Integrating Weather Data into Deep Learning-Based Statistical Arbitrage
Deep Learning (STAT3007/7007)
Report - Semester 1, 2025.

Filip Orestav	Hans Stemshaug	Nila Saravana	Volter Entoma	Weiming Liang
49316997	49060423	48773799	44782711	46375489

May 31, 2025

1 Introduction

Statistical Arbitrage is a trading strategy that aims to profit from the relative price movements of two or more assets. It involves identifying mispricings in the market and taking advantage of them by simultaneously buying and selling the assets. Statistical arbitrage is based on the idea that pairs or groups of historically similar stocks are expected to maintain their statistical relationship in the future, allowing traders to exploit temporary deviations from this relationship. (Avellaneda and Lee, 2010)

The goal for this project is to develop a deep learning model that can apply Statistical Arbitrage to identify unfairly priced electricity prices in the European market, and to create a profitable trading strategy.

2 Literature Review

(Guijarro-Ordonez et al., 2021)

- There model we use in this report
  - Architecture design: what are the components, the purpose of their components
  - Main findings of the paper and how that's relevant to our current study
  - Insert a figure of their architecture
  - Discuss which part of their model and we're going to use and why
- https://ieeexplore.ieee.org/abstract/document/9383387 "Electricity price forecast based on weekly weather forecast and its application to arbitrage in the forward market" (This paper should have something about how weather affects the supply and demand of electricity and hence the prices)
- success stories of incorporating different data source into statistical arbitrage and get good results

3 Theory

3.1 Statistical Arbitrage: Cointegration

Cointegration is a statistical property of time series variables that indicates a long-term equilibrium relationship between them, even though the individual series may be non-stationary. In statistical arbitrage, cointegration is used to identify portfolios where a linear combination of asset prices or returns results in a stationary residual, enabling mean-reverting trading strategies.

Consider N assets with log cumulative returns $R_{1,t}, R_{2,t}, \ldots, R_{N,t}$. These assets are said to be *cointegrated* if there exists a vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_N)$ such that the linear combination

$$e_t = \beta_1 R_{1,t} + \beta_2 R_{2,t} + \cdots + \beta_N R_{N,t}$$

or, more compactly,

$$e_t = \boldsymbol{\beta}^\top \mathbf{R}_t \tag{1}$$

where $\mathbf{R}_t = (R_{1,t}, R_{2,t}, \ldots, R_{N,t})^\top$, is a *stationary process*.

The stationary residual e_t oscillates around a constant mean. When e_t deviates significantly from the mean, traders take positions assuming it will revert.

- **Long** the undervalued assets.
- **Short** the overvalued assets.

For example, if $e_t > \theta$, where θ is a threshold, it indicates that the portfolio is overvalued. Traders would short the portfolio, expecting the price to revert to its mean. Conversely, if $e_t < -\theta$, it indicates that the portfolio is undervalued, and traders would go long on the portfolio.

The key assumption of cointegration is that the linear combination of asset prices or returns is stationary, even if the individual series are not. This means that the spread between the assets will revert to its mean over time, allowing traders to profit from temporary deviations from this mean.

In the context of European electricity prices, we expect the prices in Europe to be stationary, as there are balancing effects of supply and demand. Many European countries are connected via cross-border transmission lines, allowing to equalize prices across interconnected countries. Additionally, neighbouring regions are expected to experience similar weather patterns, and seasonal demand, leading to correlated markets. Lastly, European energy policies and regulations are often harmonzied, reducing energy structural difference between countries.

3.1.1 Log returns

We take the log of the cumulative returns of the time series for each asset. This is done to capture any compounding differences between the assets and stabilizes large variances. Additionally, taking the cumulative returns instead of the raw prices allows us to capture the relative change in price over time, normalizing the data and making it easier to compare across different assets. Cointegration between log returns is more likely to be stationary than cointegration between raw prices.

Specifically, the log return of asset n at time t:

$$R_{i,t} = \ln\left(\frac{P_{i,t}}{P_{i,0}}\right) \tag{2}$$

Where $P_{i,t}$ is the price of asset i at time t and $P_{i,0}$ is the price of asset i at time 0. The log return is a measure of the relative change in price over time.

