Homework 02

Summary:

Adaptive Conformal Inference

Data analyzed: Microsoft returns

Naive strategy

AgACI

Group:

Elisa Battista

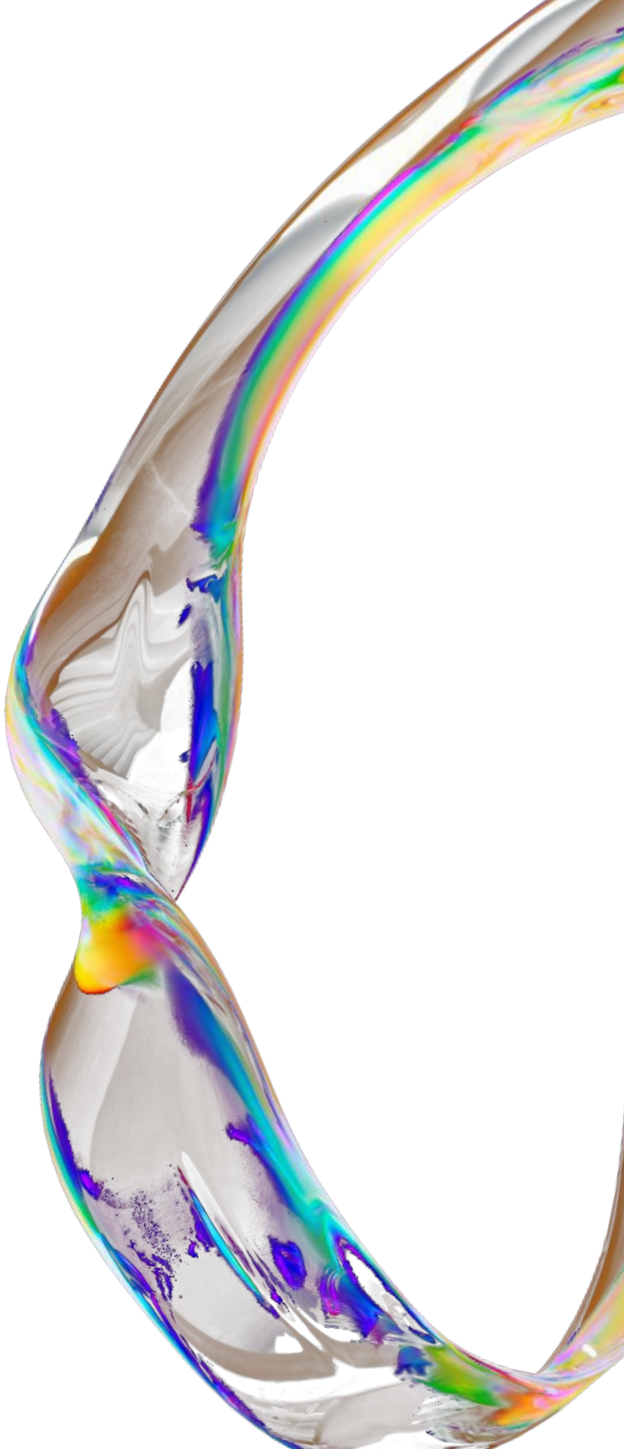
Simone Cuonzo

Lorenzo di Giannantonio

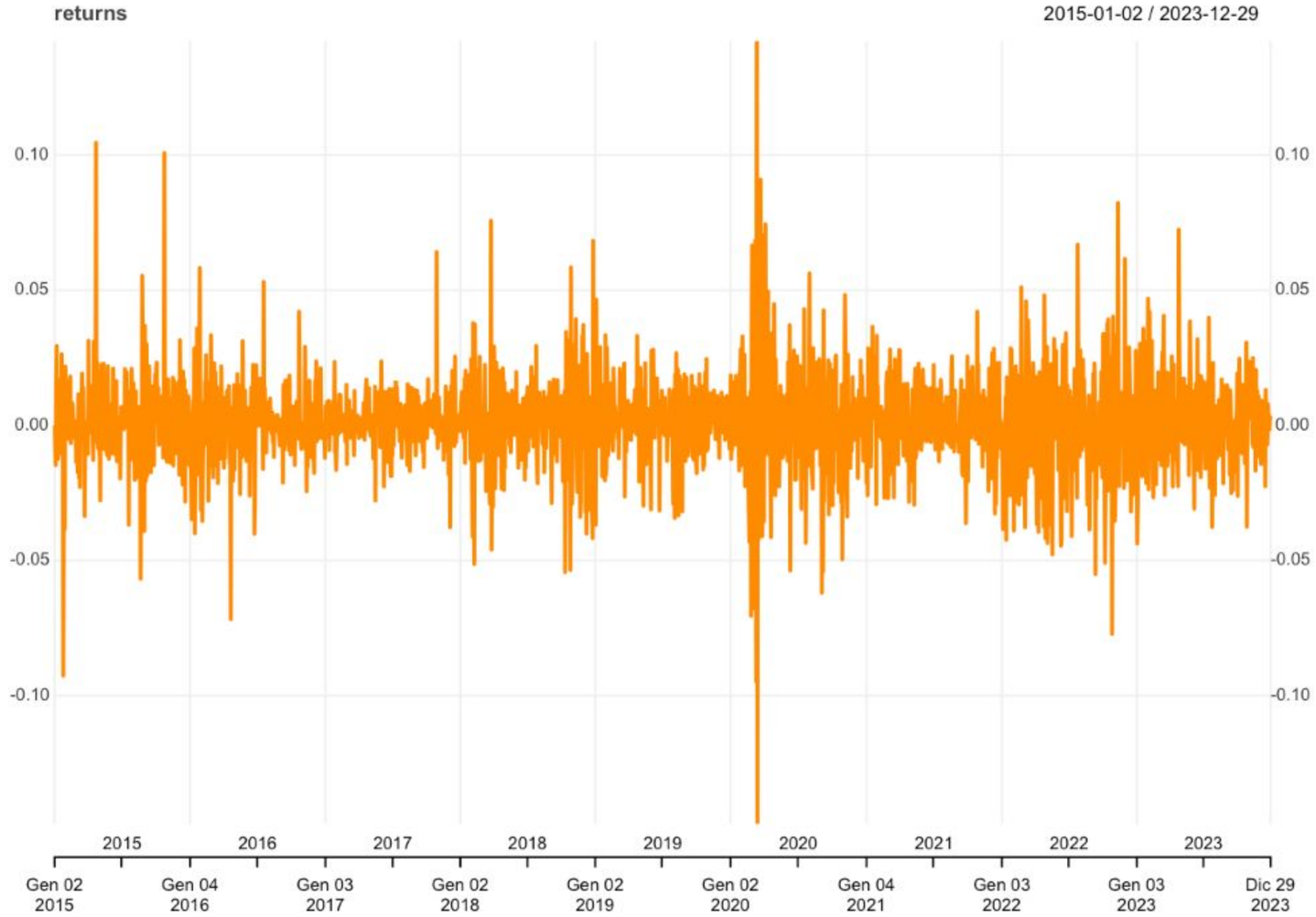
Adaptive Conformal Inference

Adaptive Conformal inference is a method to build prediction sets that are robust to changes in the marginal distribution of the data

- it is designed to adapt to arbitrary **distribution shifts**
- instead of a target level alpha, here we use a **parameter** which will change in time
- intuitively what happens is that if the interval did not cover at t-1 (so we have an error=1) the parameter of interest, alpha, is reduced to make the interval wider
- **the cumulative mean of error** at each time plays the role of the estimate of the miscoverage frequency
- we deal also with a new adaptation parameter called gamma, which is crucial in terms of **adaptability and stability** of the model.

