



| FGV Crypto x FGV Quant |

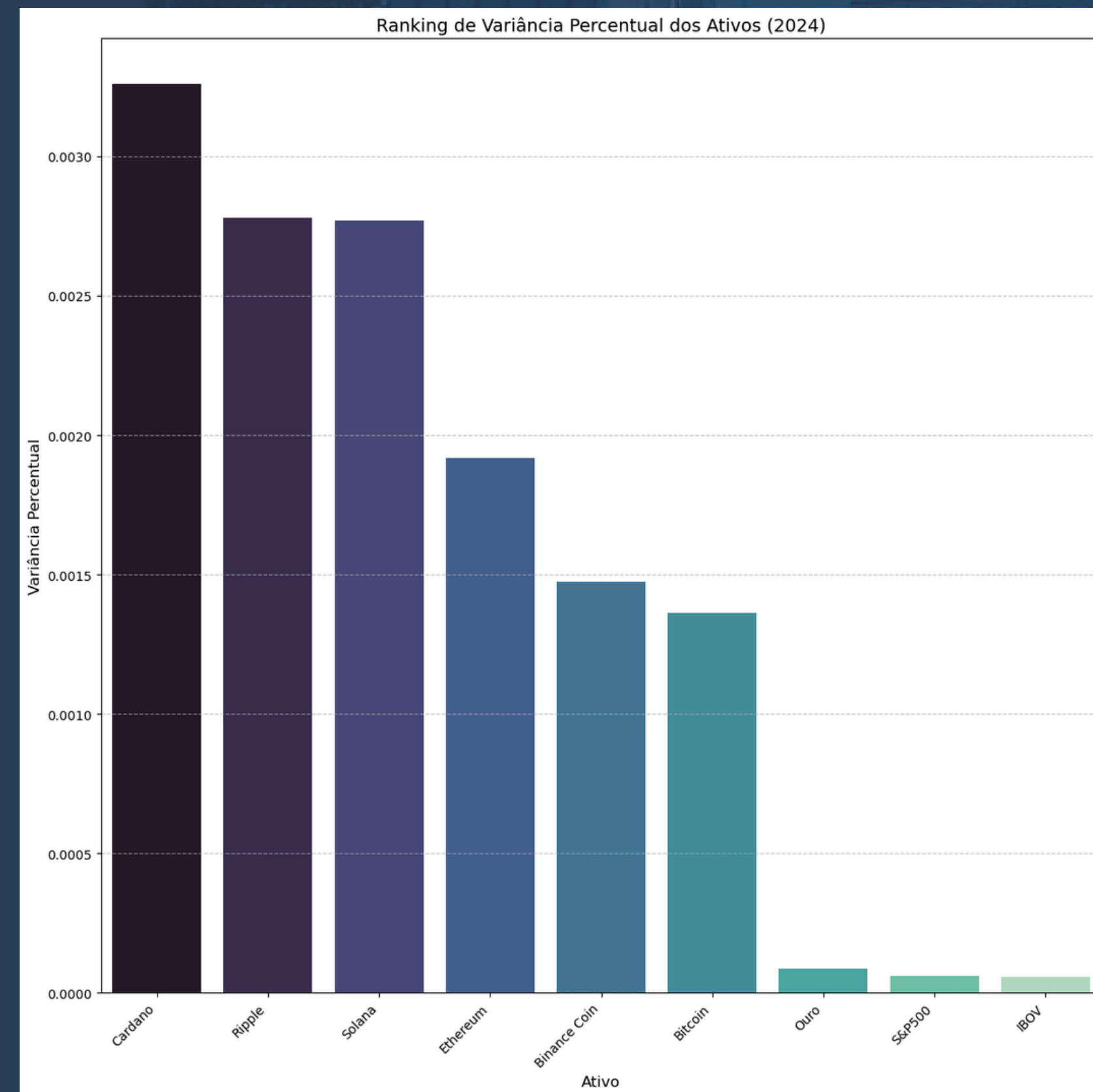


Previendo a Volatilidade de Criptoativos

André Togni | Bernardo Rubin | Francisco Zipman | Leonardo Ono

Introdução

- **O mercado de criptoativos é extremamente volátil:** o Bitcoin tem uma variância 22 vezes maior que a do S&P500, 24 vezes maior que a do Bovespa e 15 vezes maior que o ouro, aproximadamente. (Fonte: Yahoo Finance)
- **Conseguir prever essa volatilidade é crucial:** investidores têm um certo nível de risco o qual eles podem tolerar. Logo, prever a volatilidade é um bom começo para entender o risco subjacente aos criptoativos e pode trazer maior adesão ao mercado. Além do mais, volatilidade é um fator muito importante na precificação de derivativos, área relevante do mercado de criptoativos.
