Statistical Arbitrage in Pairs Trading

1 Abstract

This paper aims to exploit statistical arbitrage opportunities by constructing a pairs trading strategy using two indices from Dow Jones. We first explain our pair selection process, then construct a copula and LSTM models to develop statistical criteria for identifying trading signals. Based on these criteria, we separate our data into training and testing sections and set up our trading strategy, which we then apply to the testing data set. In the last part of the paper, we discuss our results and the intuition behind them and identify areas for further improvements.

2 Introduction

"Markets can remain irrational longer than you can remain solvent" is a famous quote attributed to John Maynard Keynes. Many investors would recognize LTCM as an example of this adage, as the firm failed in the face of liquidity risks. Yet, LTCM had generated relatively high returns in their initial years via their primary trading strategy, which was to exploit mispricings – ironically, often arising because of differences in liquidity – between pairs of bonds sharing common fundamentals, which is an example of pairs trading.

A pairs trading strategy is a convergence strategy in which investors seek to identify temporary mispricings of two highly correlated assets and then obtain profits from buying low and selling high [13].

The methods and techniques for pairs trading, such as the distance and cointegration approaches [10], are traditionally based on mean-reversion concepts [2]. More recently, copula modeling techniques have been applied to capture non-linear tail-move relations between various equities [5][12].

In the paper, we investigate a pairs trading strategy applying copula and machine learning techniques to two Dow Jones sector equity indices: the Dow Jones U.S. Pharmaceuticals Index (DJUSPR) and the Dow Jones U.S. Health Care Index (DJUSHC).

3 Pair Selection

This section summarizes the underlying data selection criteria for a pair of equity indices.

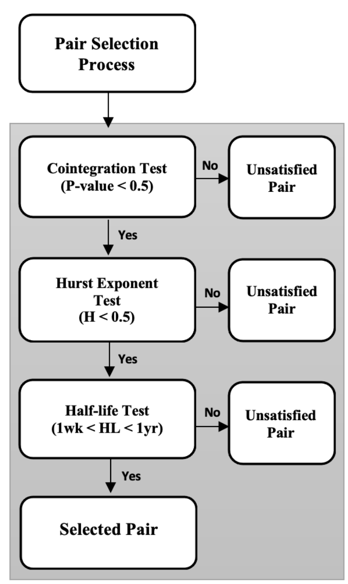

Figure 1: Pair Selection Procedure

The data used in this paper are daily close prices, rolling 200-day simple moving averages of those close prices, and daily volumes of the selected pairs of indices sourced from Bloomberg Terminal. We select some sector equity indices from SP 500 and Dow Jones Industrial Indices and equity indices in the Chinese market [1]. The dataset was chosen to cover eighteen years, from January 1st, 2005, to December 31st, 2022, to ensure a broad range of economic and market cycles is represented.

Below is the four-step pair selection process, which follows the procedure illustrated in Figure 1 (above).

3.1 Data Normalization

We perform a data normalization process by dividing the daily prices by the 200-day simple moving average (SMA200) to smooth and construct a stationary data set for each index pair. We compute the daily and normalized daily returns and denote the normalized prices of the indices as S_{1t} and S_{2t}.

3.2 Cointegration Test

Cointegration offers a means to identify a stationary linear relationship between multivariate time series. We tested for cointegration between index pairs using an Engle-Granger two-step approach [6]. The beta of a linear combination of two indices' normalized price series has to be stationary and explained by the formula below,

$$S_{1t} = \mu_t + \beta S_{2t}$$

We then run an ordinary least squares (OLS) regression to estimate beta based on two normalized price time series data. Lastly, we perform a stationary test on the estimated residual term μ_t. If the p-value of the regression is less than 0.05, the pair will pass the cointegration test.

3.3 Hurst Exponent Test

We need to find the Hurst Exponents (H) of the selected indices [4], which can help us identify whether the normalized price time series would follow a short and long-term mean-reversion pattern. The Hurst Exponent testing function is described in the equation below,

$$\mathbb{E}[\frac{R_i(n)}{S_i(n)}] = C \cdot n^H, \text{ where } n \to \infty, \; i = 1, 2$$

3.4 Half-Life Test

Continuing from the data process above, we need to find the mean reverting half-life of the indices that passed through the Hurst Exponent test [4]. The half-life measures the time period that a price series of index reverts to half of the difference between the divergent value and mean. The half-life is calculated with the equation below,

$$\Delta = p(t) - p(t-1)$$
$$p_{t-1} = \alpha + \beta\Delta + \epsilon$$
$$\text{where } \Delta, \epsilon \text{ are vectors}$$

