GROUP 4
IRENE FERRARI
SILVIA FIECCHI
JACOPO STRINGARA
GIULIO VENTURINI

FINAL PROJECT - BUSINESS CASE 4

INVESTMENT REPLICA

FINTECH COURSE A.Y. 2022-23

OUR TEAM

Irene Ferrari

Mathematical Engineering - MMF

Silvia Fiecchi

Mathematical Engineering - MMF

Jacopo Stringara

Mathematical Engineering - MMF

Giulio Venturini

Mathematical Engineering - MST

CONTENT

GOALS AND OBJECTIVES

Objective n° 1

Replicating an investment strategy by means of different mathematical models

Objective n° 2

Determining which model, among the ones we used, is the best by **comparing** different financial indexes

Objective n° 3

Creating an accessible **app** that enables clients to replicate the investment strategy, using the desired bundle of available futures

DATA PREPROCESSING

TARGET

FEATURE SELECTION

APP DATA

NN DATA

MONSTER INDEX

We chose as **target** of our portfolio replica a weighted combination of HFRXGL, MXWO, and LEGATRUU. The reason of this, is that their price does not depend upon an underlying asset, since they're not financial derivatives.

LLL1 REMOVAL

By means of a stationarity analysis, we found out that LLL1 had a stationary price. This usually means that the contract is **not traded** anymore. Therefore, we decided not to use it.

BUNDLE SELECTION

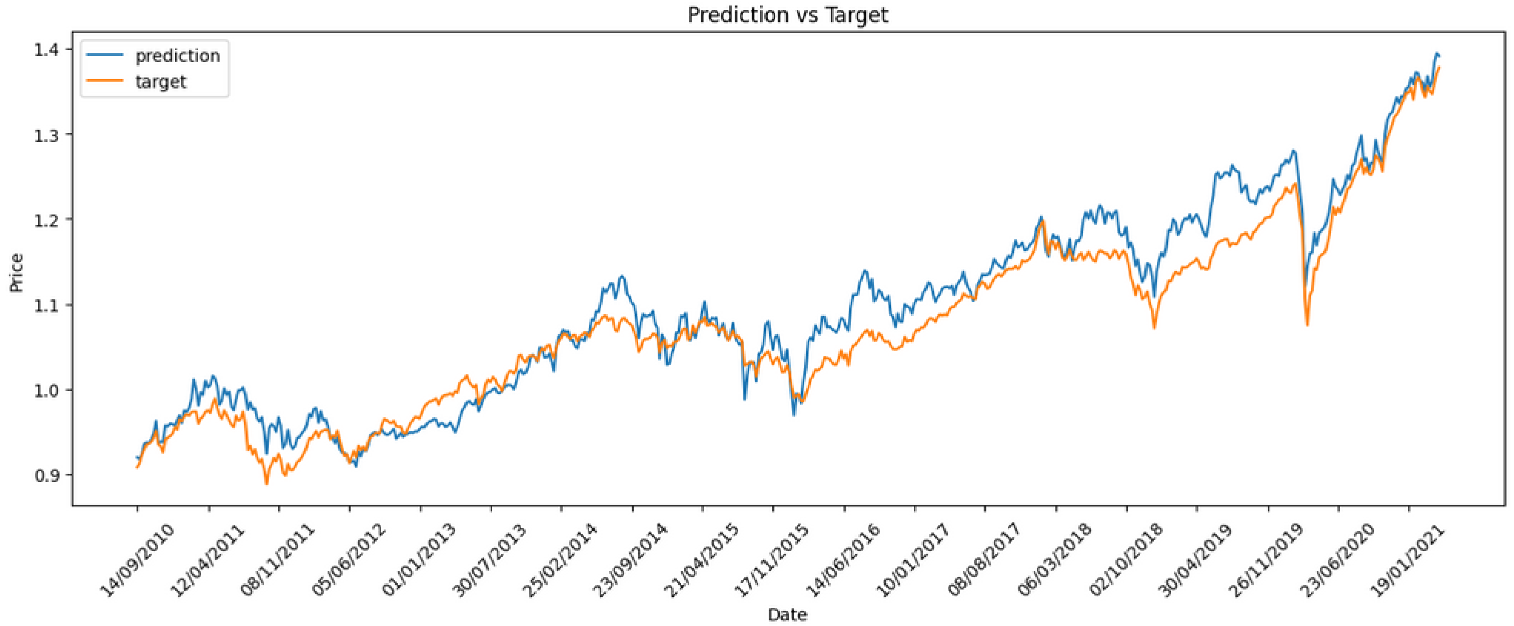
For the app implementation, we decided to use **3 bundles** of futures. We selected these bundles analyzing the correlation between the available futures. In this choice, we also considered the mean and variance of the contracts.

ANALYSIS FOR NN

Since Neural Network models need **stationary and standardized data** to work, we performed a stationarity test on the data and switched to returns. Finally, we normalized them by means of z-scoring.