- **Objetivo:** visto a alta volatilidade e a importância de prevê-la, nos propomos a modelá-la.
- **Primeiro modelo:** Generalized Autoregressive Conditional Heteroskedasticity (GARCH).
- **Segundo modelo:** Long Short-Term memory (LSTM).
- **Terceiro modelo:** Random Forest (RF).
- **Quarto modelo:** AFTER.



Revisão Bibliográfica

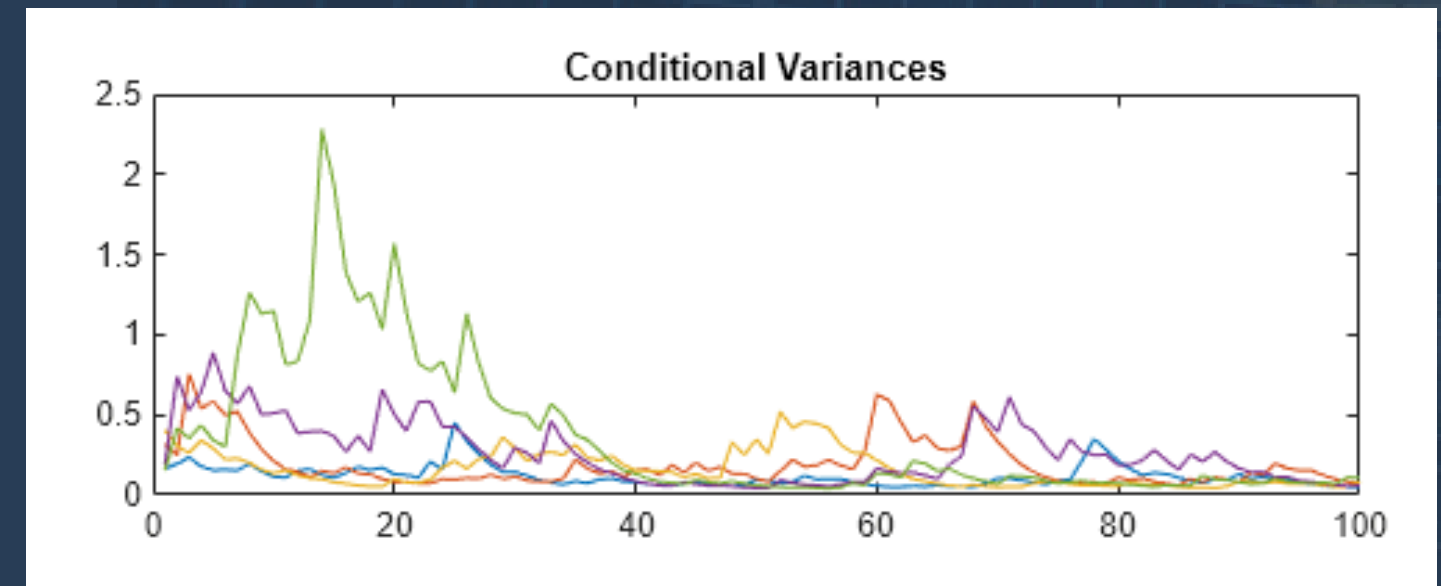
- A bibliografia do tema ainda é recente e não é muito extensa. Além disso, há pouco consenso entre os pesquisadores.
- Sabry, Farida, et al (2020) - GARCH é o mais utilizado para prever volatilidade, mas tem dificuldade quando se trata de criptoativos. Modelos que usam Machine Learning (ML) e a mistura de ML com GARCH e suas variantes parecem promissores (AFTER). Contudo, é necessário mais pesquisa sobre o tema.
- Dudek, Grzegorz, et al (2024) - Reforçam que modelos GARCH e suas variações são os mais utilizados pra prever volatilidade e que podem encontrar dificuldades em modelar criptoativos. Por isso, métodos mais complexos de ML são sugeridos.
- Wang, Andreeva e Martin-Barragan (2023) - Modelos de machine learning como RF e LSTM performam melhor que o GARCH em criptoativos. Esses modelos, apesar de muito usados em retorno de ações, são raramente usados para prever a volatilidade de criptoativos.
- Guo, Tian, Bifet, e Antulov-Fantulin (2018) - Constroem modelos GARCH e ML, concluem que a melhor opção para prever a volatilidade é uma mistura temporal dos dois.
- **Conclusão:** modelos GARCH são muito utilizados para prever a volatilidade, mas especificamente em criptoativos, eles talvez não sejam os mais adequados. O ideal, de acordo com as pesquisas, é a junção de GARCH com modelos de ML (modelos híbridos).

