POLITECNICO
MILANO 1863

PORTFOLIO REPLICA STRATEGY

FINTECH FINAL PROJECT

Our Team
Group 2

Matteo Callini

Alessandro Howe

Cecilia Gaspari

Carlotta Carzaniga

Michela Galletti

POLITECNICO
MILANO 1863

About The Project

We Bring Ideas To Life

We aim to reverse-engineer a secret investment portfolio - a financial black-box - that publicly reports its returns but hides its internal composition.

- Unknowns: Asset selection and weightings.
- Knowns: Historical returns and universe of tradable instruments such as stocks, bonds and ETFs.

Goal:

Identify a combination of liquid assets that replicates the black-box's return pattern by analyzing its performance over time.

Explore the project

POLITECNICO
MILANO 1863

Our Project
Key Steps

Step 01

Portfolio Replication
Replicate the target portfolio's returns and infer its hidden structure.

Step 02

Model Selection
Identify the best-performing model based on financial metrics.

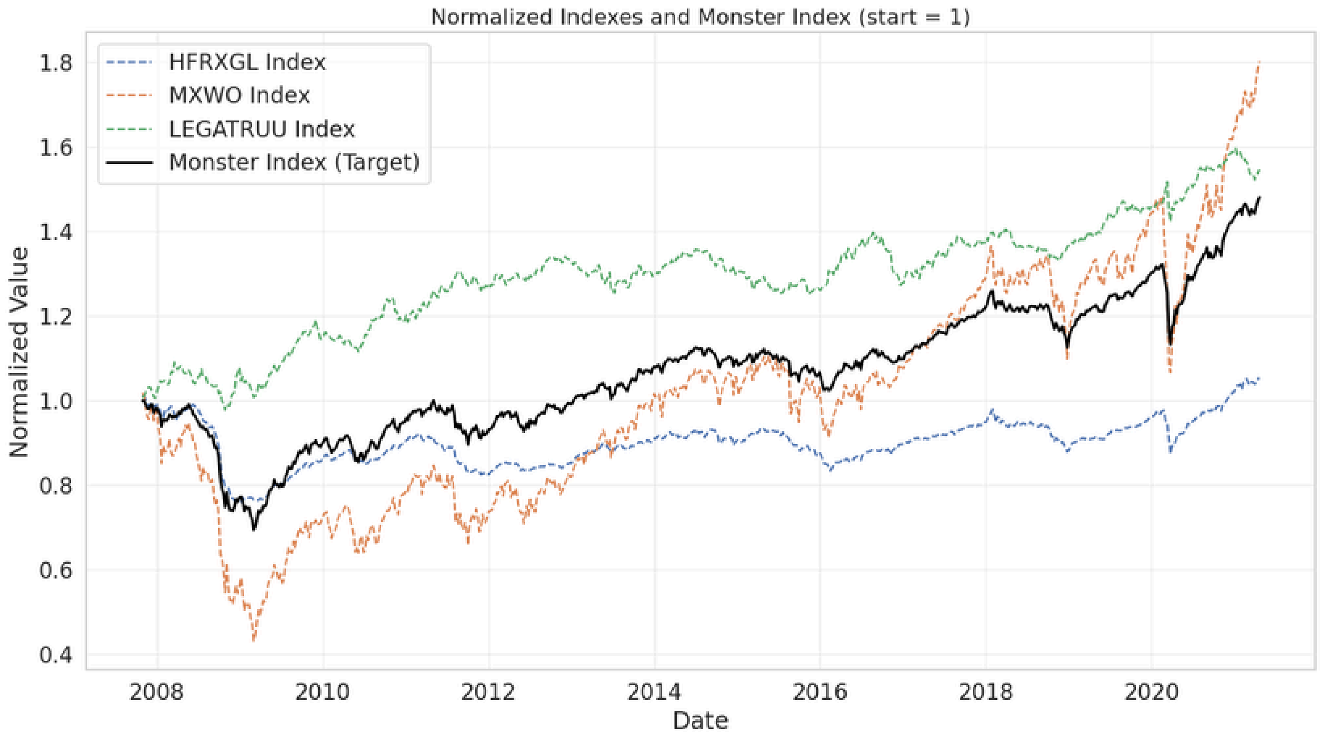
Step 03

Strategy Design
Build an investable strategy that emulates the target's performance.