REGRESSION MODELS 1/2

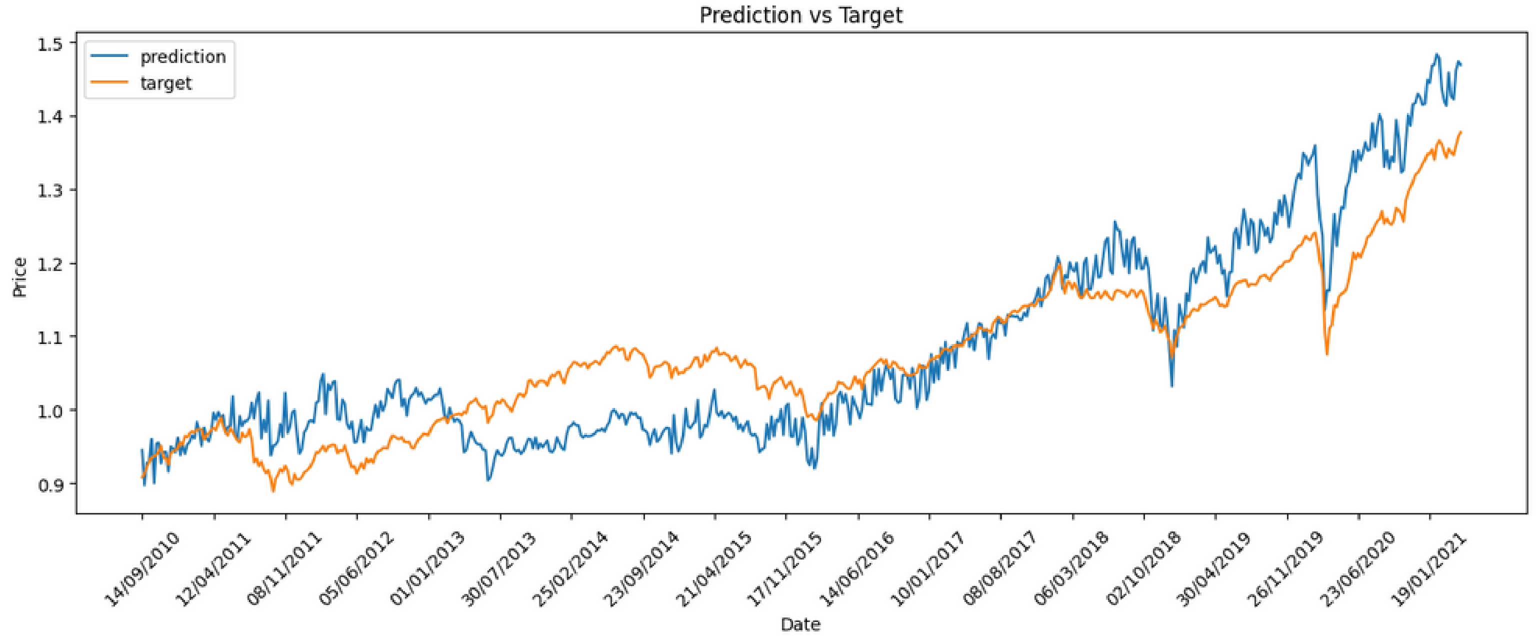
LASSO REGRESSION



- Through l1 penalty, introduces **regularization**
- Being a **sparse** model, brings to the table a sort of feature selection
- Generates **biased** coefficients, that tend to zero
- Struggles with **correlated features**

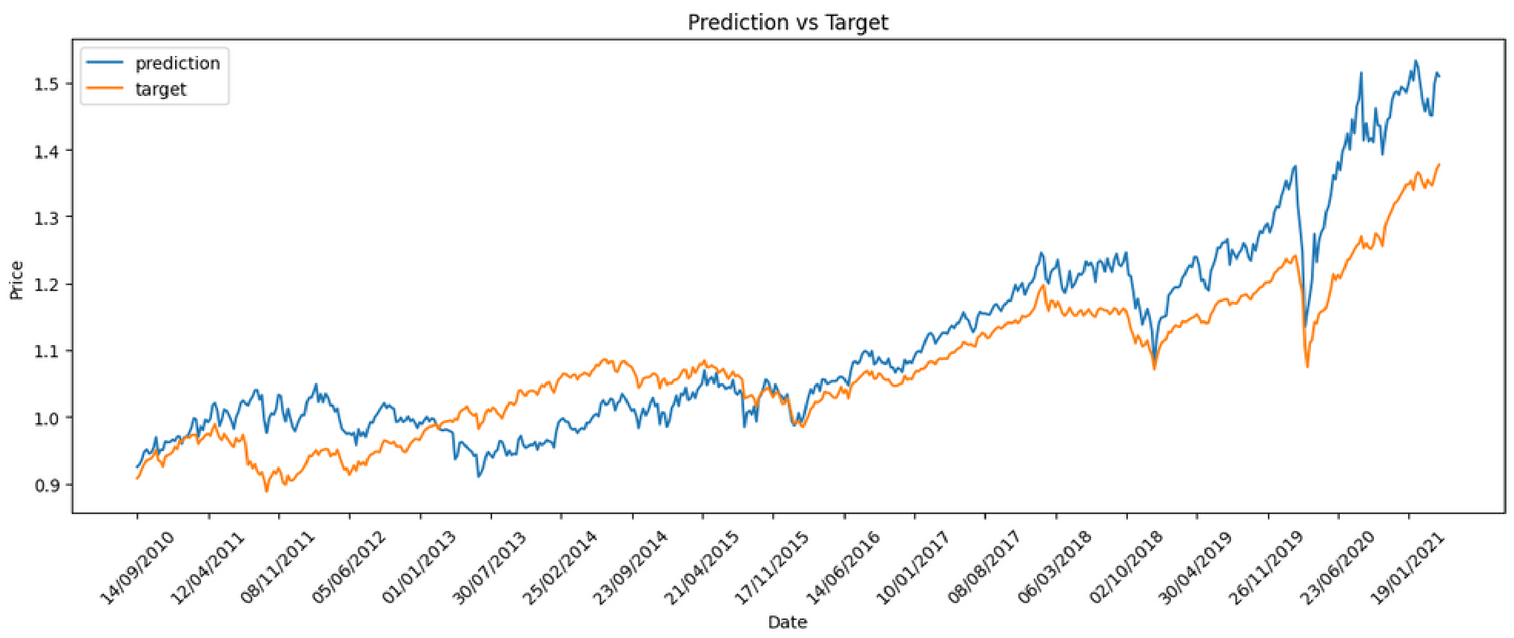
PASSIVE AGGRESSIVE REGRESSOR

- Being an approach used mainly for **online learning**, it applies well to our setting
- Needs **huge quantities** of data to estimate properly
- It's **not widely used,** so there's not a lot of literature about model optimization