Metodologia

- Tendo em vista:
 - A importância da previsão da volatilidade de criptoativos;
 - A pouca pesquisa sobre a utilização de ML para esse fim (em especial de modelos LSTM e Random Forest);
 - O fato que esses métodos aparentam ser promissores.
- Decidimos explorar esses modelos para tentar entender qual é o mais adequado para modelar criptoativos e notar as falhas e acertos de cada um.
- Em especial, nos atentamos a mistura dos modelos que, conforme explicam Sabry, Farida, et al (2020) e Guo, Tian, Bifet, e Antulov-Fantulin (2018), parece ser uma boa opção para as criptoativos, apesar de serem pouco utilizados.

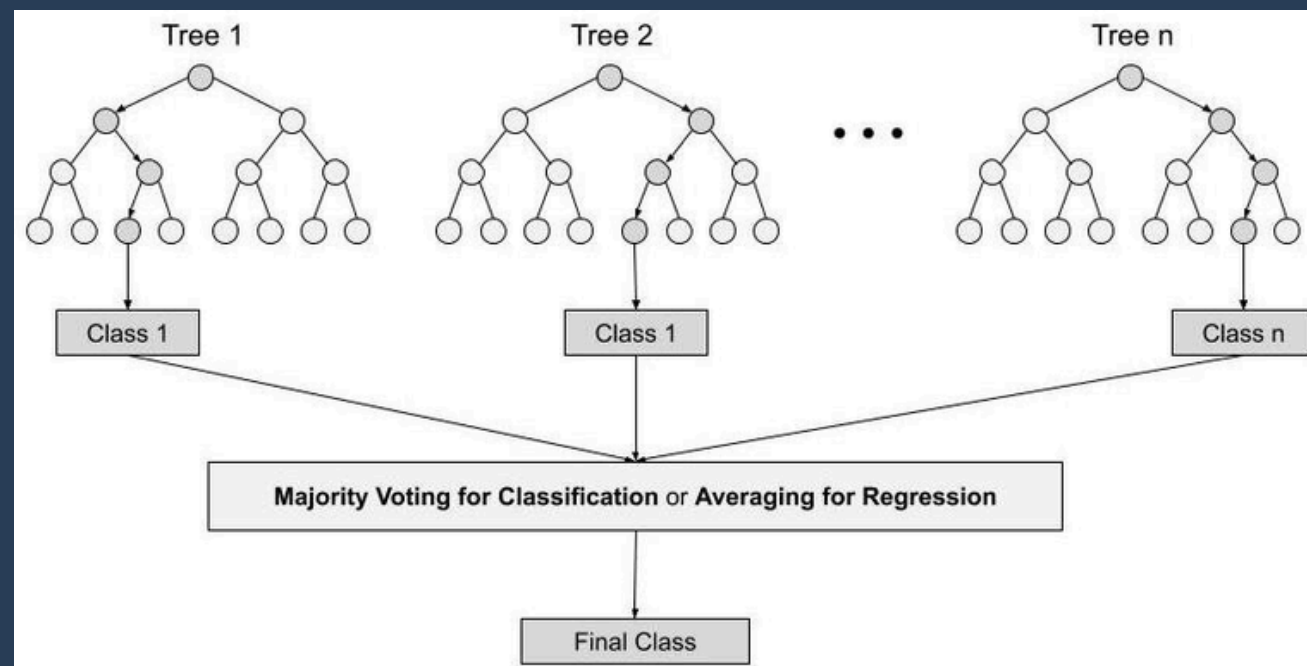
- Para montarmos modelos híbridos, utilizamos a metodologia AFTER, presente em Zou, Hui, e Yuhong Yang (2004).
- **Modelo GARCH:** O mais utilizado para previsão de volatilidade, especialmente no mercado financeiro como um todo. A volatilidade nesse modelo não é constante, já que depende dos erros de previsão de períodos anteriores e da variância de períodos anteriores. Um modelo GARCH (p,q) é dado por:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^q \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2$$