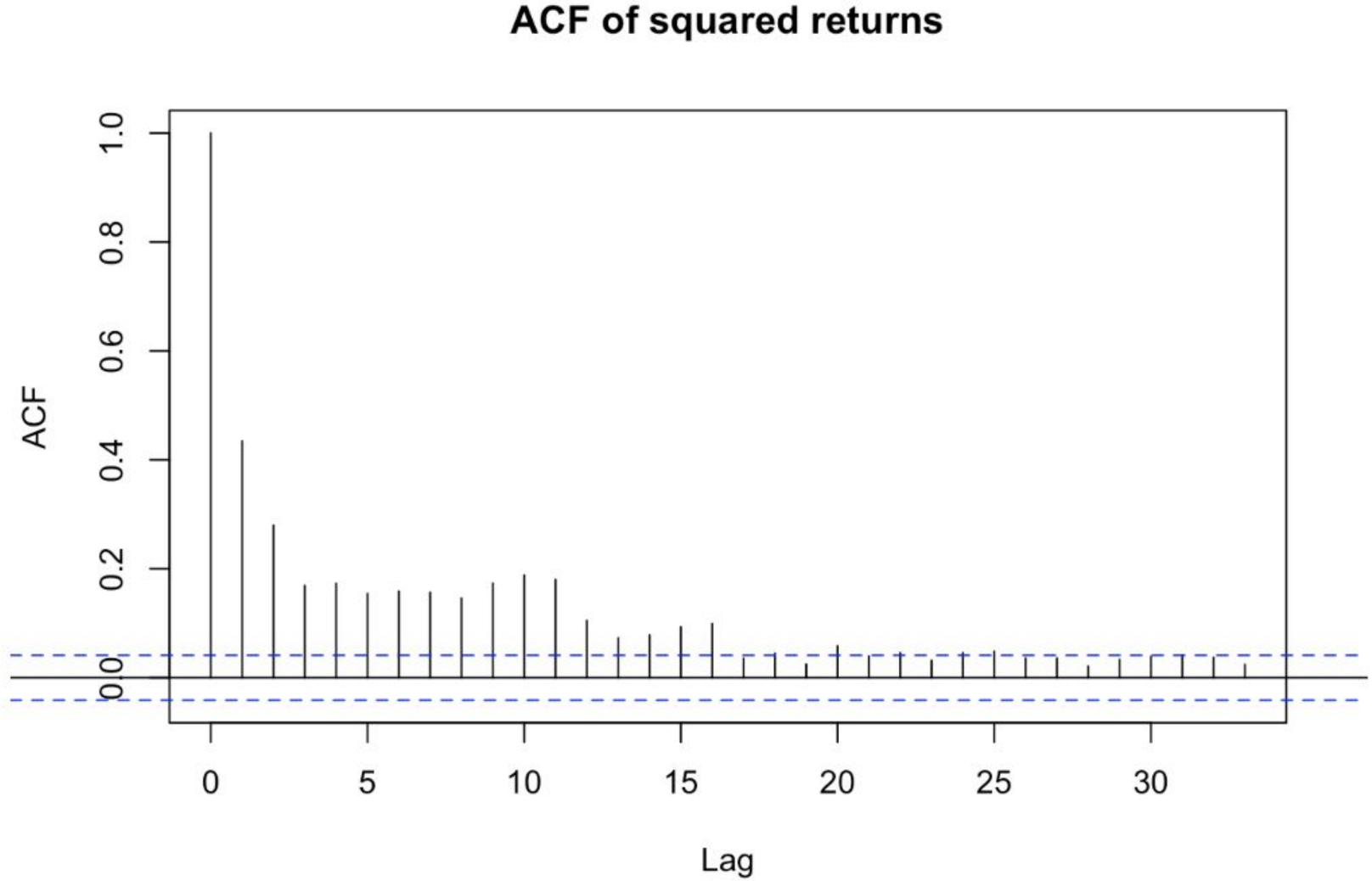


Data analyzed: Microsoft returns



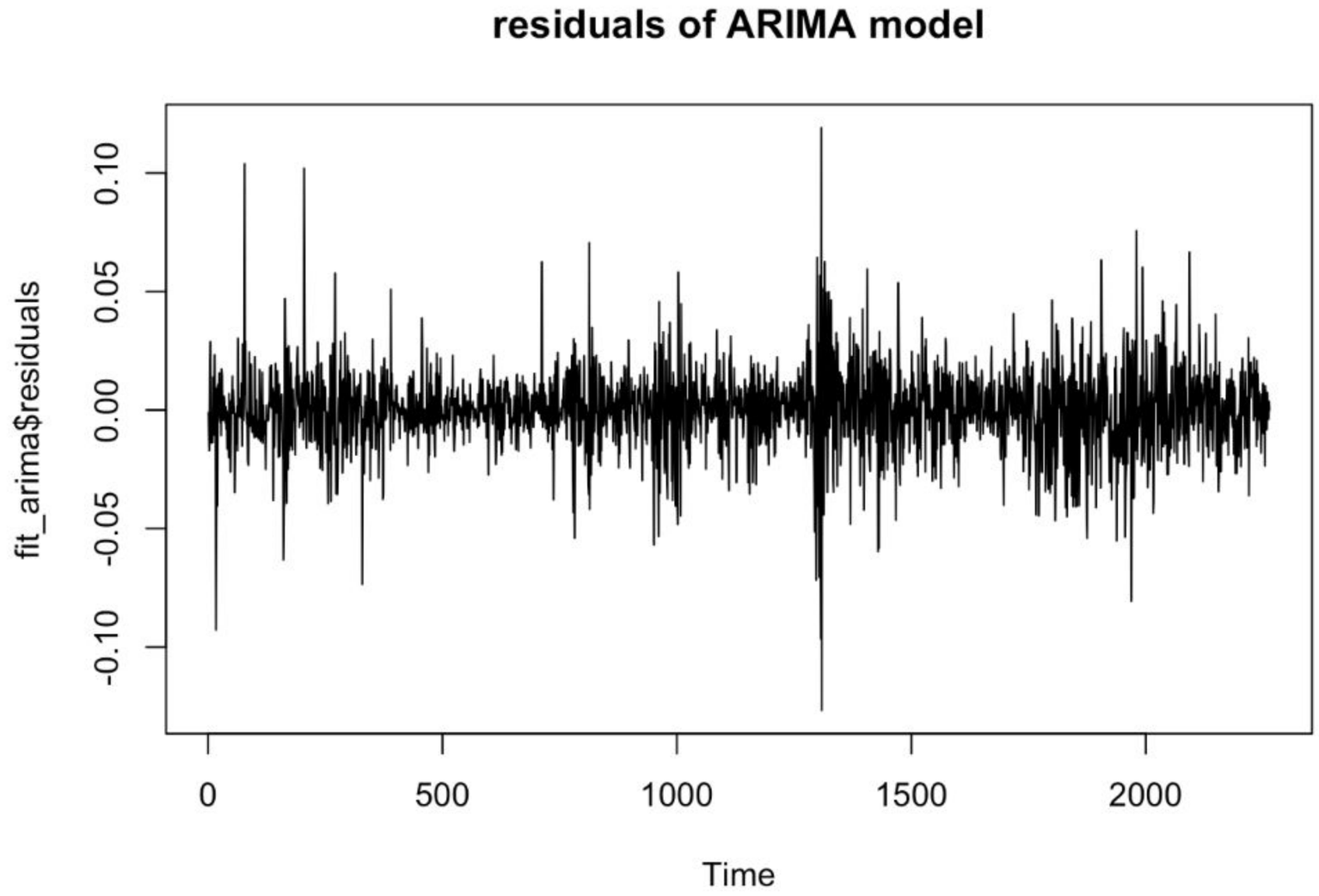
Returns have almost zero mean but it is clear that we have some volatility cluster

Data analyzed: Microsoft returns



ACF of squared (and absolute) returns is positive and it is linked to ARCH effects

Data analyzed: Microsoft returns



If we fit an ARIMA model we notice that the residuals doesn't look "white" meaning that we should combine a GARCH model to fully capture the behaviour of the data

Models used:

- **Standard GARCH model with normal distribution**
- **The second best GARCH model in terms of AIC, BIC (since the first one required too much computational time to run the algorithm): eGARCH**

It is important to formalize the **error** defined in the previous slide as:

$$\text{err}_t := \begin{cases} 1, \text{ if } Y_t \notin \hat{C}_t(\alpha_t), \\ 0, \text{ otherwise,} \end{cases} \qquad \text{where } \hat{C}_t(\alpha_t) := \{y : S_t(X_t, y) \leq \hat{Q}_t(1-\alpha_t)\}.$$

For each model we run ACI using:

- **two score functions**:

simple: $S_t := |V_t - (\hat{\sigma}_t^t)^2|$

normalized: $S_t := \dfrac{|V_t - (\hat{\sigma}_t^t)^2|}{(\hat{\sigma}_t^t)^2}$

- **two adaptation methods**:

simple: $\alpha_{t+1} := \alpha_t + \gamma(\alpha - \text{err}_t).$

momentum:

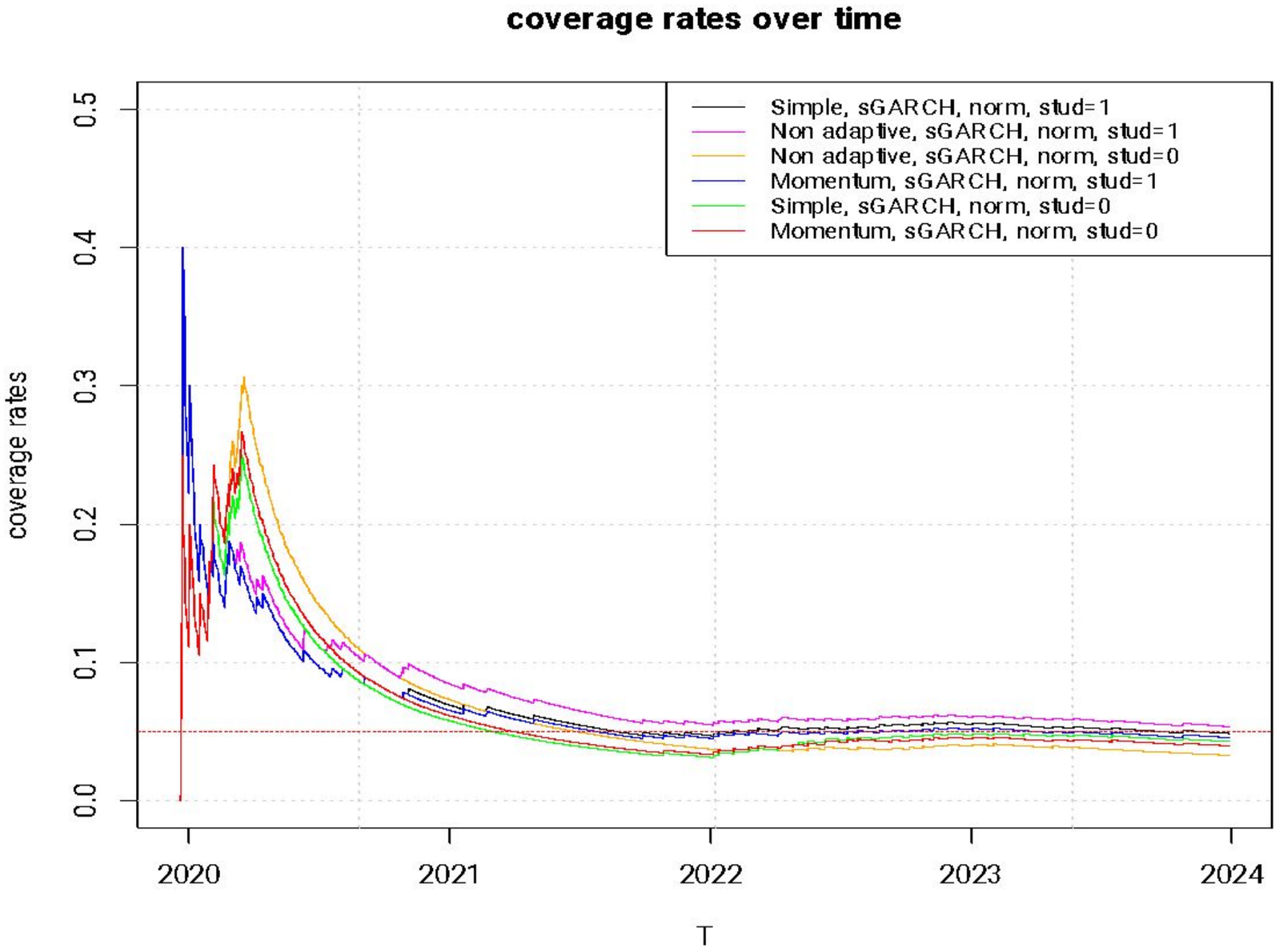
$$w_s := \frac{0.95^{t-s}}{\sum_{s'=1}^{t} 0.95^{t-s'}}$$

$$\alpha_{t+1} = \alpha_t + \gamma \left(\alpha - \sum_{s=1}^{t} w_s \text{err}_s \right)$$

To measure the performance of these methods across time we examine their local miscoverage frequencies defined as the average miscoverage rate

$$\frac{\sum_{s=1}^{t} \text{err}_s}{t}$$

If the methods perform well then we expect the local miscoverage frequency to stay near the target value alpha across all time points.