REGRESSION MODELS 2/2

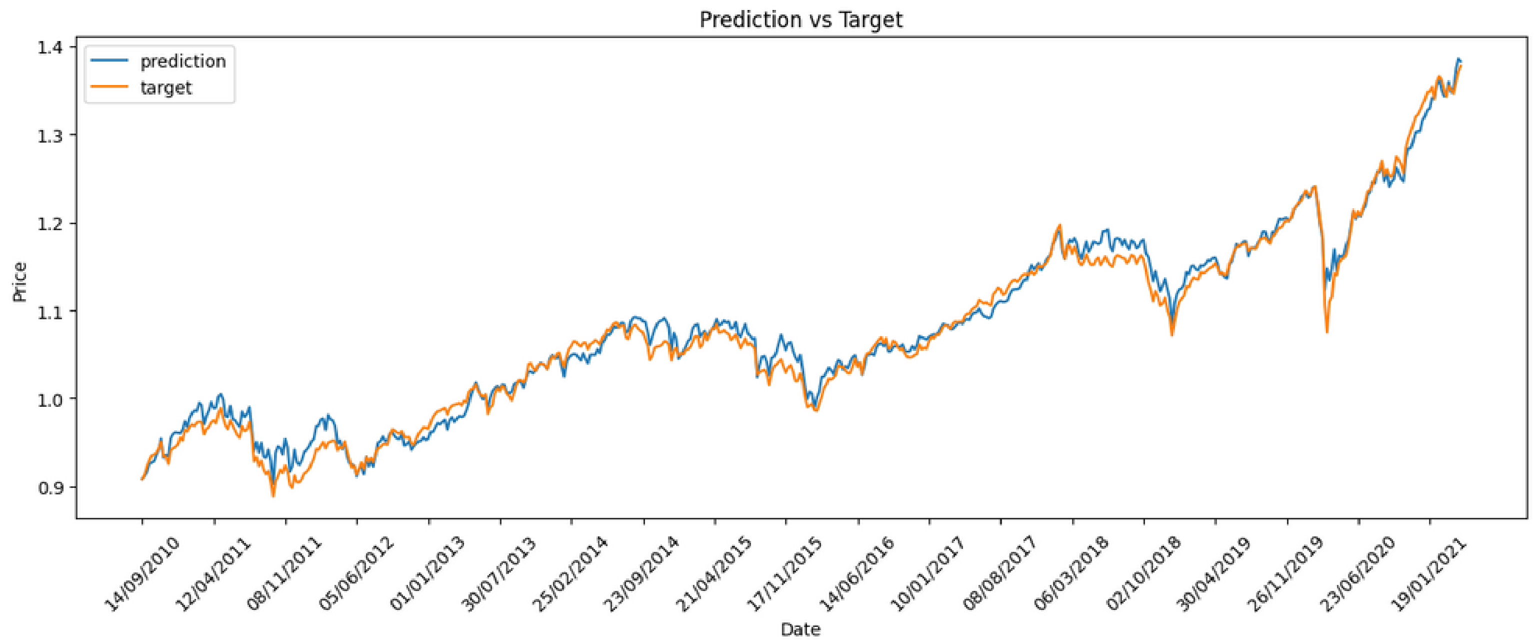
STOCHASTIC GRADIENT DESCENT



- **Efficient** and easy to implement
- Supports multiple choices of **loss and penalties**
- Needs a lot of **hyperparameter tuning**
- Needs **big quantities** of data to estimate properly

ELASTIC NET

- Combines l1 and l2 penalties, taking the best from both Ridge and Lasso in terms of **regularization**
- Like Lasso, it is a **sparse** model, implementing a sort of feature selection
- Needs **tuning** for parameter alpha (solved through **grid-search**)

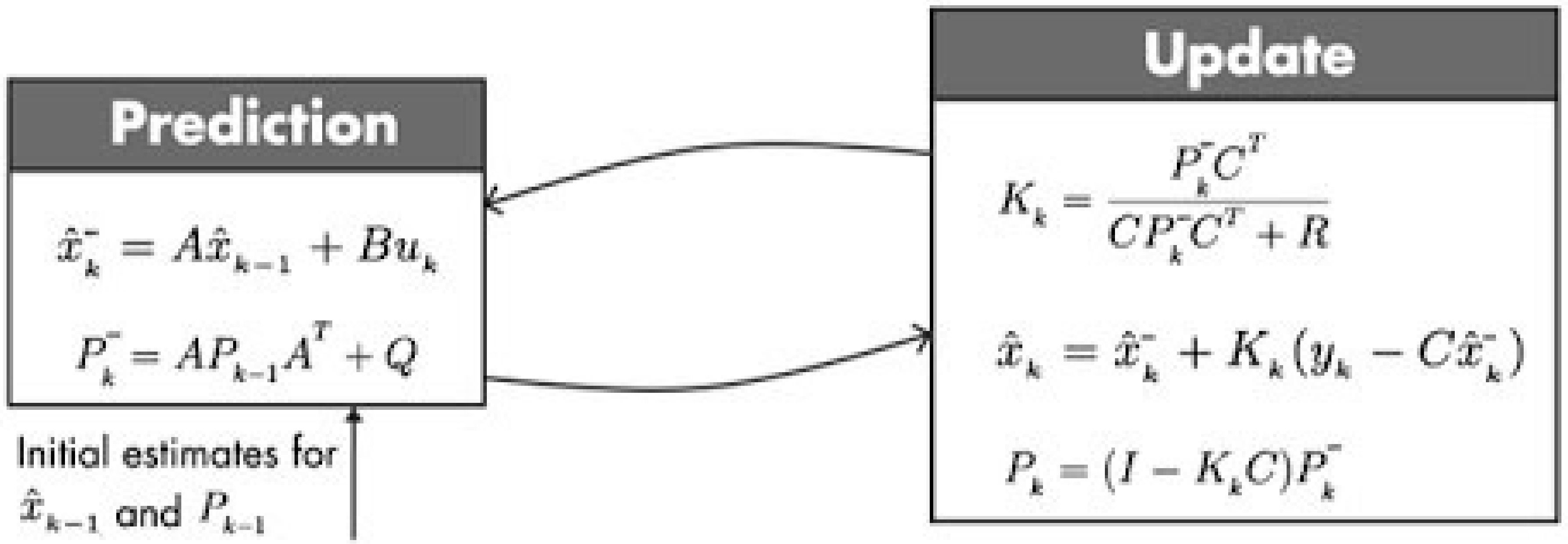


KALMAN FILTER

A powerful mathematical tool for **estimation** of **hidden states** leveraging a sequence of noisy observations.

- In the realm of investments, K.F. **estimates the weights (states)** of the portfolio replica by assuming **linearity** in the state transition process and **mapping** to the target **returns (output)**.

- K.F. is particularly suitable for **dynamic environments** capturing **evolving dynamics** effectively and dealing with **noises**.



