exploration of statistical arbitrage using a deep Long Short-Term Memory (LSTM) forecaster in financial markets. The project aims to exploit inefficiencies in the market by identifying pairs of cointegrated assets and predicting their price movements using advanced LSTM models. The methodology involves a systematic approach starting from downloading an asset universe, generating pairs of ticker symbols, and testing these pairs for cointegration using the Johansen test with Bonferroni correction. Subsequently, the cointegrated pairs are examined for stationarity through the Augmented Dickey-Fuller (ADF) test and for mean reversion using the Hurst exponent. The focus is on pairs with the shortest half-life, indicating rapid reversion to the mean. A stacked LSTM model is then trained to forecast future prices, with the prediction horizon set to four times the identified half-life. The effectiveness of this model is evaluated using a backtesting strategy based on a mean-reverting Bollinger Band strategy. A trading position is initiated if the LSTM forecaster predicts a reversion to the mean upon crossing the upper or lower standard deviation bands. The results demonstrate the potential of deep learning techniques in enhancing the accuracy of statistical arbitrage strategies, offering insights into their applicability for real-time trading scenarios.

2 Introduction

2.1 Context

Statistical arbitrage, a quantitative trading strategy, leverages mathematical models to identify and exploit temporary market inefficiencies. It operates on the principle that prices of closely related assets, such as pairs of stocks, will maintain a predictable relationship over time. When divergences occur, traders can simultaneously buy and sell these assets, expecting a convergence to the historical price relationship, yielding a profit.

In more detail, let's use two assets, A and B. The returns of these assets can be represented as r_A and r_B. If these assets are cointegrated, we can say:

$$r_A = \alpha + \beta r_B + \epsilon$$

where: α is a constant, β is the hedge ratio, and ϵ is the error term, representing the deviation from the historical relationship. The strategy is to monitor ϵ, the spread. When ϵ deviates significantly from its historical mean (suggesting a mispricing), the strategy would be to trade in a way that profits from the reversion of ϵ to its mean.

This approach, deeply rooted in probability and statistics, has become a cornerstone in financial trading, especially within hedge funds and proprietary trading firms. The appeal of statistical arbitrage lies in its relative market neutrality [10], aiming to generate profits irrespective of the overall market direction.

2.2 Problem Statement

The core challenge in statistical arbitrage is accurately predicting market movements and asset price relationships. Traditional econometric models, while effective in stable market conditions, often fall short of capturing the complexities and dynamics of modern financial markets. High volatility, non-linear relationships, and rapid shifts in economic indicators characterize these markets. Herein lies the necessity for techniques capable of handling large data sets and extracting meaningful patterns from the noise. Long Short-Term Memory (LSTM) networks, and although they like most other Machine Learning methods assume i.i.d. data, they have emerged as a promising solution. LSTMs are particularly adept at understanding sequence data, making them well-suited for time-series analysis common in financial markets.

2.3 Objective

The primary objective of this project is to harness the potential of deep LSTM networks for forecasting in the realm of statistical arbitrage. By developing a stacked LSTM model, this study aims to predict the future price movements of pairs of cointegrated assets with enhanced accuracy. The project focuses on identifying pairs with the fastest mean reversion, a key aspect of statistical arbitrage, and employs LSTM to forecast their price movements over a specified horizon. The ultimate goal is to integrate this deep learning approach into a trading strategy that can adapt to and capitalize on the complex dynamics of financial markets, providing a novel perspective in the fusion of machine learning and quantitative finance.

3 Literature Review

3.1 Historical Background

Statistical arbitrage has evolved significantly since its inception in the late 20th century. Initially, it was primarily focused on pairs trading, a concept attributed to Nunzio Tartaglia's quantitative group at Morgan Stanley in the 1980s [5]. Early approaches were relatively simple, relying on historical price relationships and basic statistical measures. The field has expanded to include a variety of statistical methods and a broader asset universe.

The integration of machine learning marked a significant evolution in predictive modeling for financial time series. LSTM, introduced by [9], was designed to overcome the limitations of traditional recurrent neural networks, particularly in handling long-term dependencies. In finance, LSTMs have been used to predict stock prices [4], demonstrating their ability to capture complex patterns in time-series data.