POLITECNICO
MILANO 1863

Data Exploration

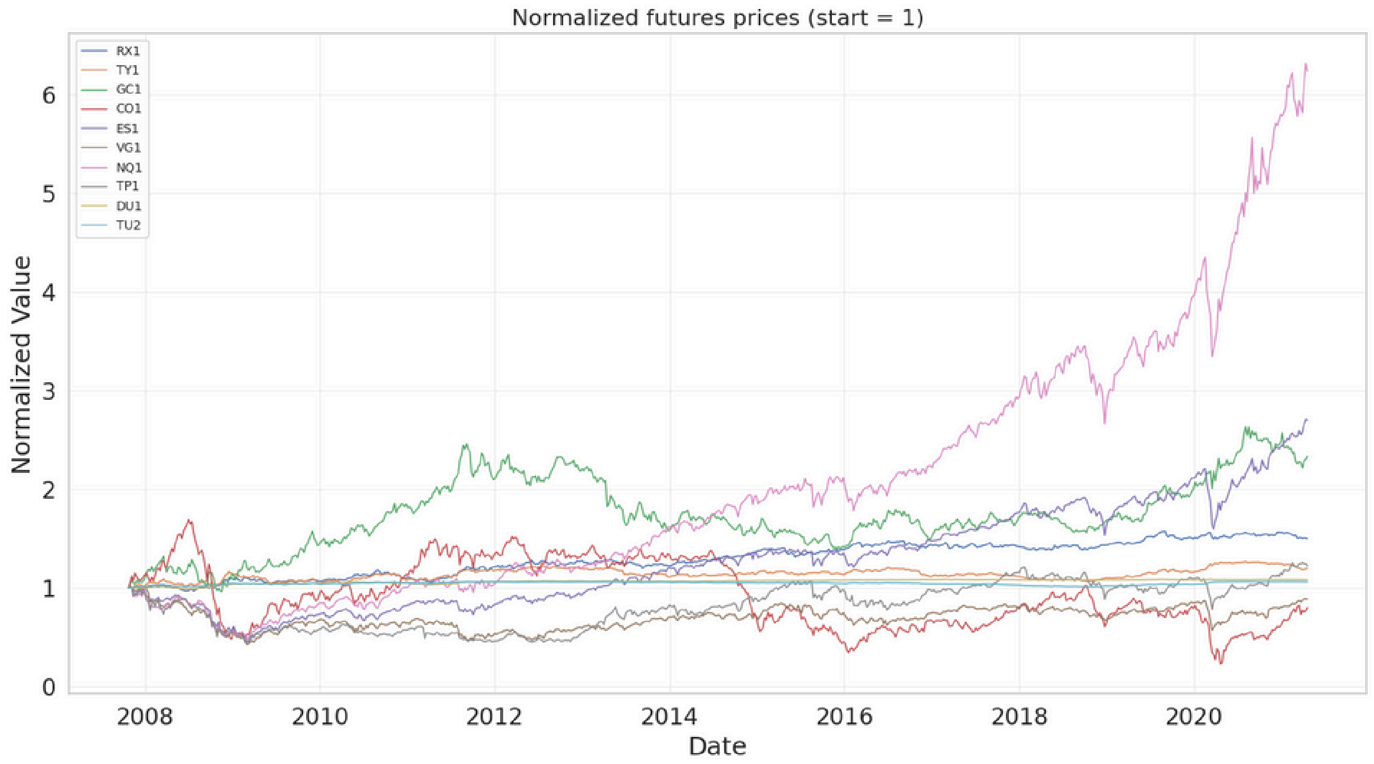
About The Dataset



We created a custom Monster index by combining multiple underlying indices, for which we tried to replicate its performance.
The index is composed by:

- HFRXGL Index: 1/4,
- MXWO Index: 1/2,
- LEGATRUU Index: 1/4

We analyzed the provided futures dataset and searched for recurring patterns, but didn't find any. The LL1 future showed no activity over time, so we removed it. All futures were normalized for consistency. We also performed an autocorrelation analysis on the target portfolio to check for seasonality or cycles, but none were identified.



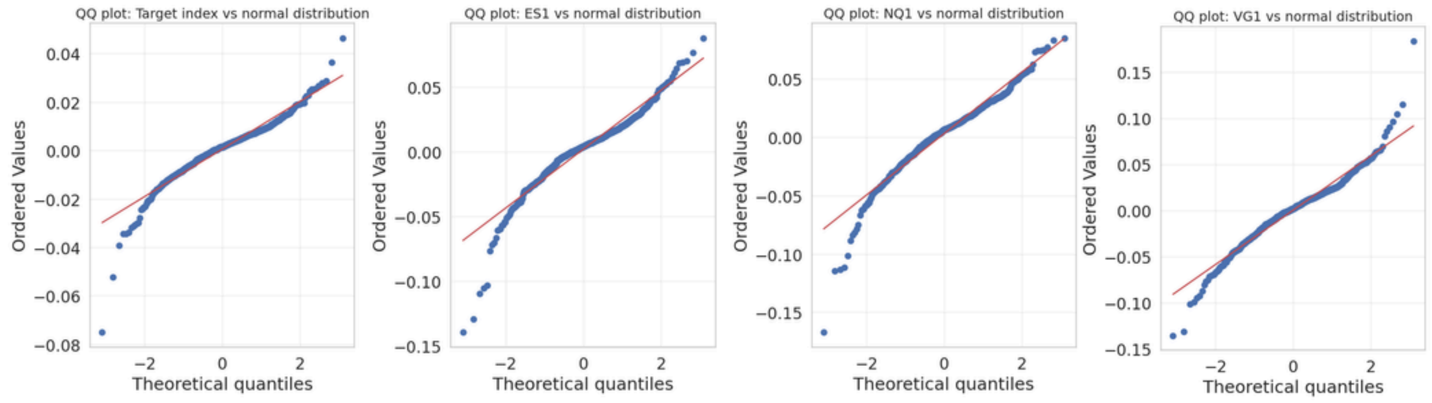
POLITECNICO
MILANO 1863

Problem Setup

About The Dataset

Gaussianity

Both the Target Index and the other indeces are proved to be not normal with the QQ-plot and the Shapiro-Wilk test. So we will not be using gaussianity hypotesis throughout our implementations.



Value-at-Risk (VaR) Calculation

We controlled downside risk in the index replication by imposing a 1-month, 1% VaR constraint.
Our approach was to use the **Exponentially Weighted Historical VaR** to capture recent market conditions more accurately by assigning higher weight to recent observations.

Transaction Costs

Transaction costs are explicitly modeled to better reflect real-world portfolio performance. Specifically, they are modeled with a 0.05% rate (5 bps) applied to the traded volume, calculated as the L1 norm between current and previous weights.
This way, the models implicitly favors more stable and cost-efficient strategies, making the replication more realistic from an execution standpoint.

Model Exploration

With the goal of replicating the performance of a complex, non-transparent index, we moved to model development with a clear principle: balancing **interpretability**, **adaptability**, and **predictive power.**

To do so, we selected three complementary models and finally a reasoned ensemble approach:

- Elastic Net Regression
- Kalman Filter
- Recurrent Neural Network
- Ensemble model

POLITECNICO
MILANO 1863

About The Methods

Elastic Net Regression:
Chosen for its simplicity and transparency, Elastic Net allows us to model stable linear relationships while automatically performing feature selection. This helps avoid overfitting and keeps the model interpretable .

Kalman Filter:
This model introduces adaptive learning, updating its estimates as new information arrives. It’s well-suited for time-varying relationships and market regimes. We used it to capture gradual shifts in portfolio structure over time, especially useful in dynamic macro environments.

Recurrent Neural Network (RNN):
While less interpretable, RNNs offer strong predictive capabilities for sequential data. They are ideal for capturing non-linear temporal dependencies and recurring patterns that may not be visible to linear models.

Finally, we integrated the first two models in an **Ensemble Framework**, leveraging their individual strengths while mitigating their weaknesses. This approach offers a more robust and flexible replication, as it avoids dependence on a single modeling strategy and adapts better to different market conditions.