K.F. IMPLEMENTATION

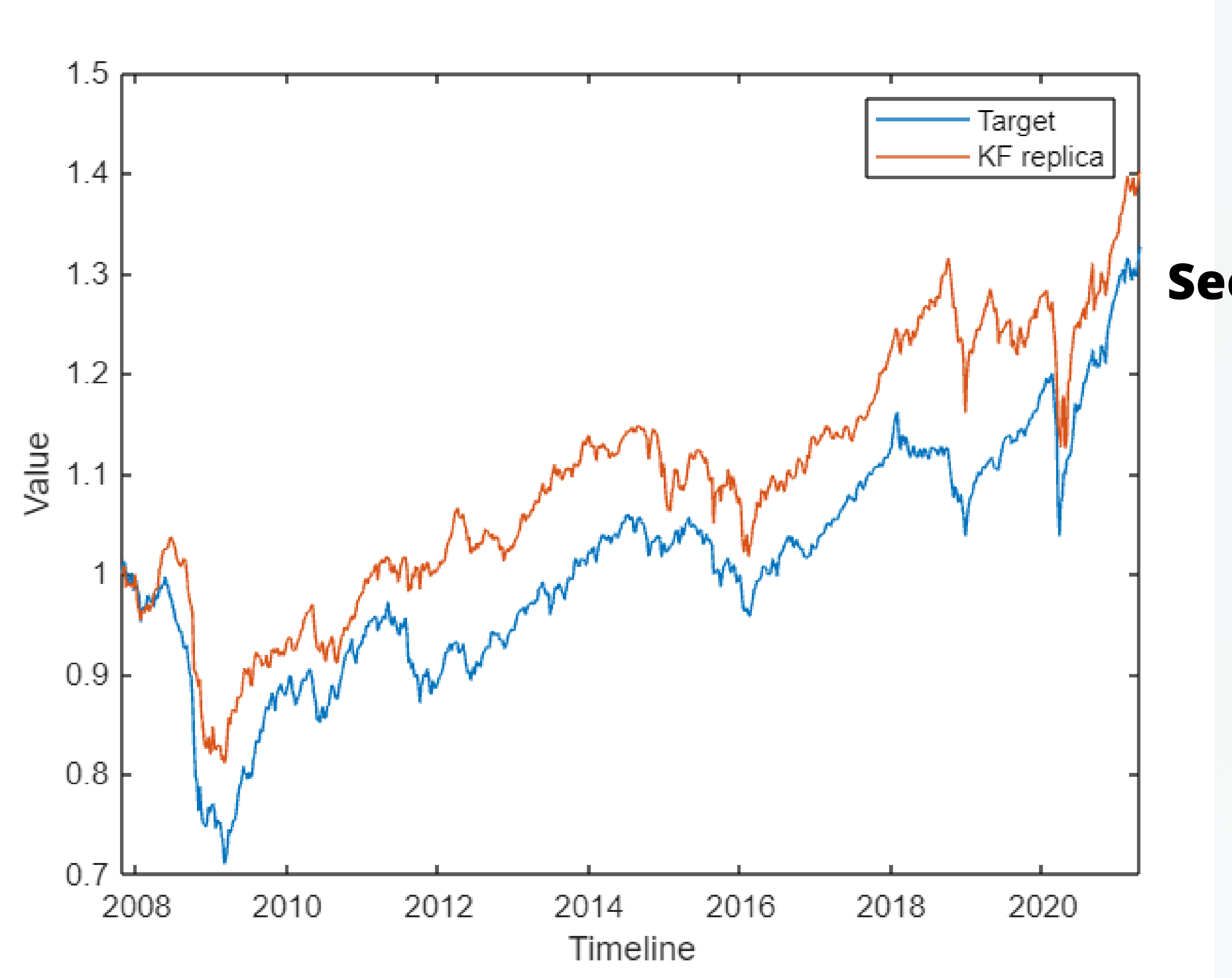
- **Initially**, the filter was implemented by **identifying the noises** using **sample variances** on futures returns and **comparing** the **prices** of the target index and futures. The **need for refinement** techniques through a meticulous **sensitivity analysis** of the filter was evident. A **standard backtest approach** was used.

K.F. with sample variances

Sensitivity analysis

See appendix

K.F. with conditional variances



- At a **graphical level**, a **significant improvement** is already evident in the replica portfolio, which closely tracks the prices of the target index.

- **Previous InfoRatio** = -0.0092 **TrackingError** = 0.0147
  Present InfoRatio = 0.0102 **TrackingError** = 0.0077

Let's compare *the latter* with the other regression models...

MODEL COMPARISONS 1/2

MEAN SQUARED ERROR

MSE	
Lasso	0,000959709
PAR	0,00424049
SGD	0,00519614
EN	0,000177529
KF	0,000059318

Mean Square Error is mininized by *Kalman Filter* method, suggesting it performs the best in terms of predictive accuracy.
However, also *Elastic net* and Lasso regression provide relative low MSE values.

TRACKING ERROR VOLATILITY

TEV	
Lasso	0,175779
PAR	0,467084
SGD	0,460137
EN	0,0893928
KF	0,0544

Annualized standard deviation of tracking error is particularly reduced using *Kalman Filter* and *Elastic Net* methods.
This suggests more **stability** and **less variability** in the futures returns compared to target perfomance.

MODEL COMPARISONS 2/2

MEAN ANNUAL TURNOVER

MAT	
Lasso	0,720391
PAR	1,59086
SGD	0,409487
EN	0,239801
KF	10,0275

Methods with lower mean annual turnover as _Elastic net_ and _SDG_ show a more **passive approach** with fewer changes in asset holdings.
On the other hand, a higher mean annual turnover indicates a **higher level of trading activity**, with possibly **higher trading costs**.

MEAN ANNUAL TRADING COSTS

MATC	
Lasso	0,000288156
PAR	0,000636345
SGD	0,000163795
EN	0,000095203
KF	0,0018

As expected, _Kalman Filter_ shows higher mean annual trading costs due to its **high trading frequency**.
On the contrary, _Elastic Net_ shows the lowest value of such costs, positioning itself as ***the most convenient cost-effective portfolio***.