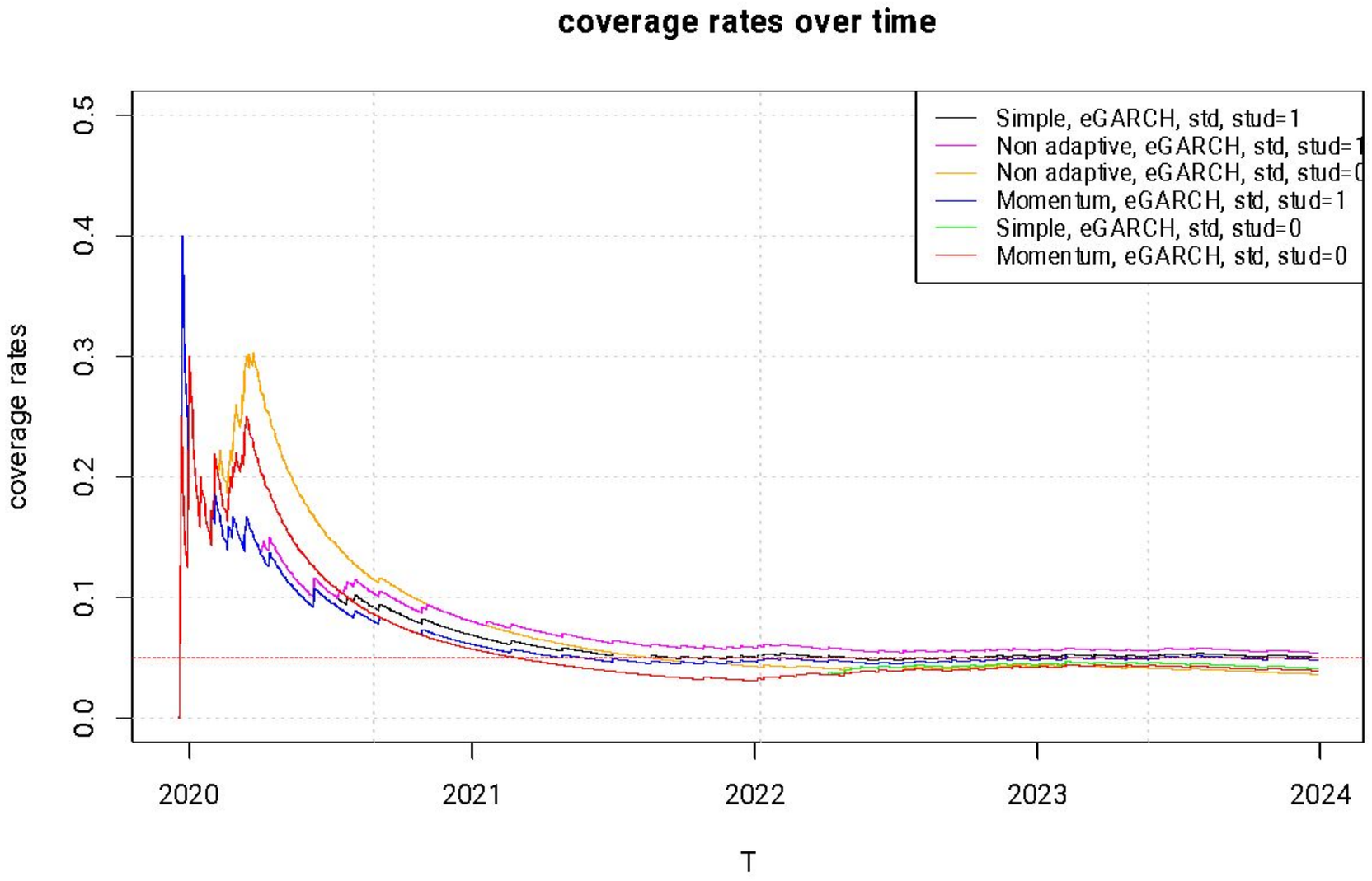
Key points:

- Best performances are achieved when a normalized score is used.
- The non-normalized score amplifies the impact of extreme events and distributional shifts, causing frequent oscillations in *errt*.
- The authors note that adopting the "momentum" strategy does not produce results significantly different from the classical approach.
- We observe that the non-adaptive strategy initially outperforms the adaptive approach when using a non-normalized score.
- Coverage for small values of t is particularly important, as the graph shows all methods perform well in the long run.
- The green and red graphs, which rely on non-normalized errors, exhibit poor performance.

Key points:

- The improved results can be attributed to the use of the ARMA(1,1)-eGARCH(1,1) model with t-student noise, which more effectively captures the underlying data distribution and volatility.

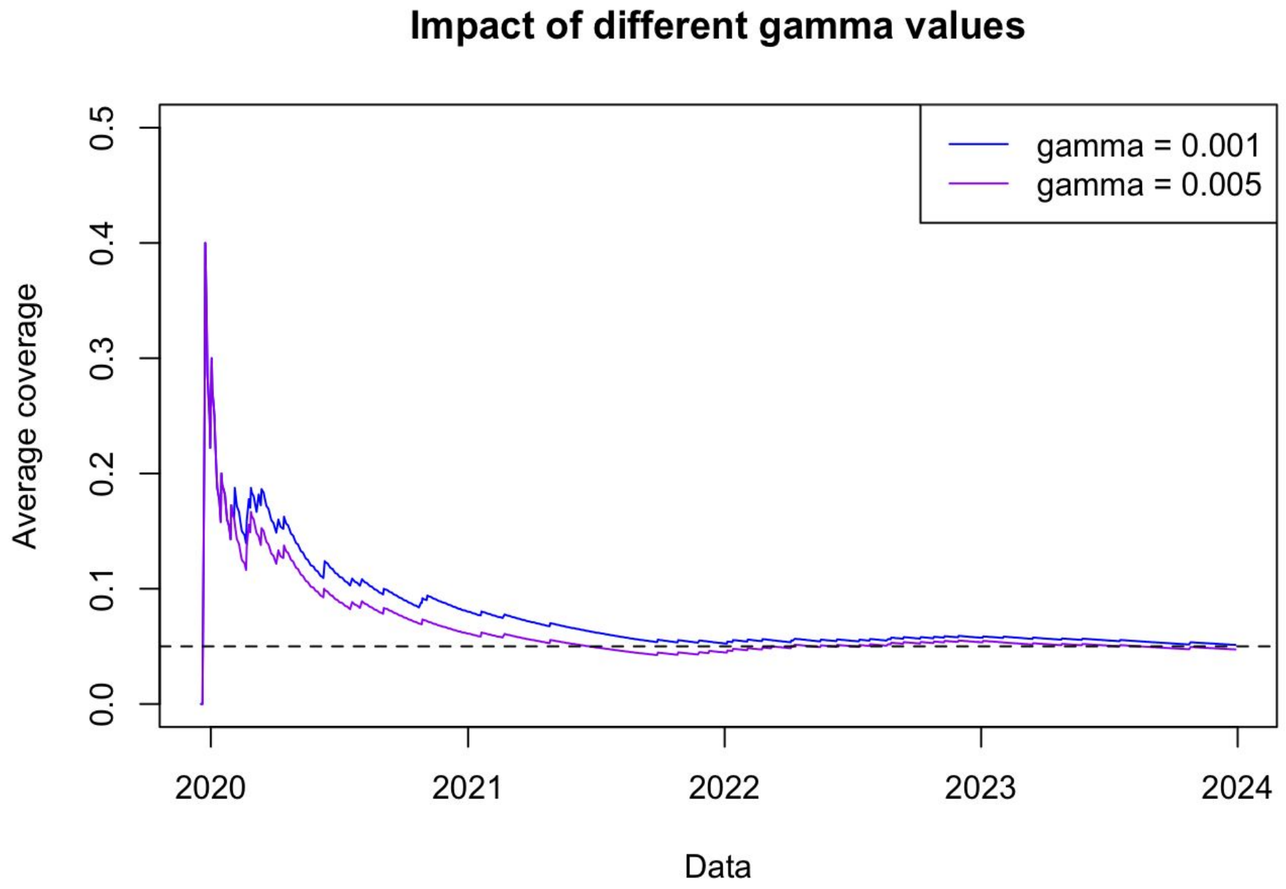
- Selecting an appropriate model is therefore critical for enhancing the performance of the adaptive algorithm.