We construct a series of differences in the daily normalized price. The next step is creating a daily normalized price series with one lag. Then we construct a regression of lagged price dependent on delta (independent) and result in the least-square solution of the regression. Lastly, we compute the half-life of each index in the pair with the formula below:

$$t_{\frac{1}{2}} = \frac{ln(2)}{\beta}$$

If the calculated half-time of each index is longer than one week and shorter than one year, the pair will pass the test for half-life. It is optimal for our analysis since it is relatively easy for the spread to converge. It gives the pairs trading strategy enough time to capture the fluctuations in spread movements.

3.5 Selection Process Return

	HC	Pharm	p-value
Hurst Exponent	0.26	0.27	
Half Life	46.7	52.8	
Cointegration			4.21e-05

Based on the above-mentioned metrics, we selected Dow Jones U.S. Pharmaceuticals Index (DJUSPR) and Dow Jones U.S. Health Care Index (DJUSHC) is the most significant paired price series to perform our pairs trading strategy.

The above chart illustrates that this pair passes our selection metrics, as it has a low p-value for the cointegration test, and the two indices both have low Hurst exponents. DJUSHC (HC) has a half-life of 46.7, and DJUSPR (Pharm) has a half-life of 52.8. The half-life of HC and Pharm not only passes our test but also is relatively similar, which means the two indices have a similar trend of divergence and convergence.

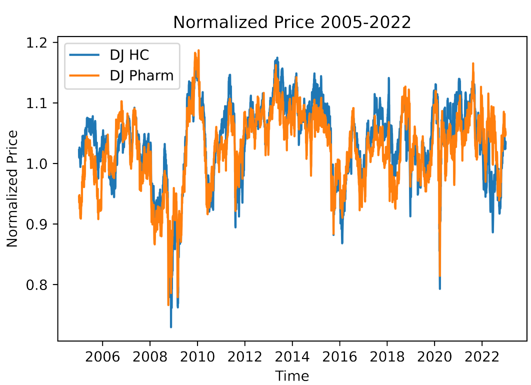

Figure 2: Normalized Price by 200SMA

4 Methodology

This section summarizes the construction of a copula model applying the Gaussian Copula approach, a machine learning model operating the Long short-term memory (LSTM) method to determine trading signals and the development of a pairs trading strategy to capture returns from the mispricing opportunities [7].

4.1 Copula Methodology

Firstly, we use t-distributions to approximate the normalized price distributions [7]. A linear trading strategy can only generate accurate trading signals when the price series have normalized distribution, which rarely happens in the financial markets. We make use of a copula approach to identify dependency features and trading signals for our pairs trading strategy to better account for both linear and non-linear dependencies between two selected indices [13].

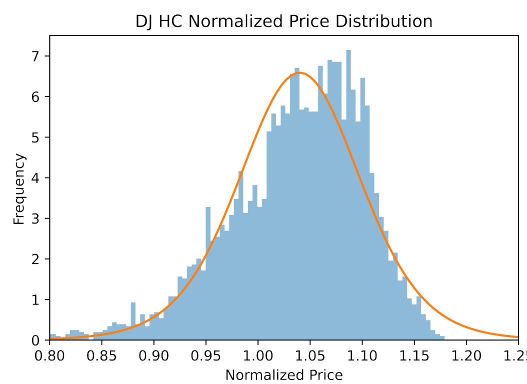

Figure 3: DJUSHC Fitted t-distribution

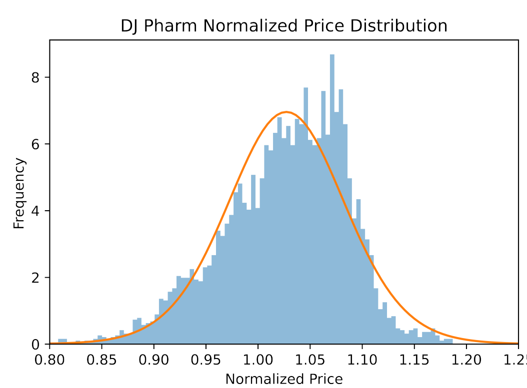

Figure 4: DJUSPR Fitted t-distribution

We denote the normalized daily returns of our two indices by $R_{HC}(t)$ and $R_{Pharm}(t)$, respectively. Based on Sklar's theorem (1958), we can derive a copula function C between the fitted continuous marginal distribution functions F_{HC} and F_{Pharm}. The corresponding joint distribution function J of the return series is below,

$$C(F(R_{HC}), F(R_{Pharm})) = J(R_{HC}, R_{Pharm})$$

We use Python's statsmodels and copulas packages to fit a Gaussian copula to the joint distribution of the two normalized index returns, as illustrated in Figure 5. The simulated distribution works sufficiently with the observed return distribution.