RISK MEASUREMENT

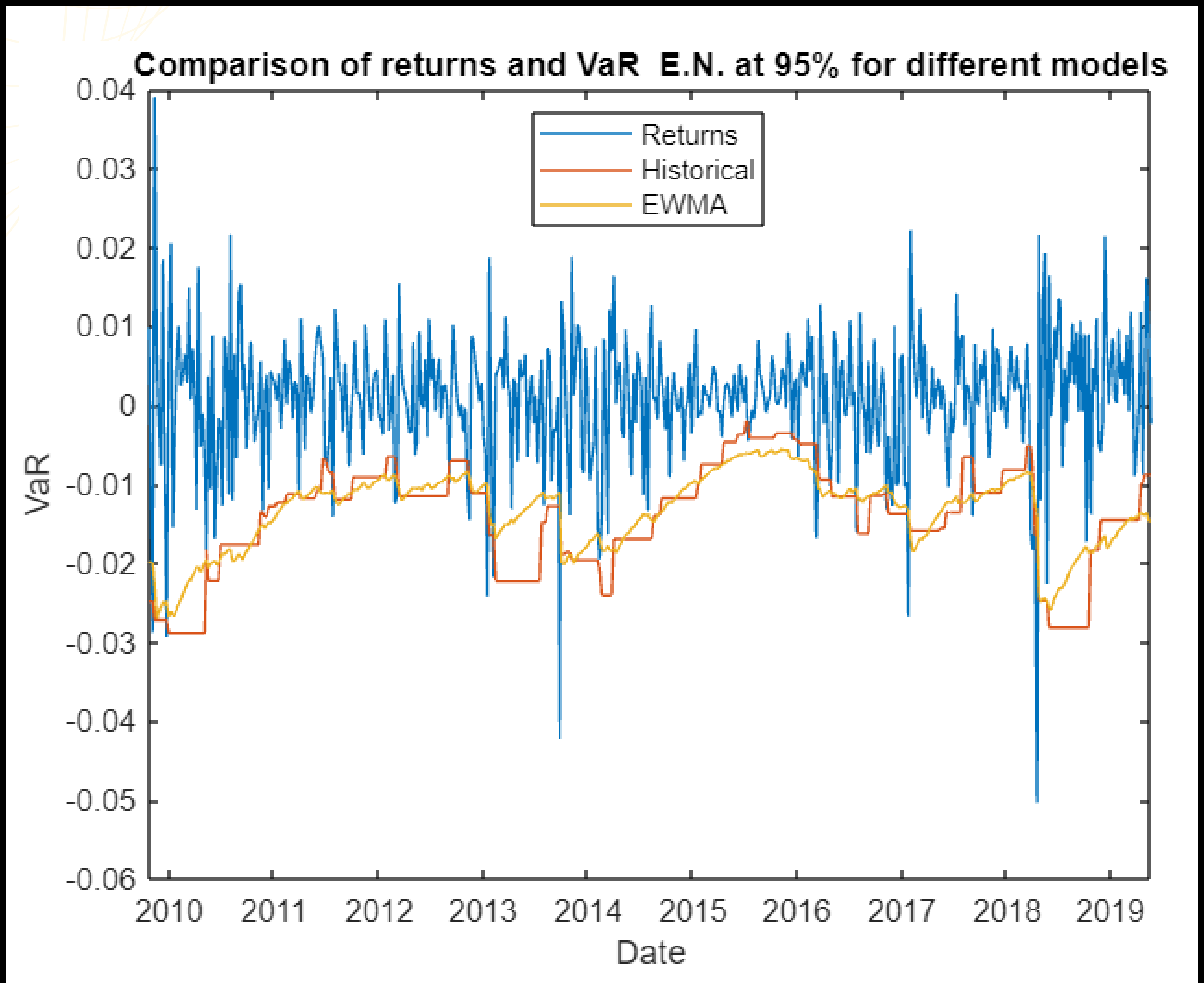
Based on the previous analysis, it can be concluded that *the most promising methods are **Elastic Net** and **Kalman Filter***.

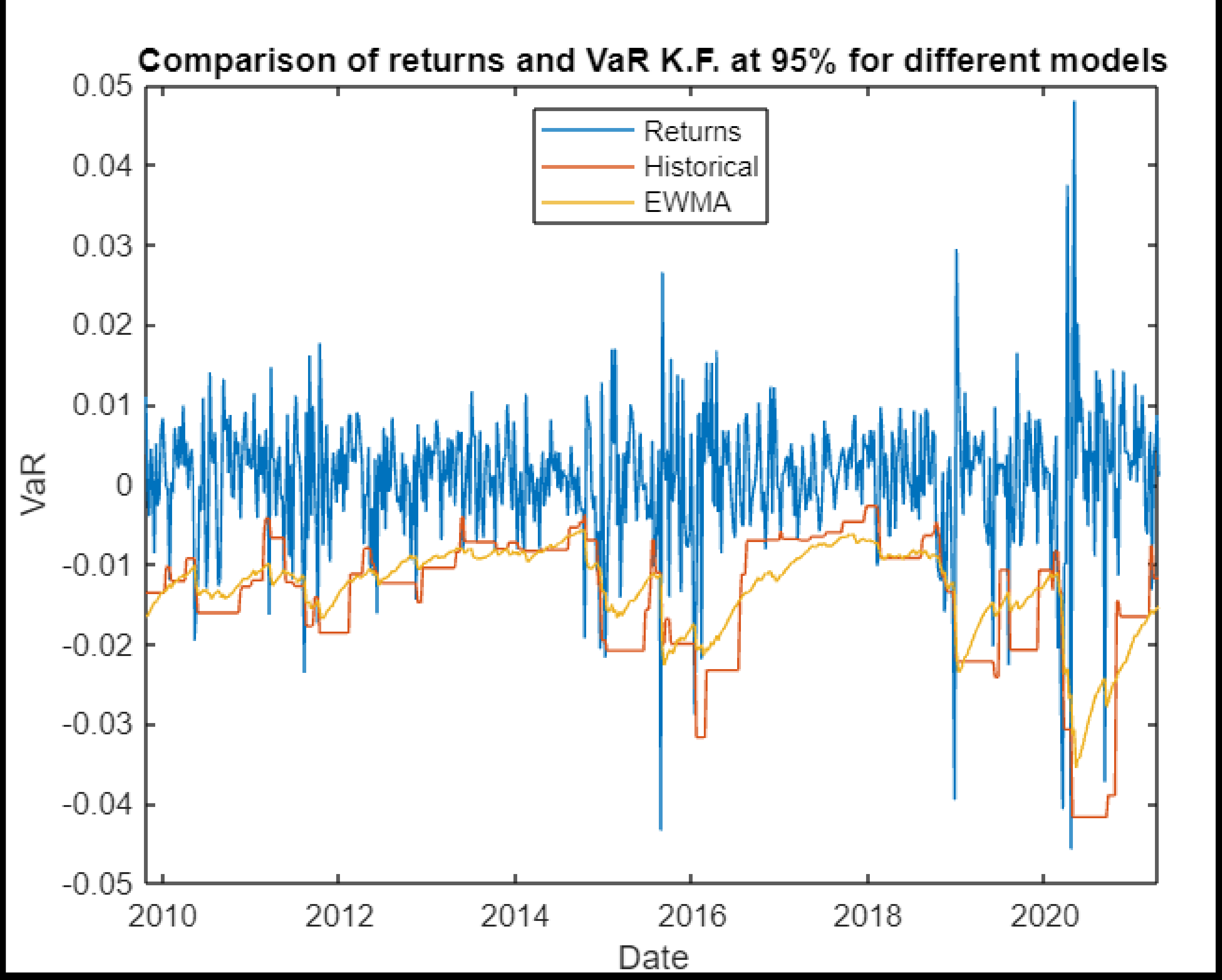
We considered **VaR based on historical simulation** and **EWMA** method to estimate the potential loss of the portfolio at *the confidence levels 99% and 95%*. The first method roots at historical data and, the latter, assigns weights to past observations.