ELASTIC NET

$$\min_{\beta}\left\{\|y - X\beta\|^2 + \lambda\left(\alpha\sum_j|\beta_j| + (1-\alpha)\sum_j\beta_j^2\right)\right\}$$

OPTUNA

We exploited optuna to automatically tune:

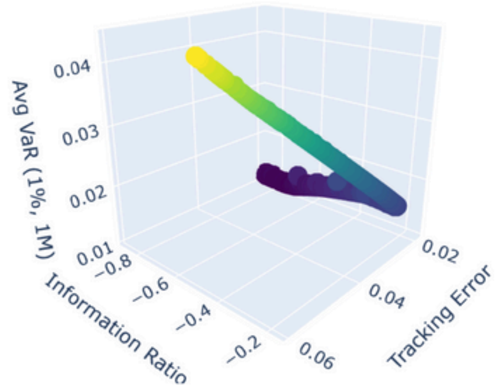
- l1_ratio: to balance L1 and L2 regularization ∈ [0, 1]
- alpha: on logarithmic scale for regularization strenght ∈ [1e-5, 1e-1]

The rolling_window was set at 1 year time

We used a multi-objective study:

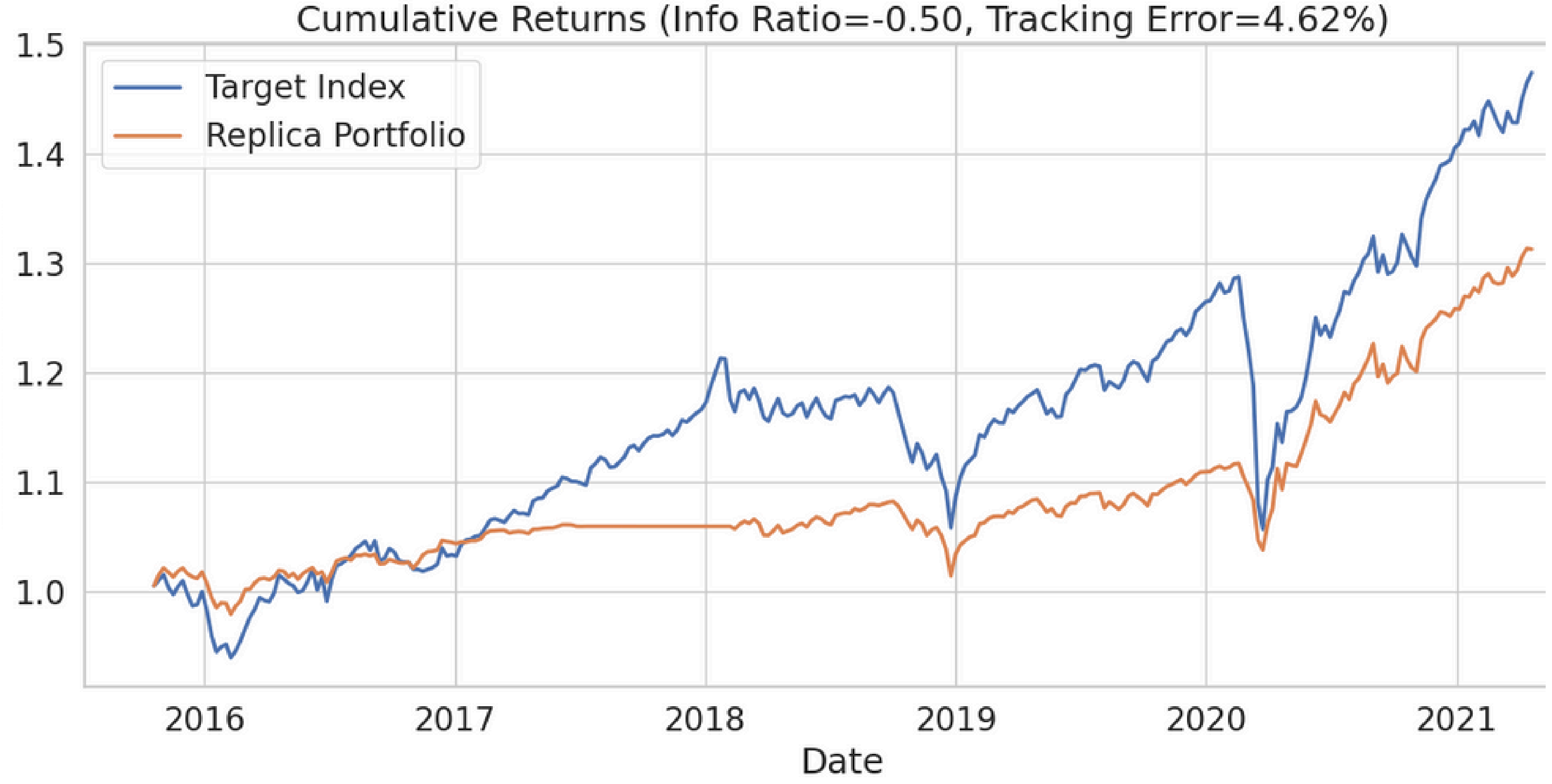
- minimizing the tracking error
- maximizing the information ratio
- minimizing the average VaR

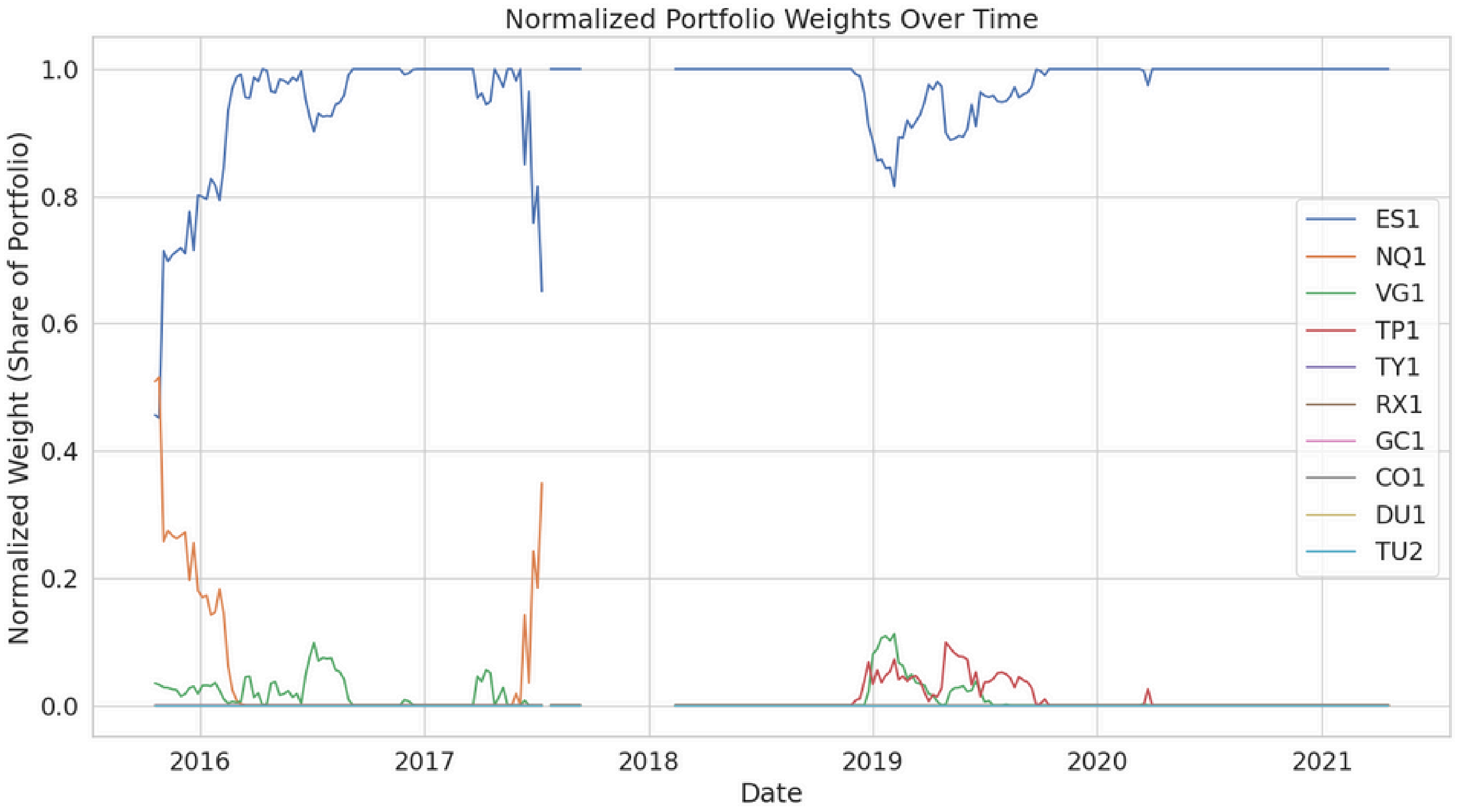
Pareto Frontier (Backtest-Based Optimization)