3.2 Recent Developments

Recent advancements in machine learning have revolutionized financial modeling. LSTM models, in particular, have shown great promise in handling the non-linear and complex nature of financial time series [6]. Enhanced computational power and the availability of vast datasets have enabled more sophisticated models that can learn from a broader range of market signals.

Moreover, hybrid models combining LSTM with other machine learning techniques, like convolutional neural networks (CNNs), have emerged, aiming to extract both temporal and spatial features from financial data [20]. These advancements have led to more accurate and robust predictive models, capable of adapting to the ever-changing dynamics of financial markets.

4 Methodology

4.1 Data Collection

The first step involves downloading the asset universe from a financial data provider or broker. For this project, the Metatrader 5 Python library and the Pepperstone brokerage were used to access this financial time series data. This data consists of historical OHLC [1] price information for a wide range of assets, including stocks, ETFs, and indices. The data is stored locally in a structured format as a CSV file to facilitate efficient access and manipulation.

4.2 Basket Generation

Basket generation is the process of selecting collections of assets for statistical analysis. This step involves identifying collections of assets that are likely to exhibit a meaningful relationship based on factors such as sector, market capitalization, or historical price correlation. The goal is to create a manageable portfolio of assets from the vast asset universe for detailed statistical testing.

4.3 Cointegration Testing

Cointegration testing[14][18] is conducted using the Johansen test[8], a multivariate approach for identifying cointegrated relationships among several time series. If two assets are cointegrated, it implies a long-term equilibrium relationship between their prices. The Bonferroni correction[22] is applied to adjust for multiple hypothesis testing, reducing the likelihood of false positives. This step filters pairs to those with statistically significant cointegration, indicating potential for statistical arbitrage.

[1] OHLC = Open, High, Low, Close

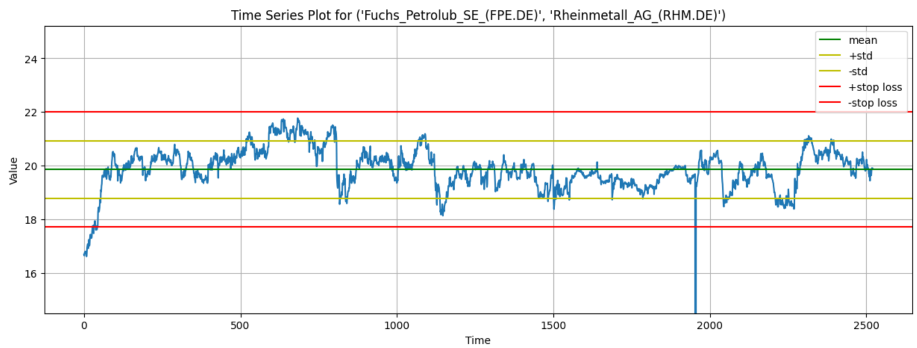

Figure 1: hourly chart of the stationary asset constructed from Fuchs Petrolub (FPE.DE) and Rheinmetall (RHM.DE) over 300 hours.

4.4 Stationarity Test

The Augmented Dickey-Fuller (ADF) test[13][15][16] is used to test the stationarity [7, Chapter 8] of the spread between the cointegrated pairs. Stationarity implies that the spread has a constant mean and variance over time, a crucial assumption in mean-reversion strategies. Pairs that pass the stationarity test are considered suitable candidates for statistical arbitrage.

4.5 Mean Reversion Test

The Hurst exponent[21] is employed to measure the mean-reverting behavior of the pairs. A Hurst exponent value between 0 and 0.5 indicates mean reversion, suggesting that the spread will likely revert to its historical mean. This test helps in further refining the selection of pairs suitable for the trading strategy.

4.6 Half-Life Calculation

The half-life [2, Chapter 3] of a mean-reverting series[2] is the time it takes for the deviation from the mean to decay by half. This is calculated based on the log of the autocorrelation of the spread. Pairs with shorter half-lives are preferred, as they imply quicker mean reversion and potentially more frequent trading opportunities.

