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

June 1, 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 Long-Short Portfolio

A long-short portfolio is a trading strategy that involves taking long positions in undervalued assets and short positions in overvalued assets. The goal of this strategy is to profit from relative movement of an asset relative to others within the portfolio. Going **long** on an asset means buying the asset with the expectation that its price will increase, allowing the trader to sell it later at a higher price. Conversely, going **short** on an asset means borrowing the asset and selling it with the expectation that its price will decrease, allowing the trader to buy it back later at a lower price and return it to the lender.

The long-short portfolio is constructed such that the total value of the long positions is equal to the total value of the short positions, allowing the trader to profit from the movements of assets relative to the other assets within the portfolio, rather than from the overall market movement.

3.2 Statistical Arbitrage: Cointegration

Cointegration is a statistical property of a set time series variables that indicates a long-term correlation between them. The basis of cointegration is that historically correlated time series can deviate from their correlation in the short term, but will eventually revert to being correlated.

Consider N assets with log cumulative returns $R_{1,t}, R_{2,t}, \ldots, R_{N,t}$. We can investigate wether the time series of the first asset $R_{1,t}$ is correlated with the time series of all the other assets $R_{2,t}, \ldots, R_{N,t}$ by looking at the residuals of the first asset after regressing it on the other assets.

$$\varepsilon_{1,t} = R_{1,t} - \sum_{i=2}^{N} \beta_{1,i} R_{i,t} \tag{1}$$

The residual $\varepsilon_{1,t}$ is the difference between the log cumulative return of the first asset and a linear combination of the log cumulative returns of the other assets, at time t. If this residual centers around a constant mean (typically at zero), it indicates that the first asset is **cointegrated** with the other assets.
We can extend this idea for all N assets, by looking at the residuals of each asset after regressing it on all the other assets. If the residuals of all the assets center around a constant mean, it indicates that all the assets are cointegrated with each other.

$$\varepsilon_{n,t} = R_{n,t} - \sum_{j=1, j\neq n}^{N} \beta_{n,j} R_{j,t} \tag{2}$$

Where $\varepsilon_{n,t}$ is the residual of asset n at time t, and $\beta_{n,j}$ is the coefficient of asset j in the regression of asset n on all the other assets.

The signficance of cointegration is that it allows us to determine whether an asset is undervalued or overvalued relative to the other assets. If the residuals of an asset deviate significantly from zero, it indicates that the asset is either undervalued or overvalued relative to the other assets. This allows traders to take advantage of these deviations by taking long or short positions in the assets.

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

For example, if $e_{i,t} > 0$, it indicates that asset i is overvalued. Traders would short the asset. Conversely, if $e_{i,t} < 0$, it indicates that asset i is undervalued, and traders would go long on the asset.

In Equation 2, it is up to the user to choose the length and time period of the cointegration. For the rest of the report, we refer to a particular period of cointegration as a **cointegration window**. The length of the cointegration window is the number of days used to calculate the residuals, and the time period is the start and end date of the cointegration window. When picking the length of the cointegration window is a trade-off between capturing long-term trends and short-term fluctiations. A longer cointegration windows allows for a more comprehsive cointegration, however, it sacrifices the ability to capture short-term trends. A shorter cointegration window allows for a more responsive model, however, it sacrifices the ability to capture long-term trends. The choice of the length of the cointegration window is a trade-off between these two extremes - for this report, we will use a cointegration window of 365 days. However, this choice is largely arbitrary, and the performance of the model may vary depending on the length of the cointegration window.

In the context of European electricity prices, we expect the prices in Europe to be highly correlated, 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.

3.2.1 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:

$$R_{i,t} = \ln\left(\frac{P_{i,t}}{P_{i,0}}\right) \tag{3}$$

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.
We can simply this by defining the vector $\mathbf{R}_t$ as the vector of log cumulative returns for all assets at time t, and the vector $\mathbf{P}_t$ as the vector of prices for all assets at time t. The log cumulative return vector is then defined as:

$$\mathbf{R}_t = \ln\left(\frac{\mathbf{P}_t}{\mathbf{P}_0}\right) \tag{4}$$

3.2.2 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 window from day t_i to t_f. For asset n, define $\varepsilon_{n,t}^{t_i,t_f}$ as the vector of the past residuals from day t_i to day t_f:

$$\varepsilon_{n,t}^{t_i,t_f} = \left(\varepsilon_{n,t_i},\ \varepsilon_{n,t_i+1},\ \ldots,\ \varepsilon_{n,t_f}\right) \tag{5}$$