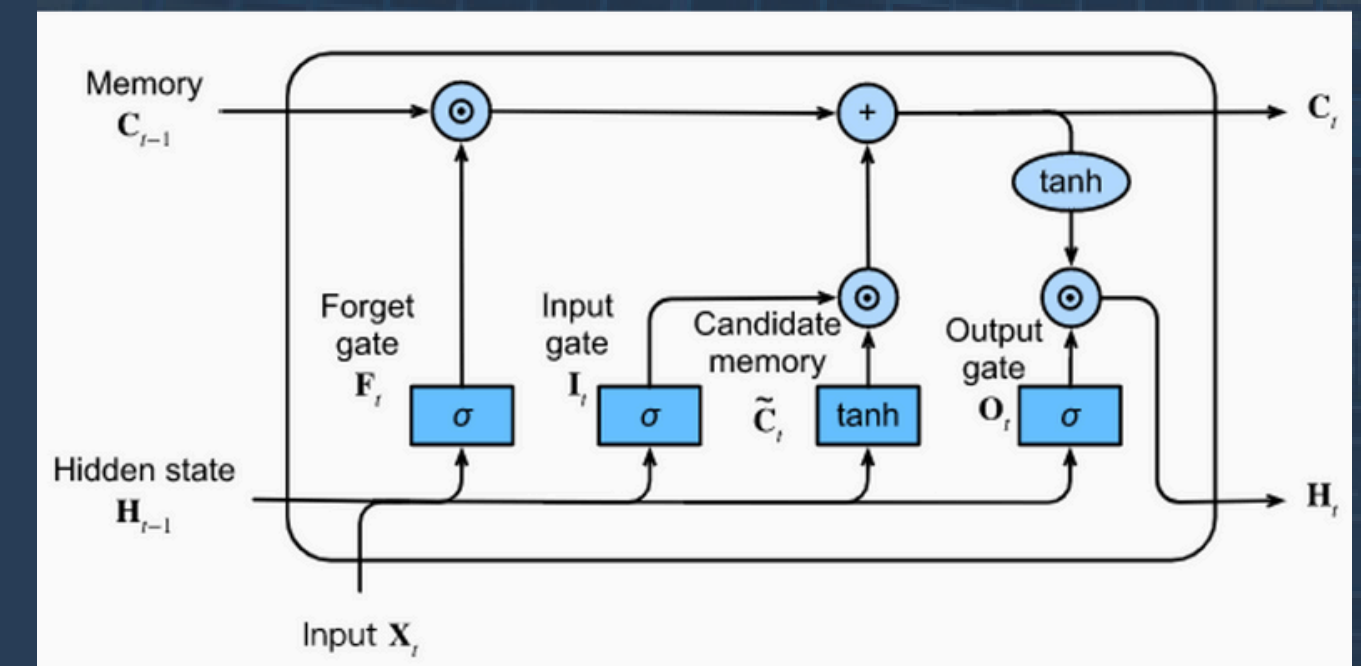
- **Random Forest:**

- É um modelo de árvores de decisão. Esses modelos dividem os dados em diversas “perguntas” binárias e observam padrões nos dados. Contudo, uma única árvore de decisão pode apresentar *overfitting* e instabilidades.
- Por isso, o RF combina diversas árvores de decisão e seus resultados.
- Cada árvore é treinada independentemente. Então, cada uma delas faz uma previsão do resultado. Pela independência, espera-se que os erros, na média, se cancelem e se obtenha um resultado mais preciso do que se utilizando apenas uma árvore.



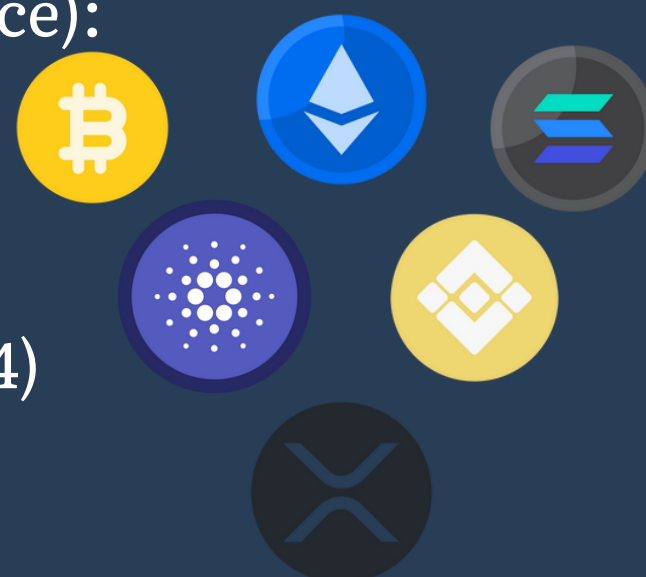
- **LSTM:**

- É um modelo popular de ML muito utilizado para séries de tempo. Mais especificamente, ele é uma recurrent neural network (RNN).
- Células de memória: São como caixas guardando a informação de períodos anteriores. Permite a rede ter uma memória de longo prazo e curto prazo e as utilizar no momento certo.
- Portas: São funções que decidem quais informações das células de memória o modelo deve lembrar, esquecer ou atualizar. São como um filtro que deixa passar apenas as informações relevantes.