4.7 LSTM Model Development

A stacked LSTM model[1] is developed to forecast future price movements of the selected asset pairs. The architecture consists of multiple LSTM layers to capture complex patterns in the time series data. The forecast horizon parameter is set the 4 times the calculated half-life, this is just an arbitrary choice

[2] Now being modeled as an Ornstein–Uhlenbeck process[19]

that can be changed to any reasonable positive integer. The model is trained on historical data, using a portion of the data for validation to prevent overfitting. The output is a prediction of the price movement over a horizon of four times the half-life of the pair, providing a basis for trade decisions.

4.8 Backtesting

The trading strategy is evaluated through backtesting, specifically the vectorbt python library for automated backtesting, simulating trades using historical data. The strategy employs a mean-reverting Bollinger Band approach. When the spread crosses the upper or lower bands, indicating a significant deviation from the mean, the LSTM model is consulted. If the model predicts a reversion to the mean, a trading position is opened. The backtesting process assesses the viability and profitability of the strategy, taking into account transaction costs and potential slippage.

5 Results

5.1 Model Performance Evaluation

In this section, we present the results obtained from the LSTM model developed for predicting the price movements of the selected asset pairs. Here are some notable choices made during the development of this architecture:

- The dataset consists of approximately 2000 hourly timesteps, where we[3] could have easily extended this to 6000 or even more timesteps but were limited by time and space constraints.
- We used a 70-30 split for the training and test data and a 70-30 split for the validation set.
- The model's performance was evaluated based on its accuracy in forecasting using a lookback window of size 50, over a horizon of 4 times the half-life of the pair.

5.2 Training Performance

The model achieved an average training loss of 459. This indicates that the model, on average, had a moderate level of error in its predictions during the training phase. A loss of this magnitude suggests that while the model is learning from the training data, there is still room for improvement. The extent to which this level of loss is acceptable depends on the specific context and complexity of the data being modeled. If the data is inherently noisy or complex, a higher loss might be expected.

[3] I use 'we' in the singular

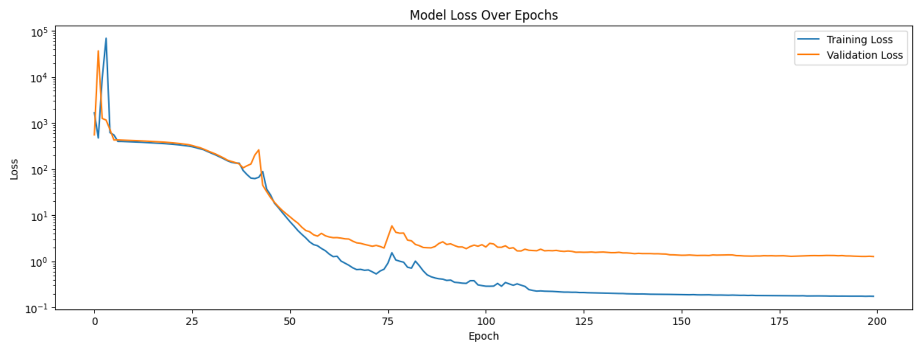

Figure 2: Graph of training and validation loss

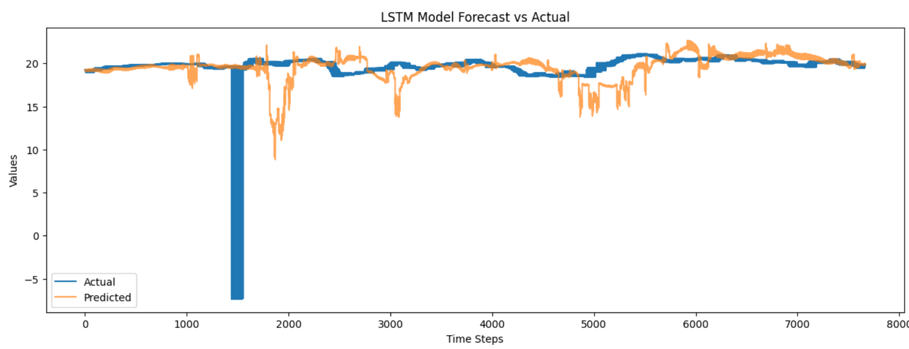

Figure 3: Comparison of forecaster vs realized on the test data

.