We took the best Pareto trial maximizing the information ratio

The **l1_ratio** selected by various optuna trials tends to one and therefore seems to prevail the ridge part.

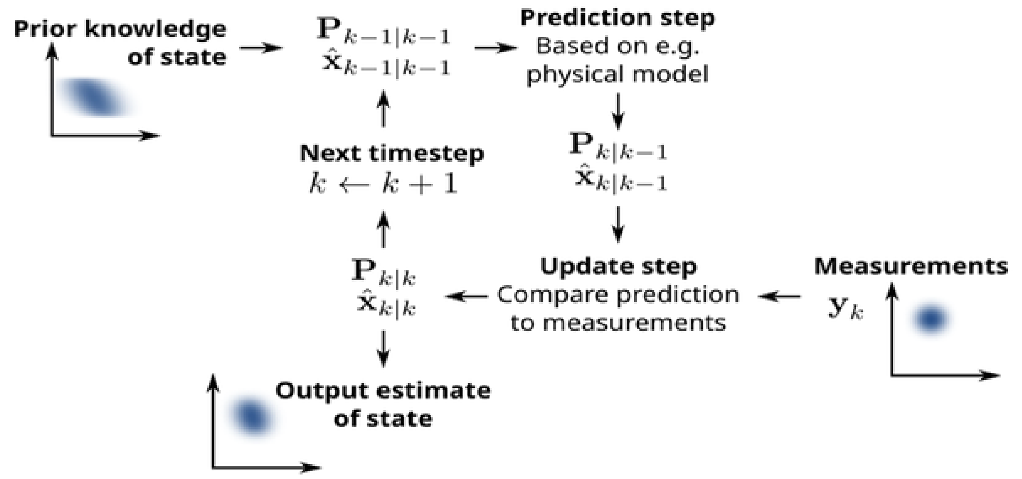




POLITECNICO
MILANO 1863

KALMAN FILTER

The Kalman Filter is a recursive algorithm ideal for estimating time-varying portfolio weights from noisy observations — perfect for financial time series.



We replicated the return of a target index by dynamically estimating portfolio weights as a linear combination of asset returns:

$$r_t^{\text{target}} = X_t \cdot w_t + \varepsilon_t$$

/here:

- X_t: asset returns at time t
- w_t: **latent (hidden), time-varying weights**
- ε_t: observation noise

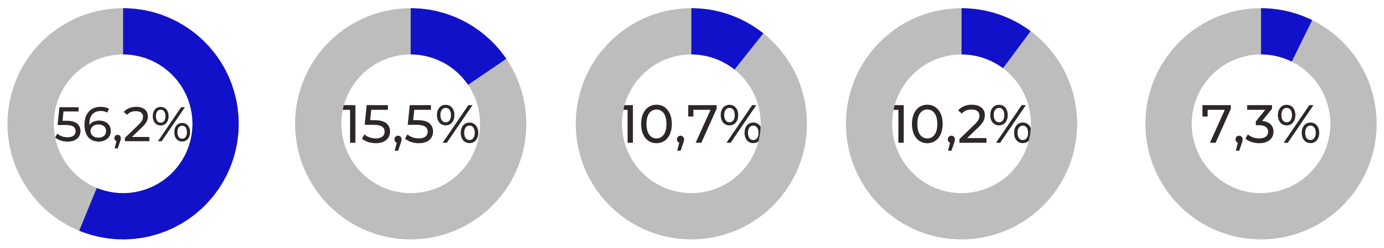
Why Kalman Over Static Methods (e.g., Ridge)?

Unlike Ridge regression, which assumes fixed weights, the Kalman Filter treats portfolio **weights** as **evolving over time**:

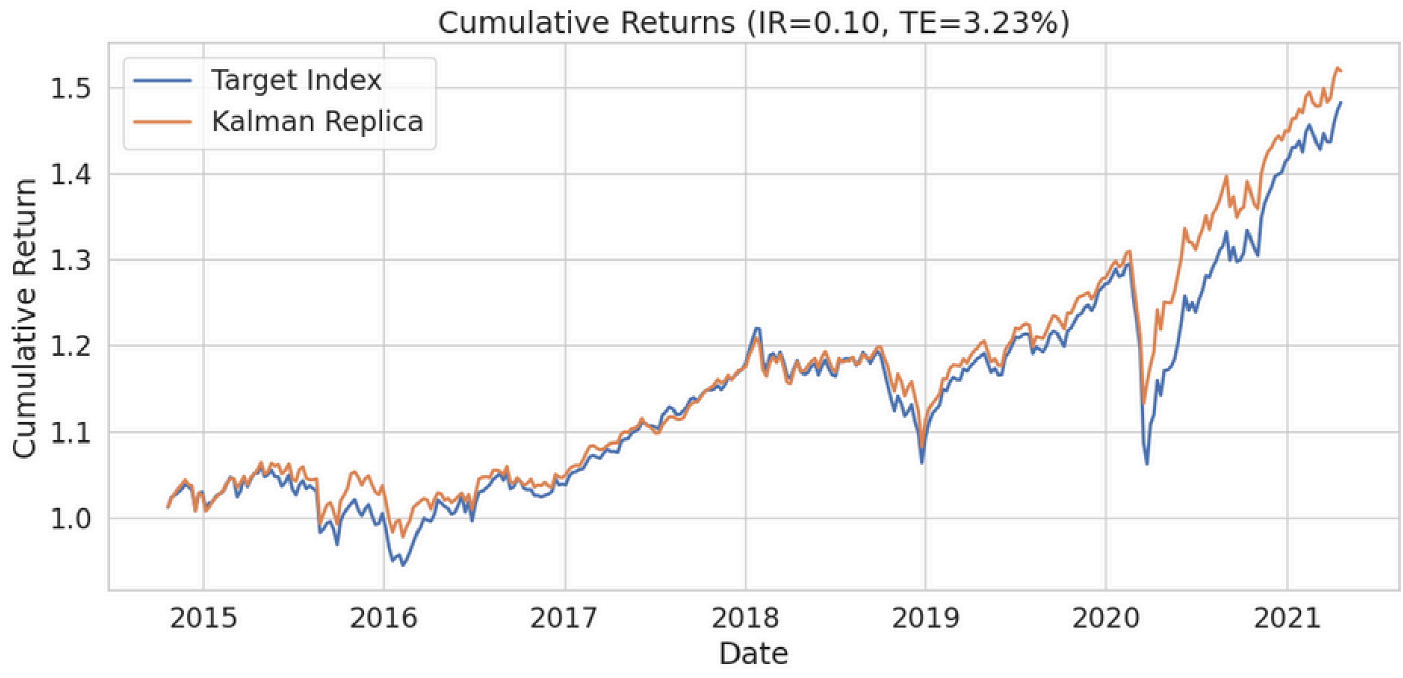
- Adapts to non-stationary relationships
- Smoothly incorporates new data
- Robust to noise and structural shifts

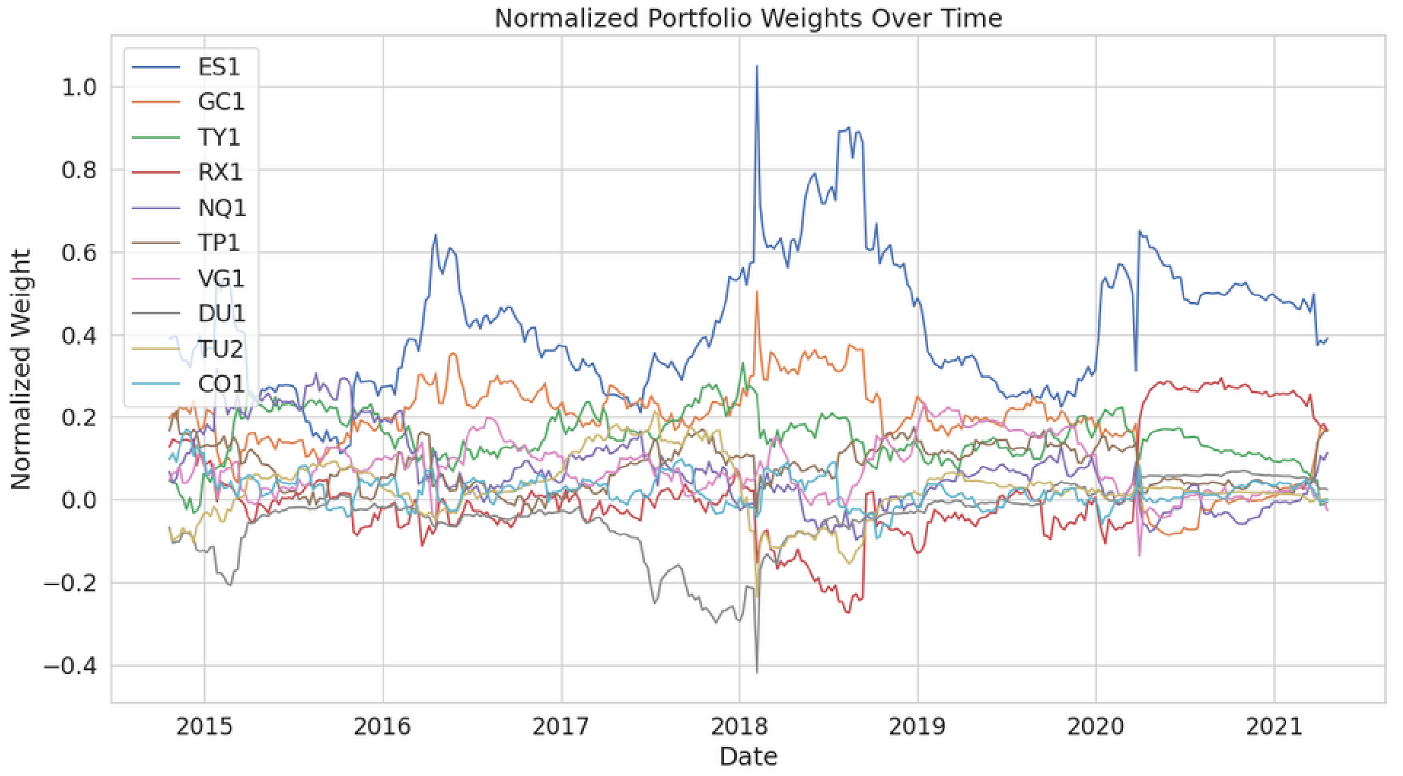
$$w_t = w_{t-1} + \omega_t \quad \text{with } \omega_t \sim \mathcal{N}(0, Q)$$

Territorial exposure based on Kalman Filter weights:



USA Global Germany Europe Japan



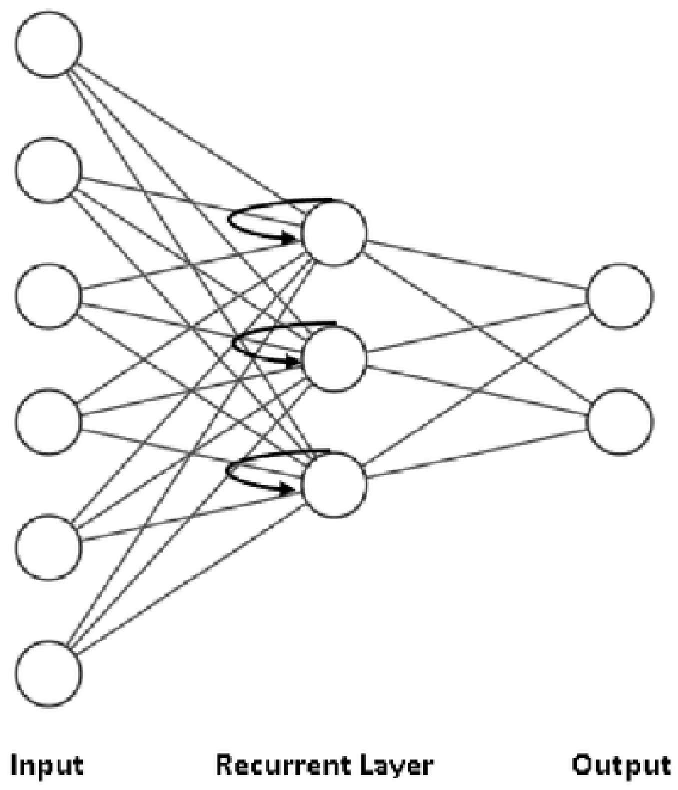


POLITECNICO
MILANO 1863

RECURRENT NEURAL NETWORK

- The core model is made by a **SimpleRNN** layer trained on historical asset returns using a sliding window approach.

TensorFlow



- A **sliding window** approach is used to generate rolling time-series sequences for training the model. Each forecast targets the next time step, and the model is retrained at every iteration, mimicking a rolling horizon strategy.
- This structure enables the model to adapt continuously to new market data, capturing evolving patterns and avoiding overfitting to historical segments.
- The architecture also includes **Dropout** and **Layer Normalization** to improve generalization and stability during training.

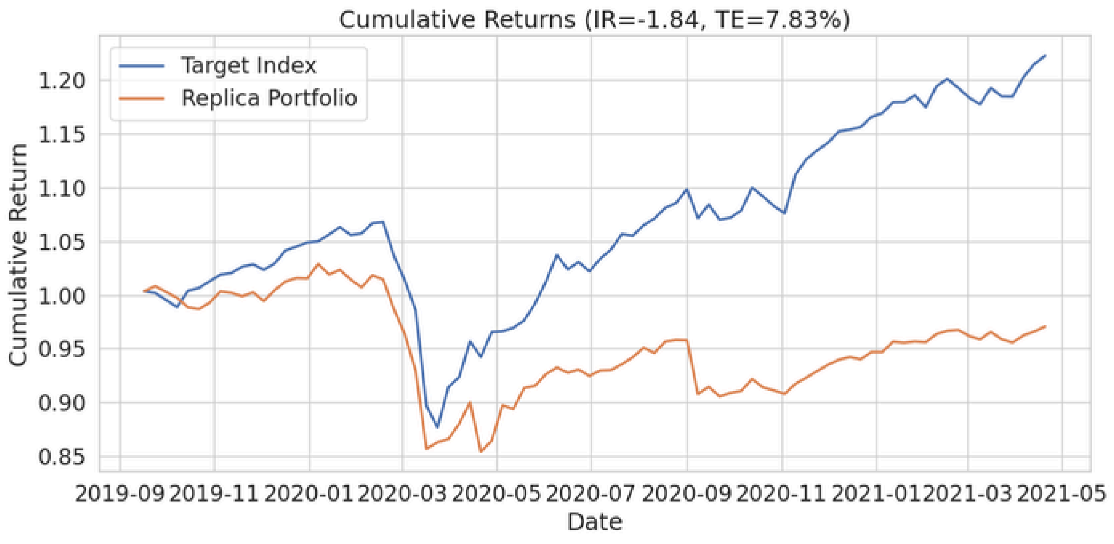
For the **hyperparameters** we tried using Optuna algorithm but it took too much time. So we used:

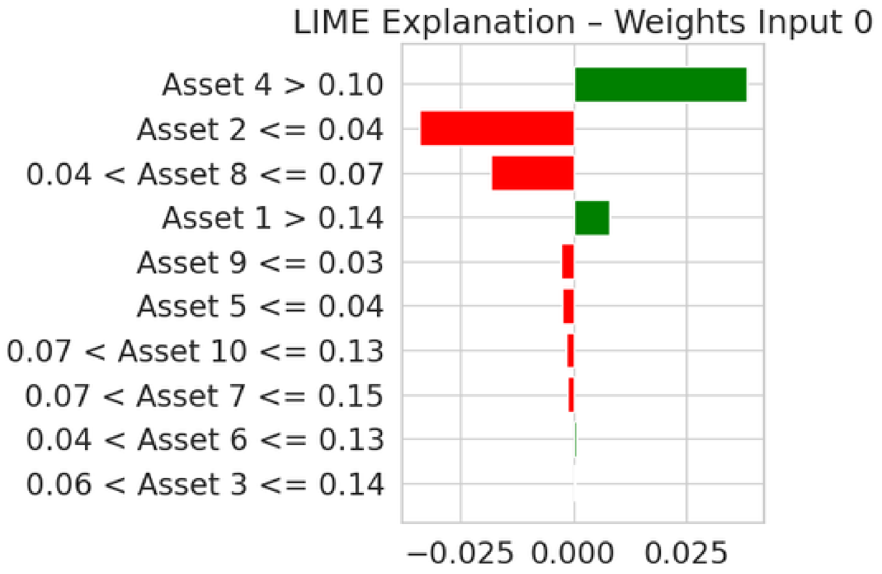
- neurons = 10
- batch_size = 32
- lr (learning rate) = 1e-3

The **multi-objective function** that we tried to use seeks to:

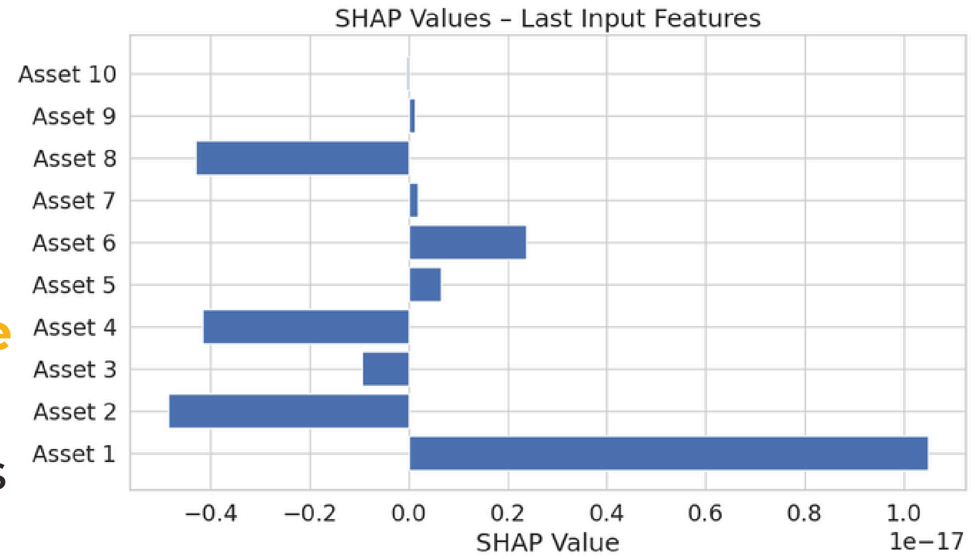
- Minimize tracking error
- Maximize information ratio
- Minimize average VaR

We considered only the last 84 weeks because it was too computationally heavy.





- **Asset 4** and **Asset 2** emerge as consistently relevant in both interpretations
- **LIME** plot provides a **more interpretable** picture of feature importance in this case for portfolio linearity



POLITECNICO MILANO 1863

ENSEMBLE MODEL

The Idea

We built an ensemble replicator, to try outperform the single models, by combining **three distinct models:** the Kalman Filter (state-space, adaptive), the Elastic Net, and the RNN. Each model provides weekly replica returns, which we aligned on a common time index for consistent aggregation.

The Weights

We calibrated the ensemble weights using the first 52 weeks of the test period. A **grid of all possible weight combinations** summing to 1 (in steps of 0.05) is generated. For each combination, we compute key metrics: annualized return, sharpe ratio, and tracking error against the target index.

Rank-Based Voting VS Traking Error

In order to avoid conflicting results with respect to different metrics, we initially thought about a clever rank-based voting technique. Each portfolio is **ranked** across all metrics, and the ranks are summed. The configuration with the lowest total score (i.e. the one with highest rank overall) is selected, balancing risk, return, and consistency.

However, we ultimately decided to focus solely on **tracking error** as the primary selection criterion. This choice was driven by the specific objective of our project where minimizing deviation from the benchmark is crucial.

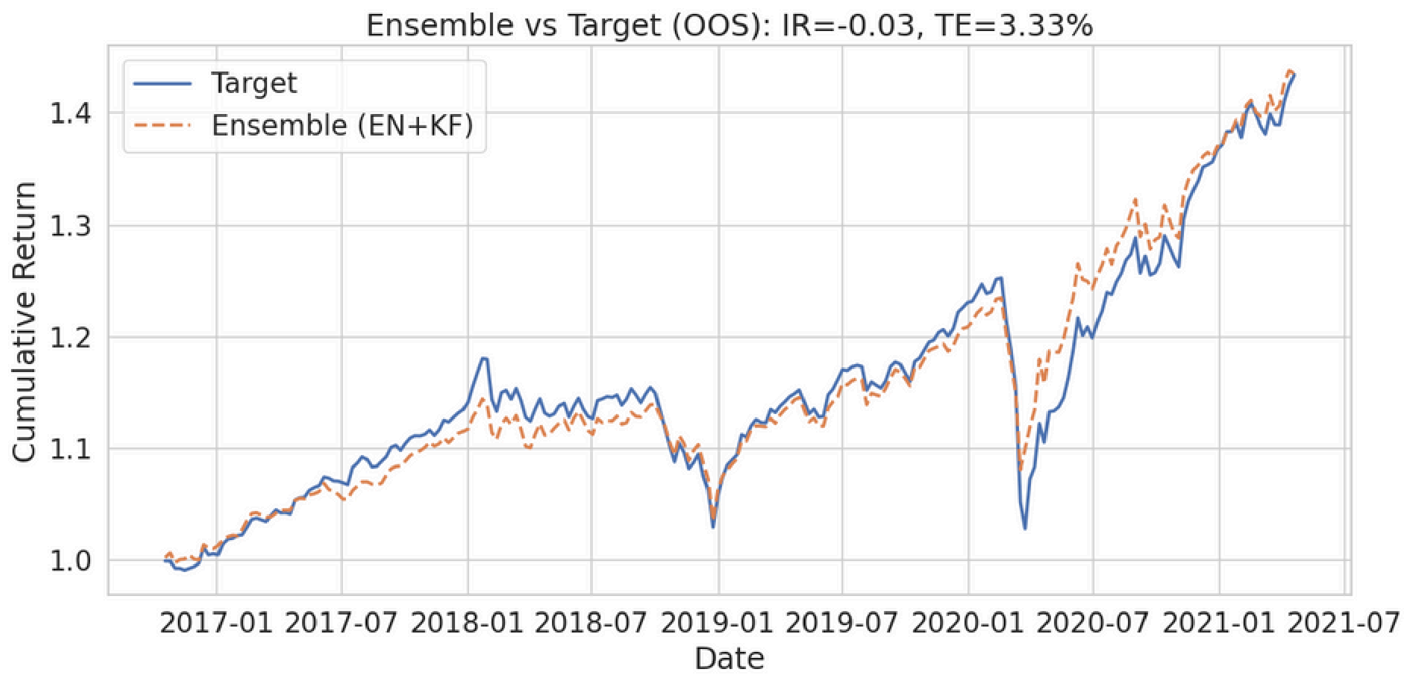
Out of Sample Result

The resulting ensemble replicator shows **strong out-of-sample performance**, with high sharpe and information ratios, low tracking error, and close correlation with the target. This confirms the benefit of model diversification and rank-based tuning.

ENSEMBLE MODEL

- For the Ensemble Model we combined the **Kalman Filter** and the **Elastic Net**, excluding the Recurrent Neural Network. The weights of the ensemble were calibrated by the model itself based on historical performance, resulting in an allocation of **0.85** to the Kalman Filter and **0.15** to the Elastic Net.

This combination was motivated by the complementary strengths of the two models:
-The Kalman Filter is highly adaptive and effective at capturing short-term dynamics, but it may introduce mild overfitting, particularly in volatile market regimes.
-The Elastic Net provides structural simplicity, regularization, and robustness, making it a valuable counterbalance to the Kalman Filter.



- By integrating these two components, the ensemble benefits from the **bias–variance tradeoff**: the Kalman Filter contributes low bias but higher variance, while the Elastic Net offers higher bias but lower variance. This tradeoff improves the model's generalization capability and stability.