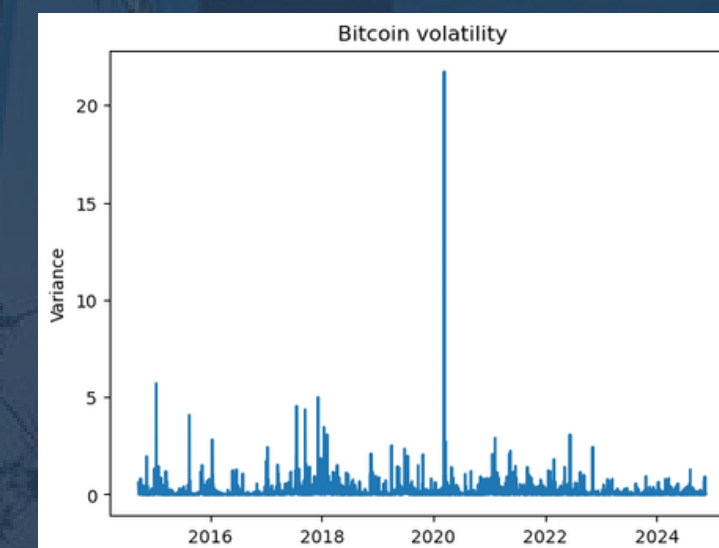
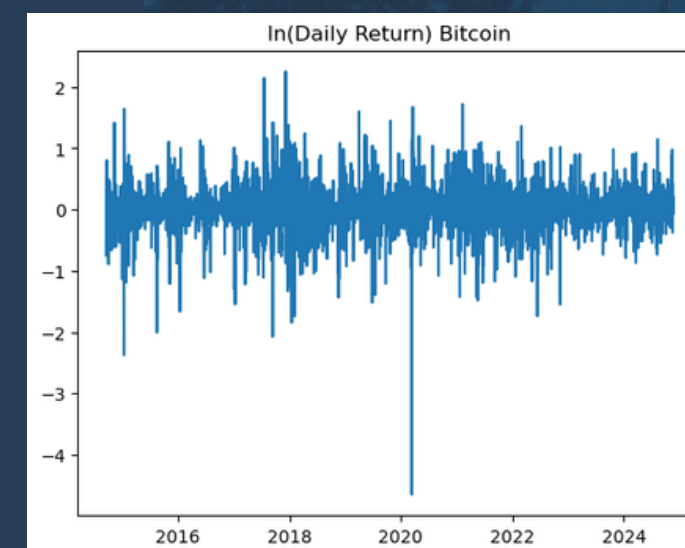


Dados

- **Escolha dos dados:** utilizamos dados do log retorno diário, pois são melhores para se trabalhar com percentuais de queda e aumento além de ser o mais usado pela bibliografia. Observamos os seguintes criptoativos (Fonte: Yahoo Finance):
 - Bitcoin (BTC) (2014-2024)
 - Ethereum (ETH) (2018-2024)
 - Solana (SOL) (2020-2024)
 - Binance Coin (BNB) (2018-2024)
 - Cardano (ADA) (2018-2024)
 - Ripple (XRP) (2018-2024)
- **Escolha dos criptoativos:** utilizamos moedas entre as 10 maiores em *market cap* (por serem mais relevantes para o mercado) e que são *layer 1*, para obtermos uma comparação padronizada.
- **Separamos a base de dados em duas partes:**
 - Base de treino: os primeiros 80% foram utilizados para treinar os modelos (in-sample).
 - Base de testes: os 20% finais para testar a precisão deles (out-of-sample).



- Gráficos do Ln(Retorno) e da Volatilidade do BTC



Estatísticas descritivas Ln(Retorno) do BTC

Observações	3716
Média	0.014284
Desvio padrão	0.365738
Mín	-4.647302
Máx	2.251190