	Historical 95	Historical 99	EWMA 95%	EMWA 99%
EN	[0.0021, 0.0289]	[0.0040, 0.0502]	[0.0056, 0.0269]	[0.0079, 0.0381]
KF	[0.0027, 0.0417]	[0.0032, 0.0457]	[0.0058, 0.0355]	[0.0082, 0.0502]

The two methods exhibit different sensitivities to recent market dynamics and, overall, both have low risk exposure.

We believe that the optimal method for app development is the **Elastic Net**, a stable and accurate approach with low risk and reduced trading costs.





APP

- The idea is to create an app that allows customers to explore their portfolio needs based on the desired bundle of futures
- The three bundles available are **BASE**, **SURE-FIRE**, and **DARK HORSE**, and a detailed description is available on the app
- It provides an **easy** and **accessible** way to visualize the pros and cons of each portfolio replica, making it easier for the costumer to choose his strategy

(click the icon on the top left to get to the app)

Streamlit

WHAT'S NEXT?

Possible future App developments

Adding the opportunity to select which model to use for the replication, choosing the best performing in terms of the customer's objective

MULTIPLE MODELS SELECTION

Widening the available data, in terms of contracts, would for sure allow a better portfolio replica, considering also the desired risk and return target

WIDER DATA

Adding the possibility to choose the combination of contracts to use in the portfolio replica, could allow a better fitting to the desired risk and return target

MIX AND MATCH

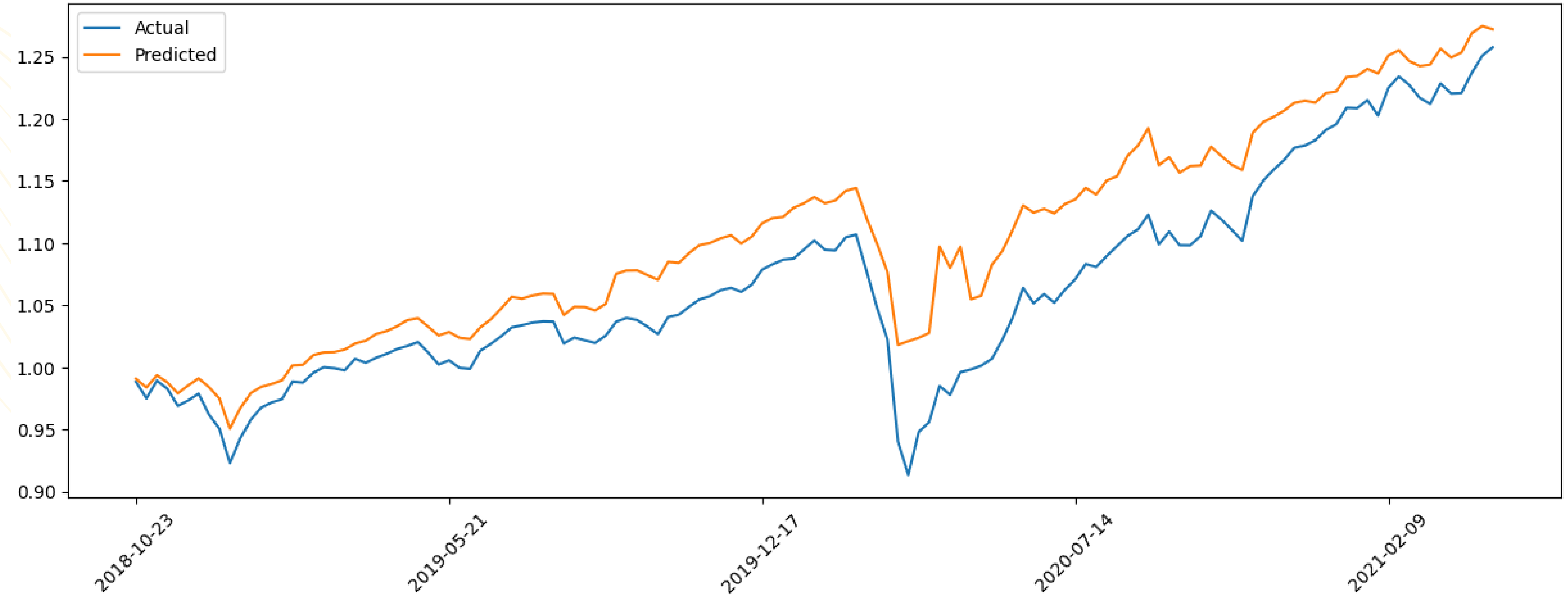
APPENDIX: NEURAL NETWORKS

Recurrent Neural Networks (RNN) are state-of-the-art tools that can capture deep long-run relationships.
We used a **Long-Short Term Memory System** to replicate the target avoiding big dips in the returns.

- Custom fit to our needs
- Loss in **explainability and feasibility**
- Grid search of many hyperparemeters
- Needs a lot of data to become reliable