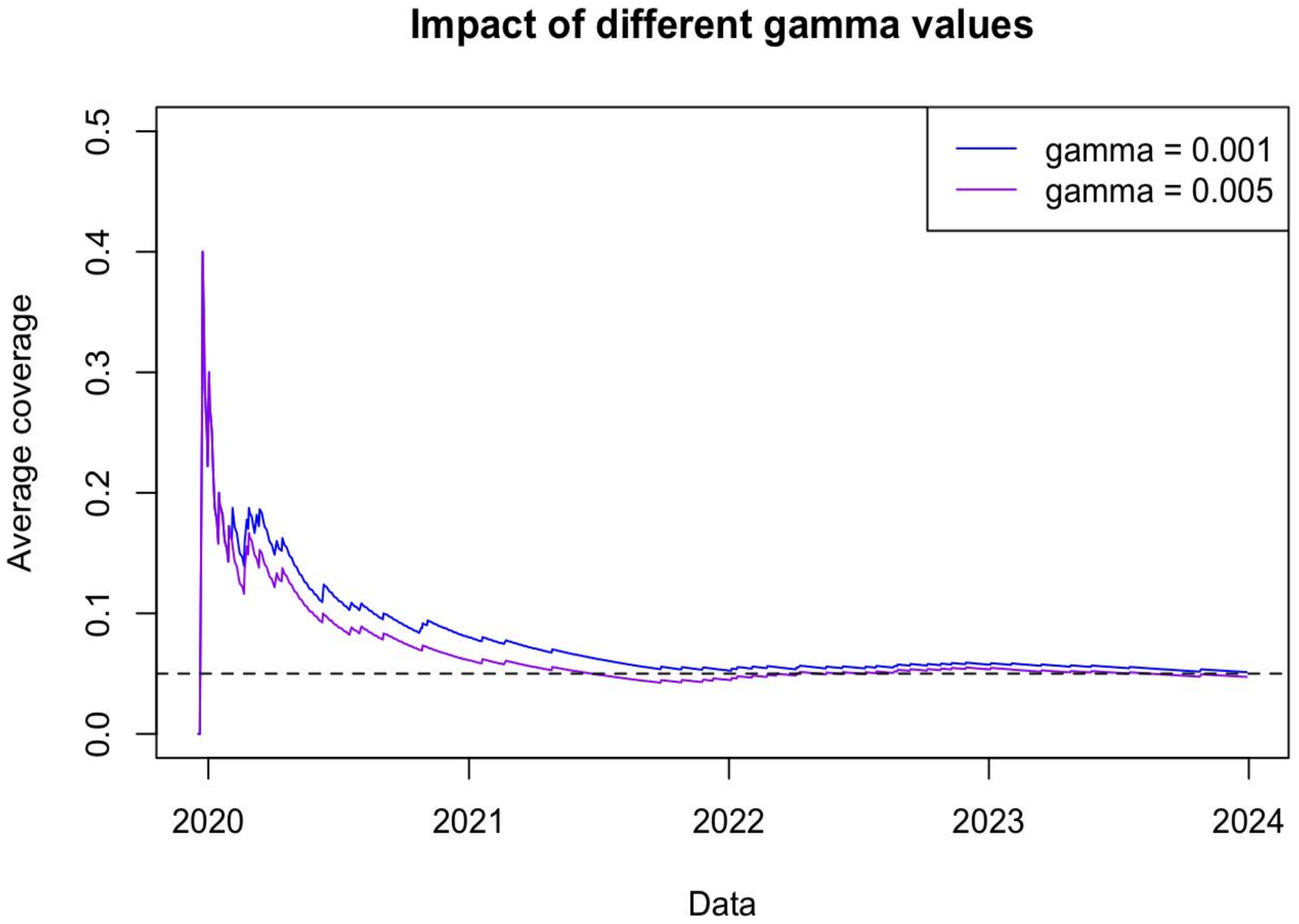
Evaluate the impact of the adaptation rate by choosing different values

- Effect of Increasing γ: When γ increases, the algorithm becomes more sensitive to changes in the data, which leads to better adaptation to shifts in the underlying distribution.
  The model adjusts more quickly to new information, improving the coverage of the forecast (i.e. how often the prediction intervals contain the actual values).