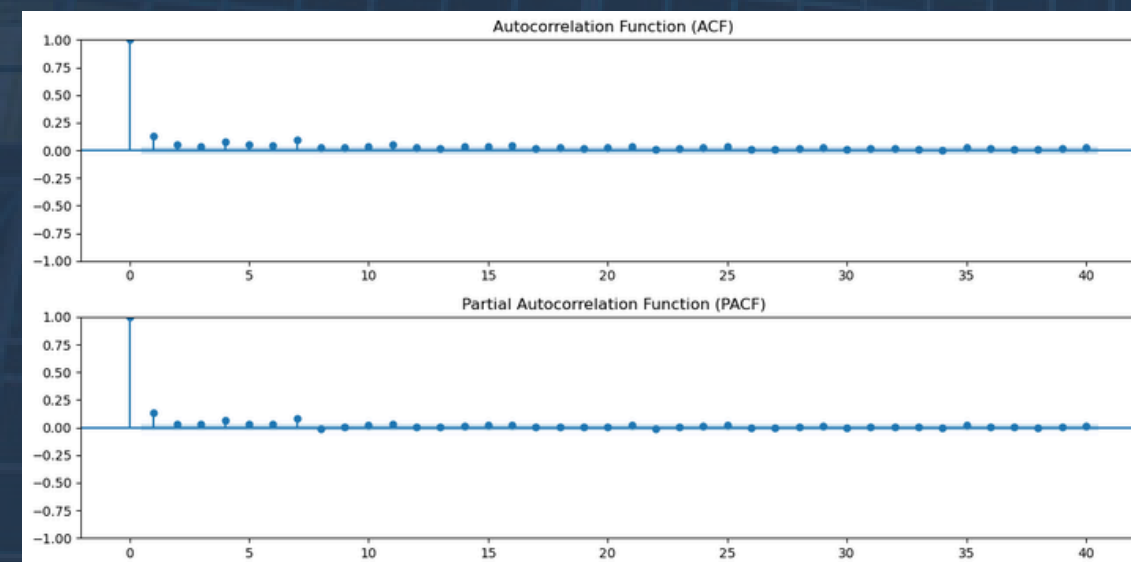
Modelos

- **GARCH:**

- Olhando para as séries de log retornos de todos os criptoativos analisados, vimos que elas são heteroscedásticas. As ACFs e PACFs mostram que essas séries não tem autocorrelação, por isso, as estimamos como: $y_t = \mu + \epsilon_t$
- Já que a série é composta apenas por um termo determinístico e um termo de erro, é fundamental estimar as características desse erro. Para isso, usamos um GARCH (p,q). As ACFs e PACFs das séries de volatilidade desses criptoativos indicam que o GARCH ideal está entre um (1,1) e (6,6).
- Estimamos todos os GARCHs desse conjunto para cada criptoativos usando a Base de Treinos.
- Usando o critério de informação de Akaike, selecionamos os modelos que performaram melhor na Base de Testes.
- Para todos os criptoativos, chegamos à conclusão que o melhor GARCH tem parâmetros de (p,q) iguais a (1,1).

- **Random Forest:**

- Ao rodar o modelo usamos alguns parâmetros:
- **Lags = 164.** Referem-se ao número de variáveis de atraso usadas como preditoras para prever o próximo valor. Foi escolhido 164 pois começamos com 100 e fomos adicionando múltiplos de 2 até que o modelo seguinte fosse pior que o modelo atual, de modo a achar um ponto ótimo;
- **N_estimators = 256.** Este parâmetro define o número de árvores de decisão na floresta. Foi escolhido 256 pois começamos com 1 fomos dobrando até que o modelo seguinte fosse pior que o atual, semelhante ao que foi feito nos lags.



- Funções de Auto-correlação total e parcial da Variância do Erro - BTC

- **LSTM:**
 - Ao rodar o modelo, definimos alguns parâmetros:
 - **Dense = 1.** Esse parâmetro refere-se a camada de saída, ou seja, no nosso caso a saída do modelo terá apenas um neurônio. Definimos desse jeito porque queremos prever um passo a frente;
 - **Lookback = 64.** Este se refere ao número de passos temporais usados como entrada para prever o próximo valor ou a saída desejada. Para escolhê-lo, fizemos o mesmo processo realizado no `N_estimators` do RF . Dessa maneira, chegamos no ótimo igual a 64;
 - **Epoch = 100.** Refere-se ao número de vezes que o modelo verá todo o conjunto de dados durante o treinamento. Quanto maior o número, melhor o treinamento do modelo, porém há risco de *overfitting*. Dessa maneira, seguindo a bibliografia, 100 é o número ótimo;
 - **Batch_size = 64.** Indica o número de amostras processadas antes que o modelo atualize seus pesos durante o treinamento. Foi feito o mesmo procedimento que o `lookback` para se chegar a 64.

- **AFTER:**
 - Esse modelo mistura o GARCH, RF e LSTM a partir da metodologia AFTER, que pondera os modelos com base no inverso de sua variabilidade.