The **cumulative residual vector $\mathbf{x}_n^{t_i,t_f}$** is then defined as:

$$\mathbf{x}_{t_i,t_f}^{(n)} := \mathrm{Int}(\varepsilon_{n,t}^{t_i,t_f}) = \left(\varepsilon_{n,t_i},\ \varepsilon_{n,t_i} + \varepsilon_{n,t_i+1},\ \varepsilon_{n,t_i} + \varepsilon_{n,t_i+1} + \varepsilon_{n,t_i+2},\ \ldots,\ \sum_{t=t_i}^{t_f} \varepsilon_{n,t}\right) \tag{6}$$

Where Int is the integration operator, and $\varepsilon_{n,t}$ is the residual of asset n at time t. For the rest of the report, we will refer to the period from t_i to t_f as the **cumulative residual window**.

This report arbitrarily uses the cumulative residual window size of $L = t_f - t_i = 30$ days. We expect that the choice of L would affect the performance of the model, however, exploring different values of L is out of the scope of this project.

3.2.3 Portfolio positions

For this report, we define a portfolio position as a vector w of weights for each asset in the portfolio. The weights represent the long or short position of each asset in the portfolio, and are soft normalized such that they sum to 1. The portfolio position is defined as:

$$\mathbf{w} = (w_1, w_2, \ldots, w_N) \tag{7}$$

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

$$\sum_{i=1}^{N} w_i = 1 \tag{8}$$

3.2.4 Model performance evaluation

Returns

Given the portfolio position $\mathbf{w}_t$ at time t, the return of the portfolio at time t is defined as the matrix product of the portfolio weights and the vector of log returns of all assets at time $t + 1$. The return is defined as:

$$\mathbf{r}_t = \mathbf{w}_t^\top \mathbf{R}_{t+1} \tag{9}$$

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 (SR) 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 $\mathbf{r} = (r_1, r_2, \ldots, r_T)$, where T is the number of returns. The Sharpe ratio is calculated as:

$$\mathrm{SR} = \frac{\mathbb{E}[\mathbf{r}]}{\mathrm{Std}[\mathbf{r}]} \tag{10}$$

Where $\mathbb{E}[\mathbf{r}]$ is the mean of the returns, and $\mathrm{Std}[\mathbf{r}]$ 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

Input Preperation

We prepare the input data for the model using the following steps:

Create cointegration windows:

- The full historical dataset (2015–2024) is segmented into **cointegration windows** of 365 days, rolling forward every 182 days.

Within each cointegration window:

1. **Log Cumulative Returns**:
    - For each asset n, calculate the log cumulative returns as defined in Equation (3).
    - Shapes:
        - Input price matrix: $P \in \mathbb{R}^{N \times T}$
        - Output return matrix: $R \in \mathbb{R}^{N \times T}$
2. **Cointegration Residuals**:
    - Compute residuals by regressing each asset's returns on all others, as defined in Equation (2).
    - Output shape: $\varepsilon \in \mathbb{R}^{N \times T}$
3. **Cumulative Residuals (Window Length $L = 30$)**:
    - Within each cointegration window, we slide a fixed-length window of 30 days across time **one day at a time** to construct the model input.
    - For each starting index t_i, we compute a sequence of cumulative sums over the next 30 days, as defined in Equation (6):
    - One input to the model:

    $$\mathbf{x} \in \mathbb{R}^{N \times L} \quad \text{(cumulative residuals across } N \text{ assets and } L = 30 \text{ days)}$$

    - If weather data is used, it is concatenated channel-wise:

    $$\mathbf{x}_{\text{aug}} \in \mathbb{R}^{(N+W) \times L}$$

    where W is the number of weather features (e.g., temperature, wind, precipitation).
4. **Next-Day Return Target**:
    - For each residual window ending at day t, define the next-day return target as:

    $$\mathbf{y}_t = \mathbf{r}_t = \mathbf{R}_{t+1} \in \mathbb{R}^N$$

    where $\mathbf{R}_{t+1}$ is defined by Equation (4).

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 7. 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 10, and the negative of the Sharpe ratio is used as the loss function for the model.