The next step is to find the conditional probability function of the two indices' returns $F_{HC|Pharm}(HC)$ and $F_{Pharm|HC}(Pharm)$ below,

$$F_{HC|PR}(HC) = P(R_{HC} < r_{HC} \,|\, R_{PR} = r_{PR})$$

$$F_{PR|HC}(PR) = P(R_{PR} < r_{PR} \,|\, R_{HC} = r_{HC})$$

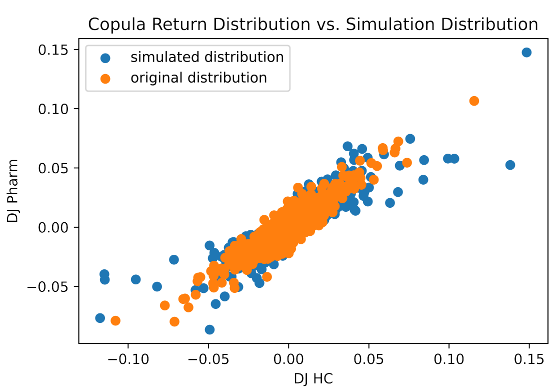

Figure 5: Simulated Distribution Copula and Simulated Spreads

The conditional probabilities can be calculated from the first-order partial derivatives of the copula function below,

$$\frac{\partial C}{\partial F(R_{HC})}, \frac{\partial C}{\partial F(R_{Pharm})}$$

Since the statistical tools could not discover the exact joint distribution and the copula function in Python, we developed an alternative method to simulate the signal-selecting process. We utilize the correlation and degree of freedom generated by the copula to fit t-distributions of returns of the two indices. The simulated spreads will be computed as the difference between simulated returns. The spread series fit well compared to the observed spreads, as illustrated in Figure 6. We will use the simulated spreads to design the trading signals in our strategy, which will be explained further in the Trading Signal part.

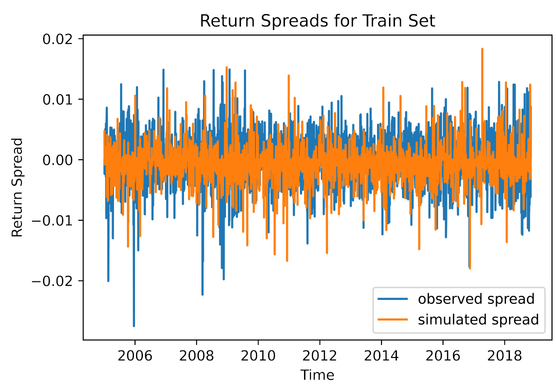

Figure 6: Simulated Spread and Original Spread

4.2 Machine Learning

In recent years, Neural network approaches have been widely used to handle financial data due to their strong ability to apply pattern recognition. Furthermore, Recurrent Neural Network (RNN) is the most suitable structure in our case due to the ability to incorporate calculations from the previous times in the sequence [8].

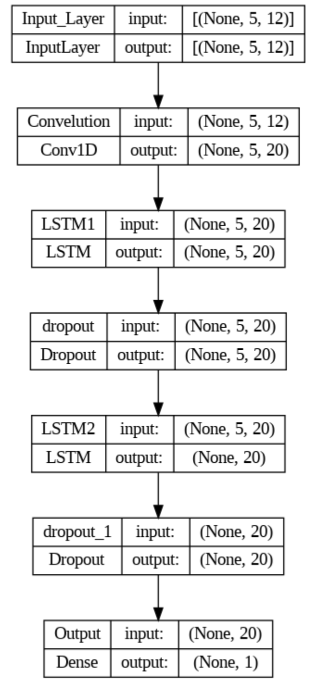

Figure 7: LSTM Model Process

Among the different RNN structures, LSTM has been proven effective in most scenarios due to its delicate architecture design [11]. The design of units, such as the input gate, forget gate, and output gate, has made the LSTM capable of remembering past calculations, to utilize the more effective calculations to improve the overall loss function. So, the LSTM can unravel the issue of vanishing gradients by improving the model's overall structure [11].