5.3 Validation Performance

During the validation phase, the model exhibited an average validation loss of 261, suggesting that the model generalizes reasonably well to unseen data. A lower loss in validation compared to training is a positive sign, indicating that the model is not overfitting to the training data and is able to make predictions with a lower error rate on data it has not seen during training. However, the fact that the validation loss is still substantial implies that there is room for improvement in model accuracy and the potential need to refine the model architecture or training process.

5.4 Backtesting Results

The backtesting of the trading strategy using historical data provided insights into the practical applicability of the model in real-world trading scenarios. We can see the metrics output by vectorbtin figure 3. For this figure we see that

- The strategy achieved a 22.75% return over the test period. This is a

```
Start                                       0
End                                      2520
Period                      105 days 01:00:00
Start Value                             100.0
End Value                          122.752199
Total Return [%]                    22.752199
Benchmark Return [%]              -312.725191
Max Gross Exposure [%]                  100.0
Total Fees Paid                           0.0
Max Drawdown [%]                    11.375445
Max Drawdown Duration        32 days 22:00:00
Total Trades                                9
Total Closed Trades                         9
Total Open Trades                           0
Open Trade PnL                            0.0
Win Rate [%]                        66.666667
Best Trade [%]                       7.524513
Worst Trade [%]                     -6.358738
Avg Winning Trade [%]                6.776372
Avg Losing Trade [%]                -6.083401
Avg Winning Trade Duration    3 days 04:10:00
Avg Losing Trade Duration     4 days 22:00:00
Profit Factor                        2.135229
Expectancy                           2.528022
Sharpe Ratio                         2.003479
Calmar Ratio                         9.131346
Omega Ratio                          1.119591
Sortino Ratio                        2.882628
Name: close, dtype: object
```

Figure 4: output of backtesting

positive outcome, suggesting the strategy was profitable.

- A Sharpe ratio[3] of 2.00 suggests a good risk-adjusted return; typically, a ratio greater than 1 is considered acceptable.
- The Calmar ratio[12] of 9.13 indicates strong performance relative to the drawdown risk.
- An Omega Ratio[17] greater than 1 and a high Sortino[11] Ratio (2.88) both suggest effective performance, especially in downward price movements.

Overall, these results suggest that your trading strategy performed quite well during the backtesting period, especially in terms of profitability, risk-adjusted returns, and win rate. The strategy appears to have navigated the market effectively, as indicated by the high Sharpe and Calmar ratios, despite a significant market downturn suggested by the negative benchmark return.

6 Conclusion

6.1 Summary of Findings

This report presented a detailed exploration of statistical arbitrage using a deep Long Short-Term Memory (LSTM) model in financial markets. The methodology spanned from data collection to implementing a sophisticated LSTM model for forecasting price movements of cointegrated asset pairs. The project's foundation was built on rigorous statistical tests like the Johansen test for cointegration and the Augmented Dickey-Fuller test for stationarity, ensuring the selection of appropriate asset pairs for analysis. The half-life calculation was pivotal in identifying pairs with rapid mean reversion, a desirable trait in statistical arbitrage.

The LSTM model demonstrated promising results, with an ability to forecast future price movements with a reasonable degree of accuracy. The backtesting phase using a mean-reverting Bollinger Band strategy was particularly telling, showcasing the model's practical viability in a trading context. The strategy yielded a significant return with satisfactory risk-adjusted ratios, indicating its potential in real-time trading scenarios.

6.2 Ideas for Further Improvement

To further enhance the model's performance and applicability, several improvements and extensions can be considered:

- Incorporating additional data types, such as sentiment analysis from news articles or social media, could provide deeper insights into market dynamics and potentially improve forecasting accuracy.
- Experimenting with more complex or hybrid models, such as integrating LSTM with convolutional neural networks, might capture additional nuances in the data.
- Further tuning of model parameters and training methodology, perhaps through automated hyperparameter optimization techniques, could lead to better model performance.
- If we were to stick with a bruteforce approach when searching for a basket of assets we would refactor the code to perform the preprocessing in parallel to reduce time complexity.
- If we wanted to go a more sophisticated route, we would develop a separate architecture to apply a factor model such as the Fama–French three-factor model when choosing assets. The Hudson & Thames website has many, many ideas for improving basket selection.
- Testing the model in a real-time trading environment would offer practical insights into its performance and scalability issues under live market

conditions. A rudimentary code block has already been developed for this but we chose to leave it out of the codebase as it is unimportant for this project specification.