Therefore, we decided to put aside this model

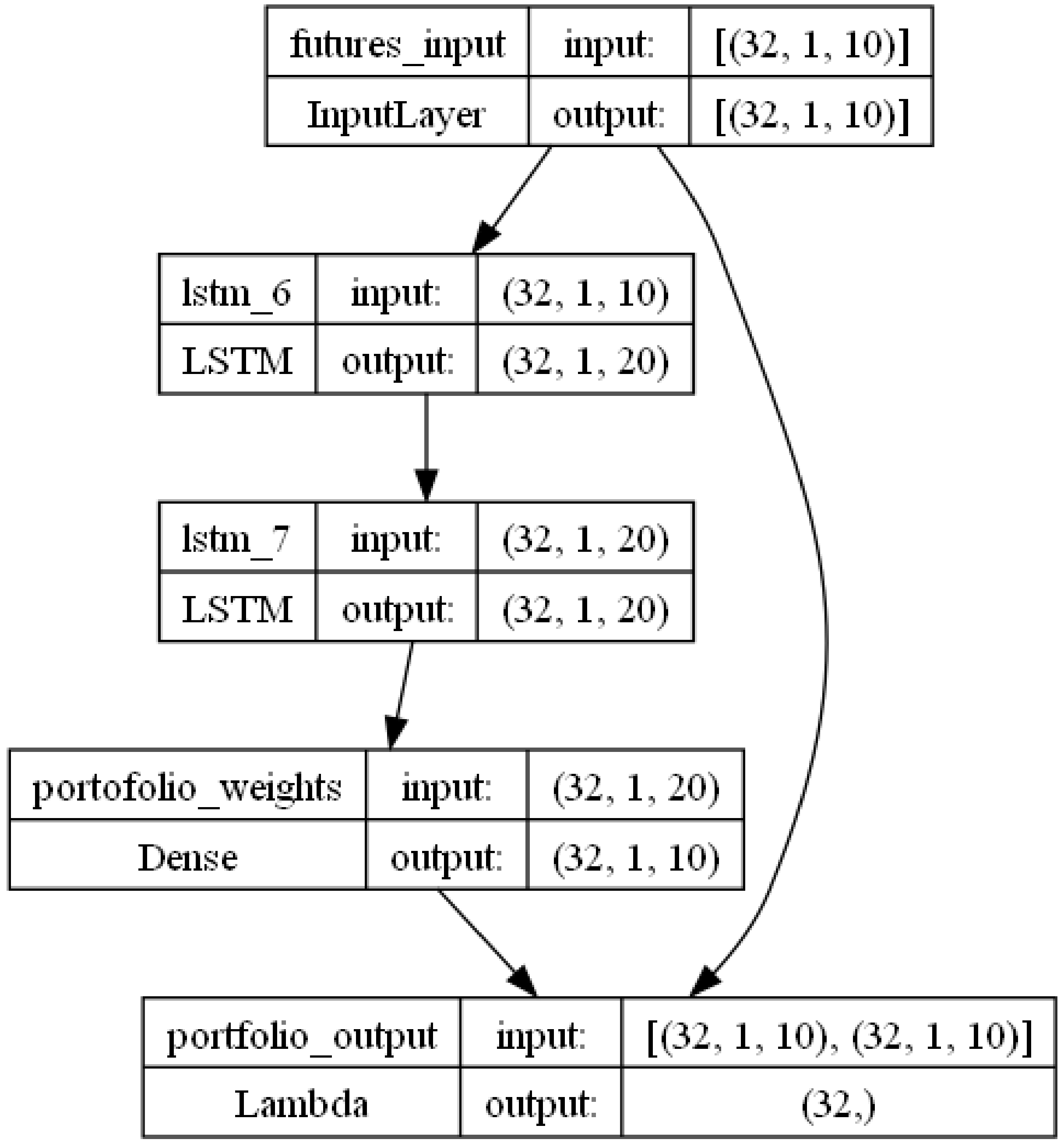
ACTUAL VS PREDICTED PRICES



APPENDIX: RNN ARCHITECTURE

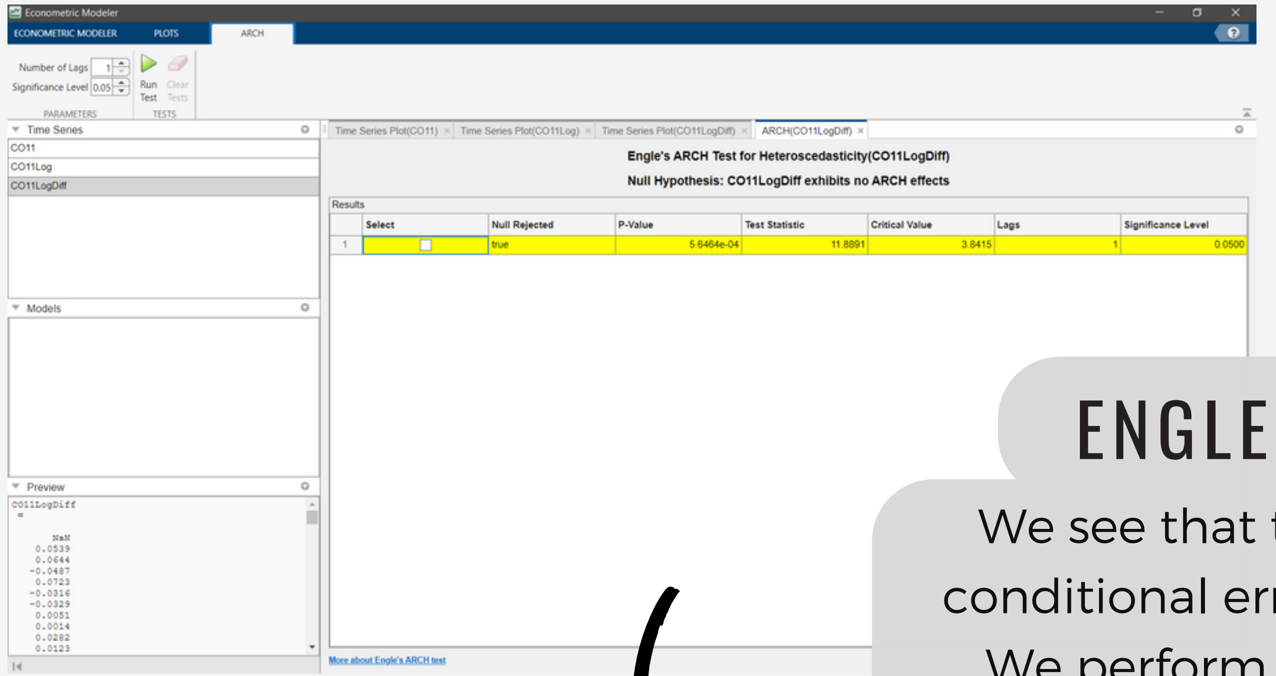
We used a stateful LSTM:

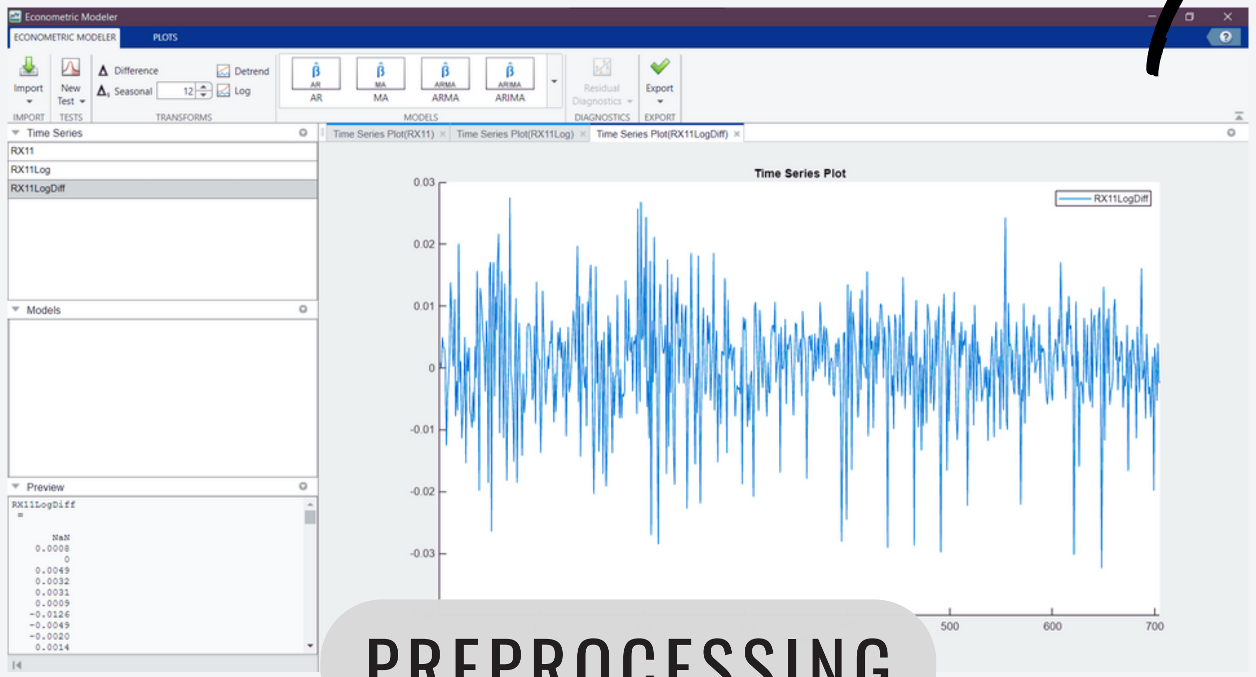
- **2 Hidden LSTM layers** with 20 neurons each and ReLU activation function
- Lambda layer to perform dot product between weights and returns
- **Custom loss function** to avoid dips in the portfolio performance
- Eliminates the need for a fixed rolling window



APPENDIX: SENSITIVITY ANALYSIS K.F.

The only **degree of freedom** available to **enhance** the Kalman filter is to **manipulate the noise.** Exploring a **variance estimation model** enables a **reduction in estimation error**.



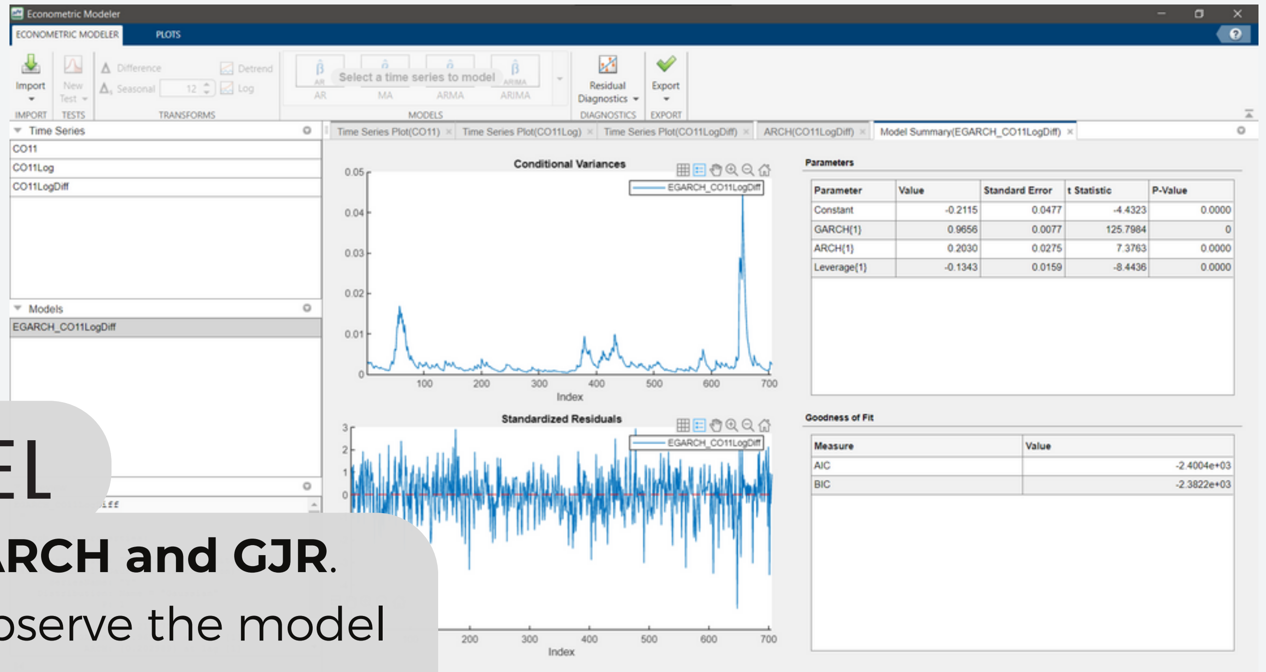


PREPROCESSING

We exploit the **Econometrics Modeler** App in **Matlab**. We give prices in input and perform *logarithmic* and *differences transformation* to obtain **logReturns**.

ENGLE'S ARCH TEST

We see that the **logreturns** exhibit conditional error **heteroscedasticity**. We perform **Engle's ARCH test** to prove heteroscedasticity.



EGARCH MODEL

Then we perform **GARCH**(1,1), **EGARCH and GJR**. Looking at the **goodness of fit**, we observe the model with **lowest AIC and BIC** is the *EGARCH*. Finally, we perform **residual diagnostics with Q-Q plot.** The residuals are approximately normal.

REFERENCES

- *“Asset Replication” by Rupak Chatterjee, 2014*
- *"Linear Models Scikit Learn User Guide" (sections about PAR and SGD Models)*
- *"Online Passive-Aggressive Algorithms"* by K. Crammer, O. Dekel, J. Keshat, S. Shalev-Shwartz, Y. Singer - JMLR (2006).
- *“An Introduction to the Kalman Filter” by Gary Welch & Gary Bishop, 2006*
- *“Introduction to Random Signals and Applied Kalman Filtering.” by Robert Grover Brown & Patrick Y. C. Hwang, 2012*
- *"Value-at-Risk Estimation and Backtesting", MathWorks*
- *"Understanding LSTM Networks", Christopher Olah*
- *"Stateful and Stateless LSTM for Time Series Forecasting with Python", Jason Brownlee*

THANK YOU!

"An investment in knowledge pays the best interest."
- Benjamin Franklin