Thus, we choose LSTM as our model for realized volatility prediction on our selected indices, HC and Pharm. Compared with the machine learning approaches that assume a linear relationship, the LSTM considers a nonlinear relationship that incorporates variables from both current and historical time. The return movements of the indices are considered time series with certain levels of autocorrelation. Thus, the LSTM method is a good choice for handling such data since it can capture the relationship between different time points of the sequence.

We design a feature engineering process. For instance, we calculate the intercept α and coefficient β of a linear regression model as the return depends on the volatility of HC and Pharm. We use the computed intercept and coefficient as features in LSTM while we consider the returns of indices will follow a random walk process.

Alpha is the drift term μ, and beta is the volatility term σ. In LSTM, our team applied lagged volatility, volume, and other characteristics as model features. We then preprocess the data by eliminating outliers, using mean to supersede NaN (Not a Number) values, and normalizing all the features.

The input layer is a 5×12 matrix, which means the LSTM will consider five days of data in all twelve selected features. The inputs are arranged into a one-dimensional Convolution layer, where we conduct 1-D and horizontal feature extractions from the original inputs. Then, the new features will be implemented into the first LSTM layer in the next step. The outputs generated from our LSTM model will be transferred into a Dropout layer. We randomly assigned input to zero in the first Dropout layer to prevent our neuron network from memorizing a fixed pattern. Then, the new output generated from the first Dropout layer will be implemented into the second LSTM and the second Dropout layer afterward. Lastly, we place the data into a dense layer and build estimated volatilities as the model outputs. We apply our LSTM model on both indices separately.

We used mean squared error (MSE) as the evaluation metric for the machine learning model's performance. Compared with other metrics, MSE has a relatively higher sensitivity to large errors, which will help us ensure we don't generate outlier predictions that misguide our trading activities. Eventually, our model gets a loss value of 64, shown in Figure 8, and the fitting process indicates that the loss value is gradually decreasing.

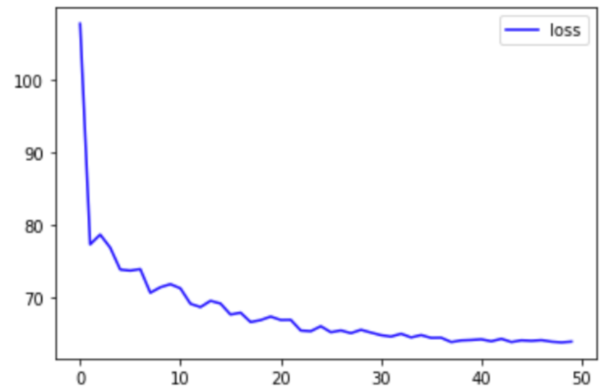

Figure 8: Loss Function

4.3 Trading Signal

We use the Gaussian Copula and LSTM Methods to identify trading signals. As the algorithm simulates new Student's t-distributions based on the copula outputs [7], the corresponding Z-score is computed to determine whether a trading signal exists in our paired indices.

$$z = \frac{\text{observed} - \mathbb{E}(\text{simulated})}{\sigma_{\text{simulated}}}$$

If the Z-score is outside the 99% confidence interval, ± 2.58, the algorithm can identify divergence signals between the two selected indices. If the data achieves the Z-score requirement, we utilize the daily estimated volatilities from the LSTM model. After the divergence is detected by achieving the Z-score requirement, we utilize the daily estimated volatilities from the LSTM model to construct an F-test [3] with a 95% confidence interval, function is shown below,

$$\text{F-Score} = \frac{\sigma^2_{HC}}{\sigma^2_{Pharm}},$$

where $\sigma^2_{HC}, \sigma^2_{Pharm}$ are variances of the two indices.

We use F-test to identify if there is a statistically significant difference between the volatilities of the two indices. The correlation between the two indices is 0.93 based on the data we have in the training set, which means the two indices have similar driving factors. We believe the fluctuations in the volatilities constitute a significant cause of the divergence besides fundamental changes happening in the market.

After both conditional tests, if the P-value is less than 0.05, the trading signal will be 1. Otherwise, it will be 0.

4.4 Trading Strategy

We divide the data into training and testing sets to ensure both contain excited and calm market periods. We utilize data from January 1st, 2005, to December 31st, 2017, as the training set; data from January 1st, 2018, to December 31st, 2022, as the testing set.