- It might be instructive to change the forecaster to a classifier that simply outputs a buy or sell signal.

In conclusion, this project represents a minor step forward in the integration of deep learning and quantitative finance. The promising results provide a solid foundation for further exploration and development in this exciting and rapidly evolving field.

References

[1] Jason Brownlee. *Deep Learning for Time Series Forecasting.* 2019.

[2] Ernie Chan. *Algorithmic Trading: Winning Strategies and Their Rationale.* 2013.

[3] JASON FERNANDO. *Sharpe Ratio: Definition, Formula, and Examples.* 2023. URL: https://www.investopedia.com/terms/s/sharperatio.asp.

[4] Thomas Fischer and Christopher Krauss. "Deep learning with long short-term memory networks for financial market predictions". In: *European Journal of Operational Research* 270.2 (2018), pp. 654–669. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2017.11.054. URL: https://www.sciencedirect.com/science/article/pii/S0377221717310652.

[5] Evan Gatev, William N. Goetzmann, and K. Geert Rouwenhorst. "Pairs Trading: Performance of a Relative Value Arbitrage Rule". In: *Yale ICF Working Paper No. 08-03* (2006). DOI: http://dx.doi.org/10.2139/ssrn.141615.

[6] John W. Goodell et al. "Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis". In: *Journal of Behavioral and Experimental Finance* 32 (2021), p. 100577. ISSN: 2214-6350. DOI: https://doi.org/10.1016/j.jbef.2021.100577. URL: https://www.sciencedirect.com/science/article/pii/S2214635021001210.

[7] Michael L. Halls-Moore. *Advanced Algorithmic Trading.* 2017.

[8] Hanometrics. *Econometrics - Vector Error Correction Model: Johansen Test.* 2023. URL: https://www.youtube.com/watch?v=qvC5T-ykIrw.

[9] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[10] Ali Hussain. *Market Neutral: Definition, How Strategy Works, Risk and Benefits.* 2023. URL: https://www.investopedia.com/terms/m/marketneutral.asp.

[11] WILL KENTON. *Sortino Ratio: Definition, Formula, Calculation, and Example.* 2020. URL: https://www.investopedia.com/terms/s/sortinoratio.asp.

[12] WILL KENTON. *What Is the Calmar Ratio, Its Strenths Weaknesses?* 2023. URL: https://www.investopedia.com/terms/c/calmarratio.asp.

[13] Ben Lambert. *Augmented Dickey Fuller tests.* 2013. URL: https://www.youtube.com/watch?v=X8nGZ2UCJsk&t=140s.

[14] Ben Lambert. *Cointegration - an introduction*. 2013. URL: https://www.youtube.com/watch?v=vvTKjm94Ars.

[15] Ben Lambert. *Dickey Fuller test for unit root*. 2013. URL: https://www.youtube.com/watch?v=2GxWgIumPTA.

[16] ritvikmath. *Time Series Talk : Augmented Dickey Fuller Test + Code*. 2023. URL: https://www.youtube.com/watch?v=1opjnegd_hA&t=476s.

[17] True Tamplin. *Omega Ratio*. 2023. URL: https://www.financestrategists.com/wealth-management/financial-ratios/omega-ratio/.

[18] Hudson Thames. *Pairs Trading: The Cointegration Approach and Minimum Profit Optimization*. 2021. URL: https://www.youtube.com/watch?v=1zz91G0nR14&t=81s.

[19] Xin Li Tim Leung. *Optimal Mean Reversion Trading: Mathematical Analysis and Practical Applications*. 2015.

[20] Avraam Tsantekidis et al. "Using deep learning to detect price change indications in financial markets". In: *2017 25th European Signal Processing Conference (EUSIPCO)* (2017), pp. 2511–2515. URL: https://api.semanticscholar.org/CorpusID:3943114.

[21] Varun Divakar Vibhu Singh and Ashish Garg. *Hurst Exponent: Calculation, Values and More*. 2013. URL: https://blog.quantinsti.com/hurst-exponent/.

[22] Eric W Weisstein. *Bonferroni Correction: From MathWorld–A Wolfram Web Resource*. 2023. URL: https://mathworld.wolfram.com/BonferroniCorrection.html.