POLITECNICO
MILANO 1863

MODELS PERFORMANCE

To evaluate the performance of a portfolio it's essential to consider a set of key metrics that capture both profitability and risk.

- **Annualized return** tells us how much the portfolio earns on average per year. It gives a sense of long-term performance, assuming the returns compound over time.
- **Annualized volatility** captures the degree of fluctuation in returns over time. It represents the risk or uncertainty associated with the portfolio's performance. The higher the volatility, the more returns can swing.
- **Sharpe Ratio** helps us understand how efficiently the portfolio takes on risk. It adjusts the portfolio's return by its volatility effectively showing how much excess return is earned for each unit of risk. A higher Sharpe Ratio generally indicates a better balance between risk and reward.
- **Information Ratio** compares the portfolio's performance not just in absolute terms but relative to a specific benchmark. It divides the excess return over the benchmark by the **Tracking Error**. This metric is particularly useful for active strategies aiming to consistently outperform a reference index.
- **Max drawdown** looks at the worst-case scenario: the largest peak-to-trough decline in the portfolio's value. It highlights vulnerability to severe losses and is critical for understanding downside risk.
- **Correlation** measures how closely the portfolio's movements align with a target – often a benchmark or another model. A high positive correlation suggests the portfolio behaves similarly to the reference, while a low or negative correlation indicates independent or opposite behavior. This insight is key when constructing diversified portfolios or evaluating ensemble models.
- **Average gross exposure** reflects the total capital allocated across all positions, long and short, without netting them. It gives insight into how much leverage or capital deployment is involved in the strategy, and how aggressively it's taking positions.
- **Average VaR** estimates the average potential loss in the worst-case scenarios beyond a chosen confidence level, it focuses on the expected loss when that threshold is breached.

POLITECNICO
MILANO 1863

MODELS PERFORMANCE

	ANNUALIZED RETURN	ANNUALIZED VOLATILITY	SHARPE RATIO	MAX DRAWDOWN	TRACKING ERROR	INFORMATION RATIO	CORRELATION	AVERAGE GROSS EXPOSURE	AVERAGE VaR (1%, 1M)
ELASTIC NET	5.03%	4.84%	1.04	7.08%	4.62%	-0.50	0.8771	0.2767	NaN
KALMAN FILTER	6.69%	7.68%	0.87	13.50%	3.23%	0.10	0.9209	0.7770	2.06%
NEURAL NETWORK	-1.22%	11.14%	-0.11	16.99%	7.83%	-1.84	0.7662	1.0000	6.86%
ENSEMBLE MODEL	8.22%	7.17%	1.15	13.50%	3.33%	-0.03	0.9211	N/A	N/A

POLITECNICO
MILANO 1863

Discussion: Key Points

1. Our recurrent neural network (RNN) model incorporates non-linearity and temporal depth, but the data structure appears largely linear and driven by low-order temporal dependencies. As a result, **simpler models** like the Kalman Filter and Elastic Net perform better, likely due to the relative smoothness of the target series and the limited benefit of capturing complex dynamics in this setting.

2. The RNN particularly struggles to capture the target starting after 2020 first quarter, likely due to the unique market conditions and **structural break** introduced by the COVID-19 crisis.

3. Our Elastic Net model performs well in the initial period, closely tracking the index, but struggles to capture the **upward trends** in returns over time. This is likely due to its strong regularization, which limits flexibility and may underfit longer-term market dynamics.

4. Kalman Filter shows strong predictive performance, particularly in capturing **short-term dynamics** of financial time series. However, its flexibility and adaptivity may lead to mild overfitting, especially in volatile regimes.

5. The Kalman Filter relies on **adaptive weights**, dynamically updating its estimates as new data becomes available which is a powerful but potentially noisy approach.

6. Initially, the ensemble model was optimized by minimizing multiple error metrics simultaneously; however, we now focus on **minimizing** the **tracking error** to better align the replica portfolio with the target index.

POLITECNICO
MILANO 1863

CONCLUSIONS

The Result

We selected the **ensemble model** as our final choice that combines the Kalman Filter and Elastic Net , aiming to balance adaptability with regularization. While it shows a slight increase in tracking error compared to the Kalman Filter alone, it delivers higher returns and greater stability in terms of volatility offering overall a **more robust** approach.
The ensemble gives more weight to the Kalman Filter due to its strong standalone performance, while the Elastic Net adds structural stability and improves predictions in the initial periods where it performs particularly well.
This model will be used to develop a **WebApp** for replicating the provided indices and two custom-designed Monster Indices.

The Theory

This approach aligns with the **bias-variance tradeoff principle**: combining a low-bias, high-variance model (Kalman) with a high-bias, low-variance model (Elastic Net) reduces overall error.
By blending complementary estimators, one that seems to overfit and one that underfits, the ensemble exhibits **strong generalization ability** and improved **robustness** to different market conditions.

The Metric

Focusing on **tracking error** minimization in the ensemble framework improves fidelity to the target index, enhancing practical portfolio replication.

POLITECNICO
MILANO 1863

CloneCapital:

Our WebApp

Our platform creates synthetic futures contracts that mirror indices with precision

Our Aim: Democratizing Index Investing

Traditional index investing faces barriers of high minimums, limited access, and excessive fees. CloneCapital breaks these constraints through innovative futures replication technology.

Target Audience

Young professionals looking to optimize their investment strategy

CLONE CAPITAL

POLITECNICO
MILANO 1863

CHECK OUT OUR WEBAPP

THANK YOU

POLITECNICO
MILANO 1863

Future Developements

Dynamic Optimization of Ensemble Weights

Currently, the ensemble model uses fixed weights for each component. Our next step would be to implement a dynamic weighting system that adapts in real-time based on market conditions and individual model performance. This will make the replication strategy more responsive and accurate.

Weight Smoothing for Enhanced RNN Target Fitting

Incorporate weight smoothing techniques within the recurrent neural network architecture to improve the model's fit to the target index. By enforcing smoother transitions in portfolio weights over time, the RNN can better capture persistent patterns and reduce noise in its allocation decisions.

Expansion of the Investable Universe

By incorporating additional asset classes—such as commodities, digital assets, or alternative instruments—we aim to enhance diversification and capture new sources of return. This will make the portfolio replicator more resilient and better equipped to handle different market regimes.

WebApp Evolution into a Full Product

We plan to develop our WebApp into a fully-featured platform for both retail and professional users. The vision is to offer customizable strategies, real-time analytics, and index replication simulations through a user-friendly and accessible interface.