3.1.2 Cumulative residuals

We calculate the residual of each asset by regressing the log returns of the asset on the log returns of the other assets:

$$\varepsilon_{n,t} = R_{n,t} - \sum_{i=1}^{N} \beta_{n,i} R_{i,t} \tag{3}$$

An input to the deep learning model are cumulative residuals. The cumulative sum mimics a *price-like* behavior, which is more suitable for identifying trading signals. (Needs citation here) Raw residuals may not capture sufficient trend information, while cumulative forms highlight *deviations* more clearly.

The cumulative residuals are calculated by integrating the time series of residuals over a rolling window of L days.

For asset n, define $\varepsilon^{L}_{n,t-1}$ as the vector of the past L residuals:

$$\varepsilon^{L}_{n,t-1} = (\varepsilon_{n,t-L}, \ldots, \varepsilon_{n,t-1}) \tag{4}$$

The **cumulative residual vector** x is then defined as:

$$x := \text{Int}(\varepsilon^{L}_{n,t-1}) = \left(\varepsilon_{n,t-L},\ \varepsilon_{n,t-L} + \varepsilon_{n,t-L+1},\ \ldots,\ \sum_{l=1}^{L} \varepsilon_{n,t-L-1+l} \right) \tag{5}$$

Where Int is the integration operator. The cumulative residual vector x captures the cumulative effect of the residuals over the past L days, providing a measure of the overall trend in the residuals.

For the purpose of this project, we cointegrate for the length of a year. A longer length of time would allow for a more comprehesive cointegration, however, it sacrifices the ability to capture short-term trends. A shorter length of time would allow for a more responsive model, however, it sacrifices the ability to capture long-term trends. The choice of the length of the cointegration period is a trade-off between these two extremes.

Additionally, this report arbitrarily uses the cumulative residual window size of $L = 30$ days. The choice of L may affect the performance of the model, however, exploring different values of L is out of the scope of this project.

3.1.3 Portfolio positions

The output for all models is a vector of N values, where N is the number of assets. The output is a soft normalized vector of the portfolio positions, where each value represents the weight of the corresponding asset in the portfolio. The weights are normalized such that they sum to 1, and they are constrained to be between -1 and 1. This means that the model can take long or short positions in each asset, but the total position in each asset is limited to 100% of the portfolio value. The output takes the form of a vector w:

$$w = (w_1, w_2, \ldots, w_N) \tag{6}$$

Where w_i is the weight of asset i in the portfolio. The weights are normalized such that:

$$\sum_{i=1}^{N} w_i = 1 \tag{7}$$

3.1.4 Model performance evaluation

Returns

There is a one return associated with every 30-day cumulative residual window. Each return is calculated as the matrix product of the the model outputs and the returns of the assets, one day after the end corresponding cumulative residual window. The return is calculated as:

$$R_t = w^\top R_{t+1} - \sum_{i=1}^{N} w_i \tag{8}$$

Where R_{t+1} is the vector of log returns for all assets at time $t + 1$, and w is the vector of portfolio weights at time t.

Sharpe ratio

The Sharpe ratio is a measure of risk-adjusted return, calculated as the ratio of the mean excess return and the standard deviation of the excess return. It is used to evaluate the performance of a trading strategy or investment portfolio. A higher Sharpe ratio indicates better risk-adjusted performance. Therefore, we use the negative of the Sharpe ratio as the loss function for our model.

Consider the set of returns $R = (R_1, R_2, \ldots, R_T)$, where T is the number of returns. The Sharpe ratio is calculated as:

$$\text{Sharpe ratio} = \frac{\mathbb{E}[R]}{\text{Std}[R]} \tag{9}$$

Where $\mathbb{E}[R]$ is the mean of the returns, and $\text{Std}[R]$ is the standard deviation of the returns. The negative of this Sharpe ratio is then used as the loss function for the model.

4 Data

We have two different datasets. One containing the electricity prices for 31 different countries in Europe, and a second dataset containing weather data for these countries, in the same period.