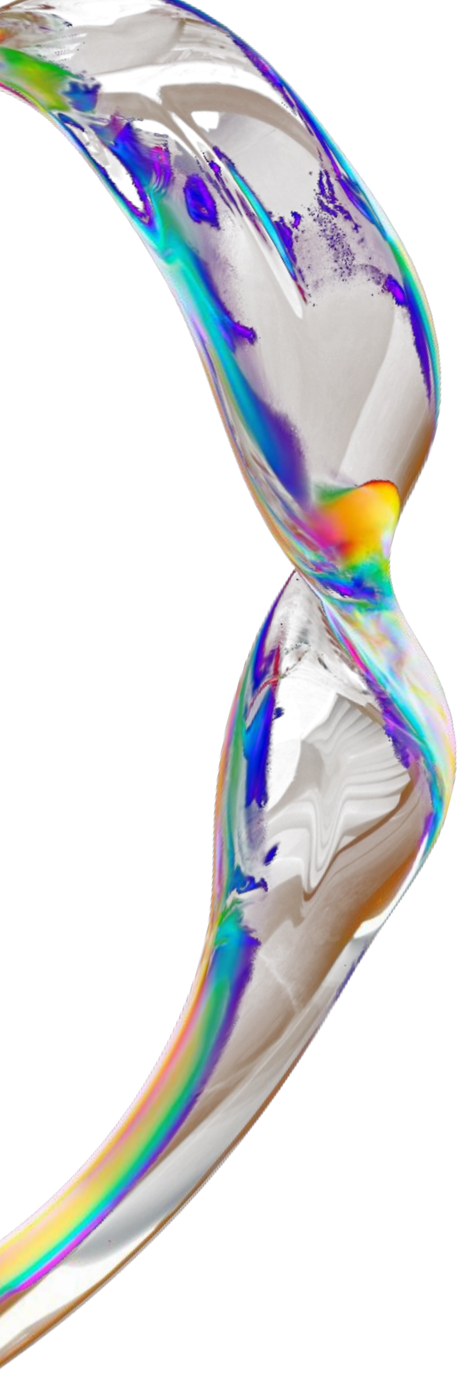
- Problem with Very Large γ Values: If γ is set to 0.01 or larger, the algorithm could stop working because such a value can lead to negative alpha (for example) and we would get an error from R trying to compute the quantiles.



Why this happen?



While increasing γ can make the method more responsive and improve the coverage, setting γ too high makes the algorithm unstable, causing the coverage rate to move outside the valid range. Therefore, there is a trade-off between sensitivity (larger γ) and stability (smaller γ).



To prevent the critical choice of γ an ideal solution is an adaptive strategy with a time dependent γ. The authors of the second paper we had to read propose two strategies based on running ACI for K ∈ N values $\{(\gamma_k)_{k\leq K}\}$ of γ, chosen by the user:

Naive strategy

AgACI strategy

The two strategies differ in the rule with which they select the K different γ at each time.

Naive strategy

The goal is to minimize predictive interval length while ensuring validity, achieved through a two-phase approach.

Warm-up Phase: Initially, the method operates arbitrarily during the first Tw steps. During this period, the value of γ is fixed (e.g., $\gamma = 0$), without optimizing it. This phase allows the system to gather enough information about the data and its behavior before trying to select an optimal γ.

Post-Warm-up: Exploiting the information gathered in the warm-up period, fot $t \geq T_0 + T\omega$ a γ^k is selected based on past performance.

Firstly we construct::

$$A_t = \left\{ k \in [1, K] \mid t^{-1} \sum_{s=1}^{t} \mathbb{I}_{y_s \in C_{\alpha_{s,k}}(x_s)} \geq 1 - \alpha \right\}$$

This is the set of the indices *k* whose correspondent ACI ensures a **valid** predictive interval.

If A_t is a non empty set, we select:

$$k^*_{t+1} \in \arg\min_{k \in A_t} \left(t^{-1} \sum_{s=1}^{t} \text{length}(C_{\alpha_{s,k}}(x_s)) \right)$$

Among the valid intervals, we select the more **efficient** (the one with the lower average length).

Otherwise:

$$k^*_{t+1} \in \arg\min_{k \in [1,K]} (|1 - \alpha - t^{-1} \sum_{s=1}^{t} \mathbb{I}_{y_s \in C_{\alpha_{s,k}}(x_s)}|)$$

If there are not the valid intervals, we select the one that has the **closest coverage rate to 1-α**