6 Results

7 Discussion

References

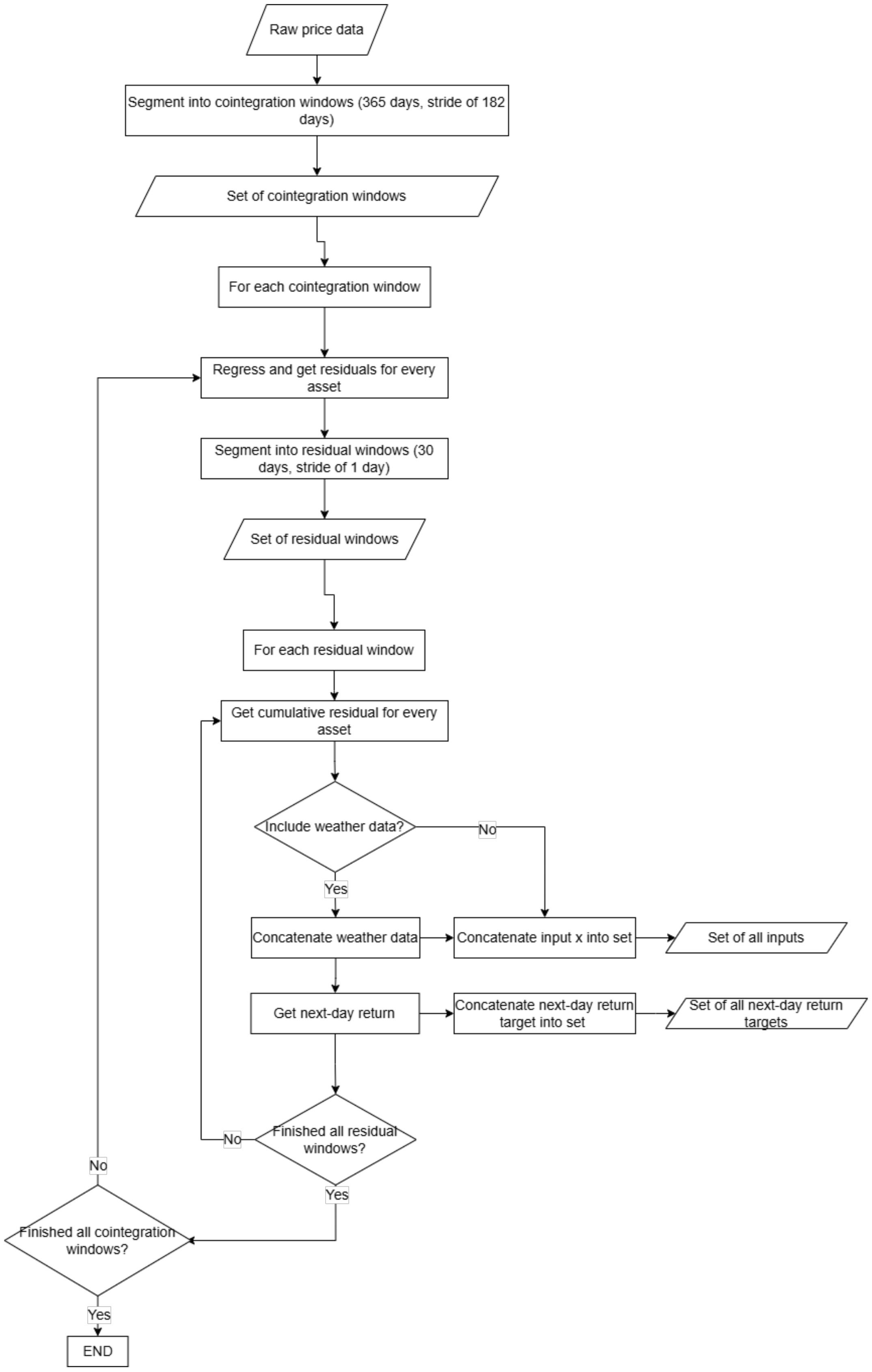
Avellaneda, M. and Lee, J.-H. (2010). Statistical arbitrage in the us equities market. *Quantitative Finance*, 10(7):761–782.

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`.

8 Appendix



Raw price data
Segment into cointegration windows (365 days, stride of 182 days)
Set of cointegration windows
For each cointegration window
Regress and get residuals for every asset
Segment into residual windows (30 days, stride of 1 day)
Set of residual windows
For each residual window
Get cumulative residual for every asset
Include weather data?
No
Yes
Concatenate weather data
Concatenate input x into set
Set of all inputs
Get next-day return
Concatenate next-day return target into set
Set of all next-day return targets
Finished all residual windows?
No
Yes
No
Finished all cointegration windows?
Yes
END

Figure 1: Input Preparation Pipeline for Deep Learning Model