4.1 European electricity prices

The electricity price data used in this project was taken from EMBER's *European Wholesale Electricity Price Data* (Ember, 2025). The data used in this report consists of daily electricity prices for 31 European countries from 2015 to 2025. The data was downloaded in CSV format and preprocessed to remove any missing values. The information contained in the dataset is the country, data, and price given in EUR per MWh.

4.2 European weather data

We used the dataset *ERA5 hourly time-series data on single levels from 1940 to present*. (Hersbach et al., 2025). This data originally contained hourly weather data, such as wind speed, precipitation and temperature above 2m above sea level. For us to use this dataset, we have aggregated the data to daily instead of hourly. For each of the data columns, we have taken the average during the day, aswell as the maximum and minimum values.

Now Filip will talk about the model and training workflow

5 Models

Model Inputs

We gather the log of the cumulative returns of the entire set of assets, using Equation 2. We then apply cointegration for an entire year (365 days) to the log returns of the assets, to obtain the time-series residuals for each asset for the entire year, using Equation 3. We then take the residuals of the last 30 days, starting from the first day of the cointegration period, and take cumulative sum of the residuals over a rolling window of 30 days, using Equation 5. This gives us a time-series of cumulative residuals for each asset, which we use as the input to the model. The input takes the form:

$$x_i = \begin{pmatrix} \varepsilon_{i,t-L} & \varepsilon_{i,t-L-1} & \dots & \varepsilon_{i,t} \end{pmatrix} \tag{10}$$

where x_i is the input for asset i, $\varepsilon_{i,t-L}$ is the residual of asset i at time $t-L$, t is the current time within the cointegration period, and L is the length of the rolling window.
The input is a concatenation of the cumulative residuals for all assets, and takes the form of a matrix x of size $N \times L$, where N is the number of assets and L is the length of the rolling window. The input is a matrix of size $N \times L$, and takes the form:

$$x = \begin{pmatrix} x_1 & x_2 & \dots & x_N \end{pmatrix} \tag{11}$$

The 30-day cumulative residuals is as a rolling window, sliding one day at a time across the the cointegration period. Therefore, the total number of inputs (data-points) for a single cointegration period is $T - L$, where T is the total number of days in the cointegration period.

Architecture:

- CNN+(Transforer)+FFN
- 2 Layers of CNN 1D with RELU Activation
- Transformer component in the (can be turned off)
- FFN which ouput the learned underlying feature of the data into a vector the same size of the number of countries in the dataset, and soft normalised for constructing the portfolio.

Data	Architecture
Price Data Only	CNN + FNN
Price Data + Weather Data	CNN + FNN
Price Data Only	CNN + Transformer + FNN
Price Data + Weather Data	CNN + Transformer + FNN

Table 1: Caption

Model Outputs

For every input, the model outputs a vector of portfolio weights *w* for each asset, using Equation 6. The output is a soft normalized vector of the portfolio positions, where each value represents the weight of the corresponding asset in the portfolio. The weights are normalized such that they sum to 1, and they are constrained to be between -1 and 1. This means that the model can take long or short positions in each asset, but the total position in each asset is limited to 100% of the portfolio value.

These weights are then multiplied by the price of the assets one day after the end of the associated input's cumulative residual window. This gives the model's next-day return for that input.

The set of inputs creates a set of next-day returns, which are then used to calculate the Sharpe ratio of the model's performance for that cointegration period. The Sharpe ratio is calculated using Equation 9, and the negative of the Sharpe ratio is used as the loss function for the model.

6 Results

7 Discussion

References

Avellaneda, M. and Lee, J.-H. (2010). Statistical arbitrage in the us equities market. *Quantitative Finance*, 10(7):761–782.

Ember (2025). European wholesale electricity price data. `https://ember-energy.org/data/european-wholesale-electricity-price-data/`. Dataset. Retrieved May 21, 2025, from Ember.

Guijarro-Ordonez, J., Pelger, M., and Zanotti, G. (2021). Deep learning statistical arbitrage. *arXiv preprint arXiv:2106.04028*.

Hersbach, H., Cobb, A., Kaandorp, M., Poli, P., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., and Thépaut, J.-N. (2025). Era5 hourly time-series data on single levels from 1940 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS). Dataset. Retrieved from `https://cds.climate.copernicus.eu/datasets/reanalysis-era5-single-levels-timeseries`.