We use these normalized return data to construct the copula model. We will re-estimate the copula model for each day in the testing sample with a new pair of daily returns. In this approach, the correlation between the two indices can be automatically updated. We also simulate new t-distributions and encounter the corresponding Z-scores to determine whether a trading signal exists daily in the testing set.

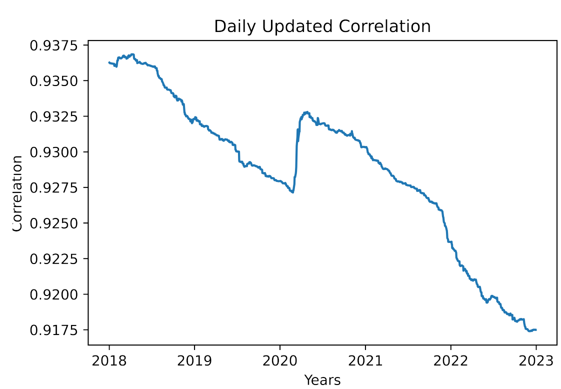

Figure 9: Correlation for the Test Set

Figure 9 shows the dynamic correlations of the testing set, where an unexpected surge of daily correlation of 0.005 occurred during the beginning of the COVID Pandemic. The chart indicates an average correlation of 0.93 over the testing set. Yet, over the entire testing data set, the correlation only changes by 0.02, which we consider insignificant. From the analysis of the dynamic correlation generated by our algorithm, we conclude that only one of the selected pairs will generate significant price movement at each time.

For simplicity, at the beginning of our trading period, we ensure the nominal values of DJUSHC (HC) and DJUSPR(Pharm) are the same, e.g., $1 for each index. In our algorithm, the difference between the daily returns of HC and Pharm will be generated, noted as the absolute spread of returns.

$$\text{Abs Spread} = |\text{HC Ret}_{Daily} - \text{Pharm Ret}_{Daily}|$$

The strategy considers successive daily returns until a mispricing signal is observed. As the trading signals are detected, we iterate over the normalized prices of the paired indices to determine the directions of the trading position. For example, if the normalized price of HC > Pharm, the direction will be noted as 1; otherwise, the direction will be noted as -1. With the multiplication between the absolute spread calculated by the above function and the direction signal, we could ensure our trading strategy follows the "buy low, sell high" method, as the trade will always go long the low return index and short the high return index.

The strategy continues to operate while the two indices remain in divergence. In this case, we keep our positions open and record the cumulative return during the period. Once the algorithm indicates an exiting signal, where the trading signal turns to 0 when the price movement of the paired indices converges back, we close out our positions and wait for the next trading signal. To specify the convergence condition of the paired indices, the strategy will process the tests mentioned above as the Z-score will fall back to the 99% confidence interval, and the F-test will be passed as the P-value larger than 0.05.

To ensure our strategy remains solvent, we designed a stop-loss trigger for our pairs trading strategy to protect us from unexpected market developments. The implementation of the stop-loss trigger is to utilize the cumulative return during the trading cycle. If the cumulative return in the trading cycle is larger than -25%, the algorithm will immediately close the positions regardless of the trading signal remaining as 1. From the previous half-life test in the pair selection section, we identify the mean reversion half-life of the HC and Pharm as 46.7 and 52.8 days. As the pairs trading strategy helps to hedge sector and market risk, we remain risk averse. We tracked the 5-year annualized historical volatility of the S&P 500 from the Bloomberg Terminal as 17.17% and set the risk-averse constant to 2. With the function below,

$$\text{Stop Loss} = \frac{\text{Daily Volatility} \times \text{Half-Life}}{\text{Risk Aversion}}$$

This stop-loss trigger can immunize us from potentially significant losses in edge cases. For instance, if the two indices take an unexpectedly long time to converge compared to the expected mean-reverting half-life, our stop-loss trigger will effectively mitigate our strategy away from the unexpected market volatility.

5 Strategy Results

In this section, we will interpret the results of our trading strategy applied in the testing set.