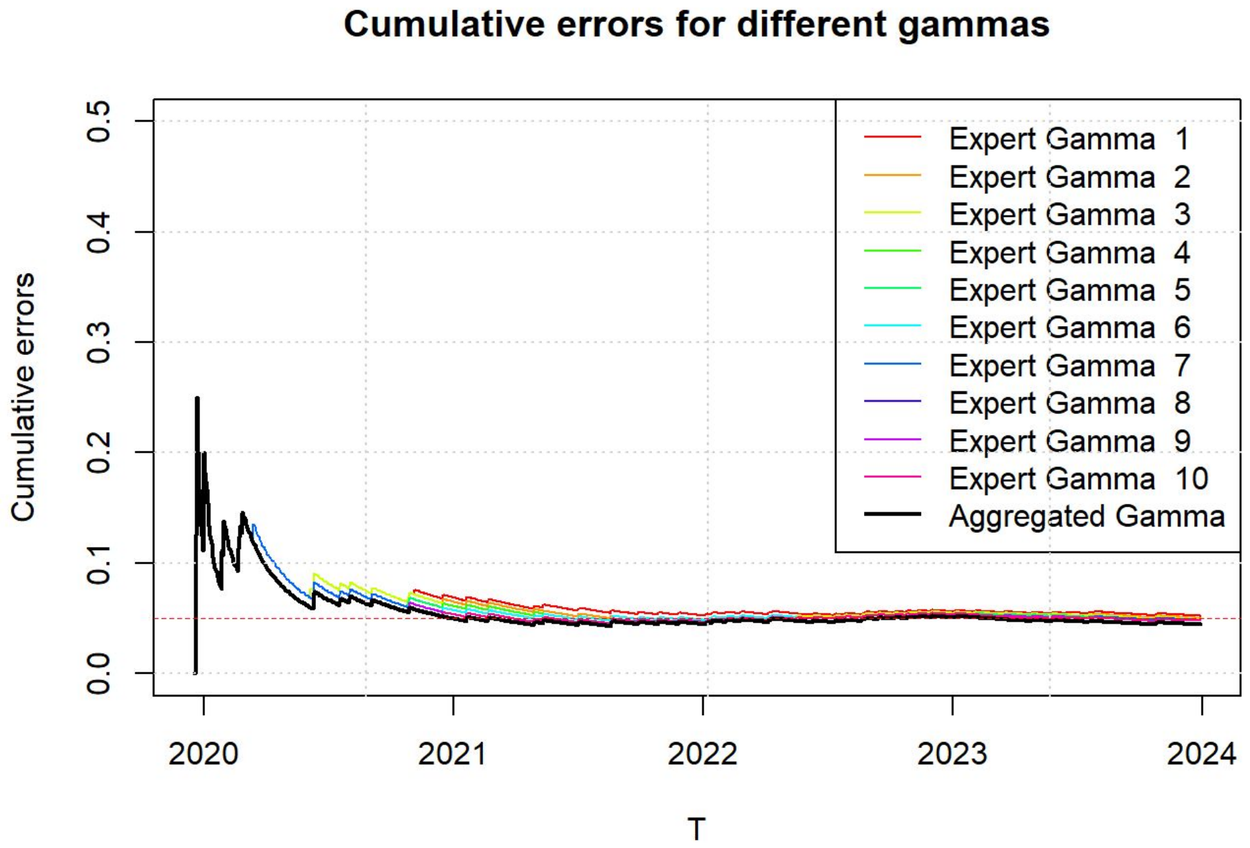
Naive strategy: results

The results for α = **0.005** shows that the naive strategy (aggregate gamma in figure) provides better coverage and prediction intervals than individual expert gammas.

The plot of average cumulative errors shows that the naive strategy consistently keeps the average cumulative error close to the target α = **0.05** (represented by the red horizontal line). The individual expert gammas, on the other hand, exhibit varying performance with some of them deviating further from the target error rate.

Moreover, the aggregate demonstrates a lower Median Interval Length compared to each expert. Although the Average Interval Length of the aggregate is not the absolute lowest, it still ranks among the smallest.

However, for α=0.001, the performance deteriorates in accordance with what is showed in the paper “Adaptive Conformal Predictions for Time Series”.



Average interval length for EACH gamma:
6.746173 6.901056 7.202726 7.239034 7.460935 7.578465 7.649217 7.606568 7.823446 7.872616

Average interval length for the Naive strategy: **7.194258**

Median interval length for EACH gamma:
6.873996 6.938913 6.976310 6.970595 7.006243 7.123205 6.968306 6.932987 6.962098 6.955754

Median interval length for the Naive strategy: **6.829778**

AgACI: our proposal

Instead of building the lower and upper bounds of the prediction interval for each expert and then building the “aggregate” bounds, we focus directly on the α□. Each expert produces an α□^{k} and what we want to do is create an “aggregate” α□ given by the weighted average of the α□^{k} . The weights of each expert are calculated with a BOA aggregation and with an MSE loss function. The weights will be higher for experts whose average miscoverage rate is close to the miscoverage parameter alpha for each time.

$$\ell(E_t^k) = (\alpha - E_t^k)^2 = \left(\alpha - \frac{\sum_{s=1}^{t} \mathrm{err}_s^k}{t}\right)^2 \longrightarrow$$ Experts’ losses at time t

$$\ell(A_t) = (\alpha - A_t)^2 = \left(\alpha - \frac{\sum_{k=1}^{m} w_t^k E_t^k}{\sum_{k=1}^{m} w_t^k}\right)^2 \longrightarrow$$ Aggregate’s loss at time t

The formula for the BOA aggregator is quite complex so we decided not to show it.
Intuitively, this procedure favours online learners which have predicted accurately and whose losses are close to those of the last aggregation, ensuring stability of the weights over time.

Once we get the weights for each time, we compute the aggregated estimate:

$$\tilde{\alpha}_{t+1} = \sum_{k=1}^{K} w_t^k \alpha_{t+1}^k$$

AgACI: our proposal

Instead of building the lower and upper bounds of the prediction interval for each expert and then building the “aggregate” bounds, we focus directly on the α□. Each expert produces an α□^{k} and what we want to do is create an “aggregate” α□ given by the weighted average of the α□^{k} . The weights of each expert are calculated with a BOA aggregation and with an MSE loss function. The weights will be higher for experts whose average miscoverage rate is close to the miscoverage parameter alpha for each time.

$$l_{k,t} = (\alpha - E_t^k)^2$$

Expert’s loss at time t

$$E_t^k = \frac{\sum_{s=1}^{T} err_t^k}{t}$$

Expert’s miscoverage rate

$$\ell(A_t) = (\alpha - A_t)^2 = \left(\alpha - \frac{\sum_{k=1}^{m} w_t^k E_t^k}{\sum_{k=1}^{m} w_t^k}\right)^2$$

$$\ell(E_t^k) = (\alpha - E_t^k)^2 = \left(\alpha - \frac{\sum_{s=1}^{t} \mathrm{err}_s^k}{t}\right)^2$$

The formula for the BOA aggregator is quite complex so we decided not to show it.
Intuitively, this procedure favours online learners which have predicted accurately and whose losses are close to those of the last aggregation, ensuring stability of the weights over time.

Once we get the weights for each time, we compute the aggregated estimate:

$$\tilde{\alpha}_{t+1} = \sum_{k=1}^{K} w_t^k \alpha_{t+1}^k$$

AgACI: results

Average interval lengths for each expert:
42.27132 29.23036 31.16750 26.43909 24.42164

Average interval length for the aggregate: **29.86819**

Median interval lengths for each expert:
29.26448 20.79938 22.22856 17.69180 14.40582

Median interval length for the aggregate: **21.49678**

Conclusions: The plot demonstrates that the aggregate achieves better coverage than any individual expert. While the Median and Average Interval Lengths of the aggregate are not the absolute shortest, they rank among the lowest. This highlights the effectiveness of the adopted strategy in combining the strengths of each expert, resulting in a prediction interval that outperforms those provided by any single expert.