Cumulative returns, and drawdowns with

different confidence intervals, are shown below

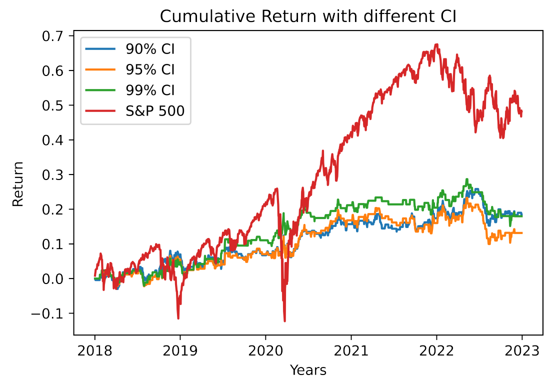

Figure 10: Cumulative Returns of the Portfolio with Different CI

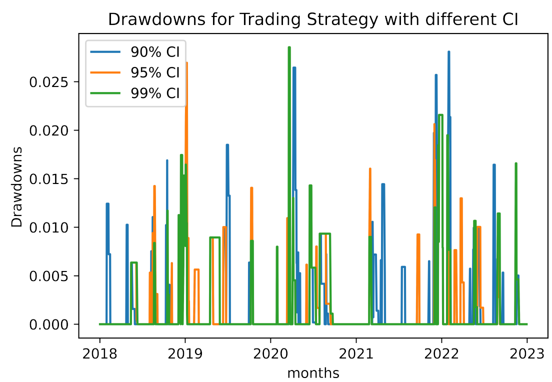

Figure 11: Drawdowns with Different CI

Based on the 99% confidence level copula model and 95% confidence level F-test model on volatilities discussed above, we identified 208 trading signals in the testing set. After applying our trading strategy, we obtained a 5-year cumulative return of 17.96%, as illustrated in Figure 13. The average annualized return is approximately 3.36%, benchmarked to the S&P 500 index average Index 5-year annual price return of 8.18%, as shown in Figure 10.

We also compute some performance metrics for our trading strategy. Assuming the risk-free rate is zero, the daily Sharpe ratio is

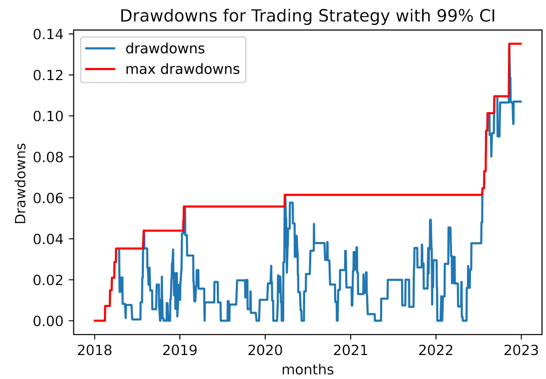

Figure 12: Drawdowns with 99% CI

0.0298 and annualized Sharpe ratio is 0.473. The max drawdown during the testing period is illustrated in Figure 12. The highest max drawdown appears in 2022 due to the high volatility in the market, with a value of 13.8%.

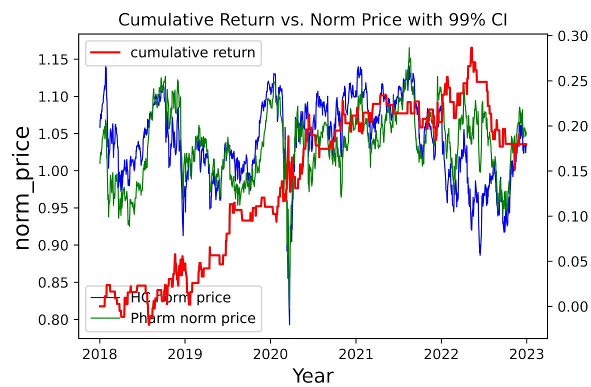

Figure 13: Cumulative Return and Normalized Price with 99% CI

6 Discussion

This section analyzes the results generated from our pairs trading strategy and compares the results for different confidence level scenarios.

Although our cumulative return is less than the return of S&P 500 index, we believe a trading strategy is a useful tool for risk diversification in investing in a portfolio containing financial instruments related to the S&P 500 index. We computed the correlation between the returns of our trading strategy and the returns of S&P 500 index. The correlation is -0.1761, showing a negative relationship. In Figure 10, we can see that, when S&P 500 index experienced a sharp decrease in cumulative returns in 2019 and 2020, the cumulative return of our strategy retains a stable increasing pattern.

We experimented with our trading strategy with 90% and 95% confidence intervals to compare with our choice of using 99%. The algorithm generates 412 and 320 trading signals, respectively. The cumulative return under these two cases is presented in Figure 10. The 99% confidence scenario has the highest cumulative return because only a large divergence can pass the Z-score test for 99% confidence level, resulting in the most accurate detection of trading signals. The cumulative return of 90% confidence interval is higher than that of 95% confidence interval. We believe there will be more false positive signals if we widen the confidence level.

7 Future research

The large number of hyperparameters in the LSTM model leads to a slow training process. Thus, grid search and random search are not suitable for the hyperparameter-tuning process of LSTM. In the future, we can use the Bayesian method for the tunning process. The Bayesian method finds the optimal hyperparameter combination by estimating a prior probability distribution of the optimal hyperparameters and continues to update by training the model on different values to get the model's actual performance [9]. This method reduces the number of hyperparameter scenarios that need to be tried. In addition to this, we can also try more types of non-linear copula functions to find the best function that could describe the joint probability distribution of the two indexes.

References

[1] Avellaneda, Marco, and Jeong-Hyun Lee. *Statistical Arbitrage in the US Equities Market*. Quantitative Finance, vol. 10, no. 7, Aug. 2010, pp. 761–82. https://doi-org/10.1080/14697680903124632.

[2] Bao, Yong, et al. *Distribution of the Mean Reversion Estimator in the Ornstein–Uhlenbeck Process*. Econometric Reviews, vol. 36, no. 6–9, July 2017, pp. 1039–56. https://doi-org/10.1080 /07474938.2017.1307977.

[3] Bogomolov, Timofei. *Pairs Trading Based on Statistical Variability of the Spread Process*. Quantitative Finance, vol. 13, no. 9, Sept. 2013, pp. 1411–30. https://doi-org/10.1080/14697688.2012.748934.

[4] Cho, Poongjin, and Minhyuk Lee. *Forecasting the Volatility of the Stock Index with Deep Learning Using Asymmetric Hurst Exponents*. Fractal Fractional, vol. 6, no. 7, July 2022, p. 394. https://doi-org/10.3390 /fractalfract6070394.

[5] Krauss, Christopher. *Statistical arbitrage pairs trading strategies: Review and outlook*. IWQW Discussion Papers, No. 09/2015, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institut für Wirtschaftspolitik und Quantitative Wirtschaftsforschung (IWQW), Nürnberg, 2015.

[6] Lee, Hyejin, and Junsoo Lee. *More Powerful Engle–Granger Cointegration Tests*. Journal of Statistical Computation

and Simulation, vol. 85, no. 15, 2014, pp. 3154–3171., https://doi.org/10.1080 /00949655.2014.957206.

[7] Luo, Xiaolin, and Pavel V. Shevchenko. *The t Copula with Multiple Parameters of Degrees of Freedom: Bivariate Characteristics and Application to Risk Management*. Quantitative Finance, vol. 10, no. 9, Nov. 2010, pp. 1039–54. https://doi-org/10.1080 /14697680903085544.

[8] Pawar, K., Jalem, R.S., Tiwari, V. *Stock Market Price Prediction Using LSTM RNN*. Advances in Intelligent Systems and Computing, vol 841. Springer, Singapore. https://doi.org/10.1007 /978-981-13-2285

[9] Putatunda, Sayan, and Kiran Rama. *A modified bayesian optimization based hyper-parameter tuning approach for extreme gradient boosting*. 2019 Fifteenth International Conference on Information Processing (ICINPRO). IEEE, 2019. https://doi.org/10.1016 /j.sigpro.2022.108826

[10] Rad, Hossein, et al. *The profitability of pairs trading strategies: distance, cointegration and copula methods*. Quantitative Finance, vol. 16, no. 10, Oct. 2016, pp. 1541–58. https://doi-org/10.1080 /14697688.2016.1164337.

[11] Sherstinsky, Alex. *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*. Physica D, vol. 404, Mar. 2020, p. N.PAG. https://doi-org/10.1016 /j.physd.2019.132306.

[12] Stübinger, Johannes, et al. *Statistical Arbitrage with Vine Copulas*. Quantitative Finance, vol. 18, no. 11, Nov. 2018, pp. 1831–49. https://doi-org/10.1080 /14697688.2018.1438642.

[13] Nadaf, Tayyebeh, et al. *Revisiting the Copula-Based Trading Method Using the Laplace Marginal Distribution Function*. Mathematics (2227-7390), vol. 10, no. 5, Mar. 2022, p. 783. https://doi-org/10.3390 /math10050783.