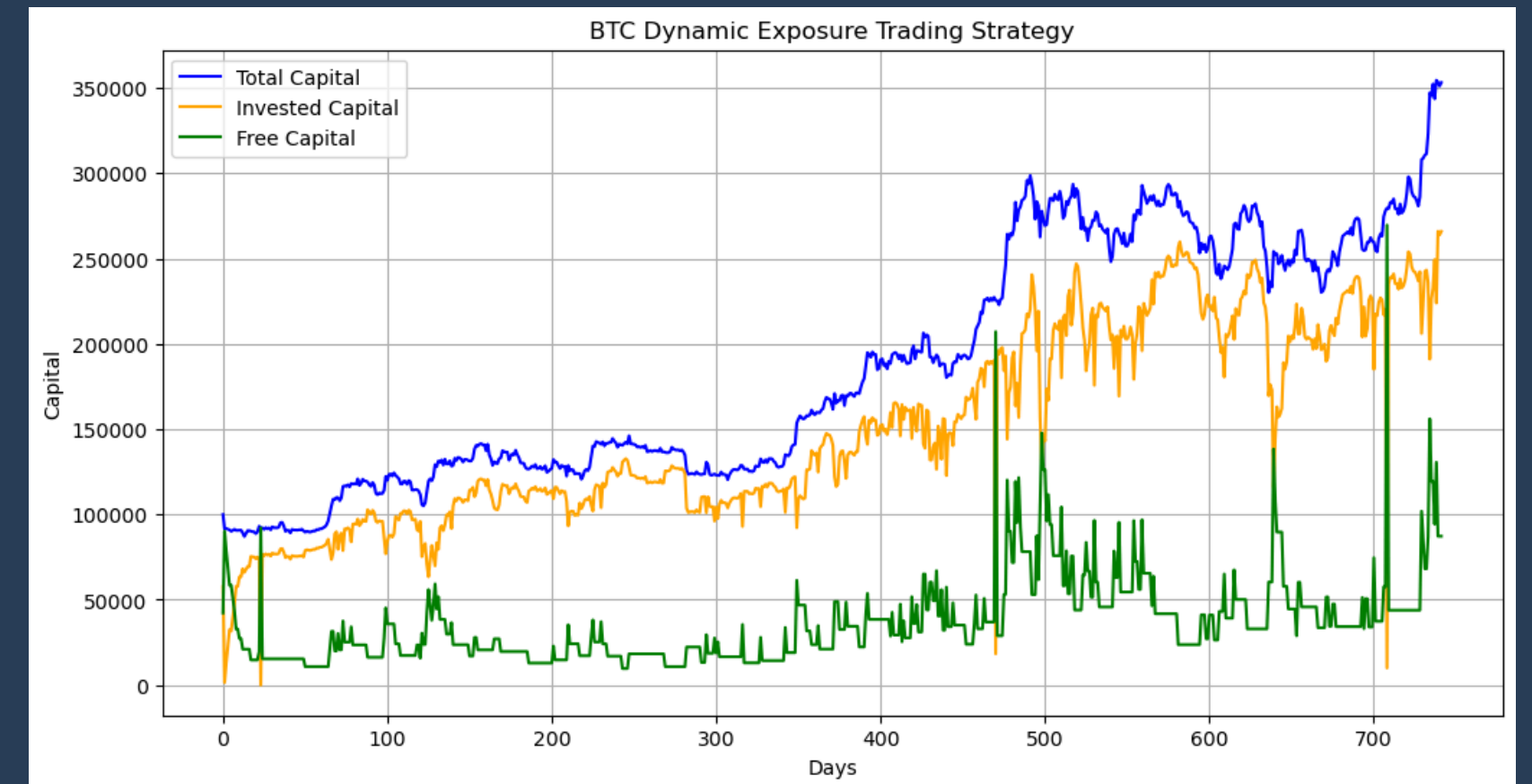
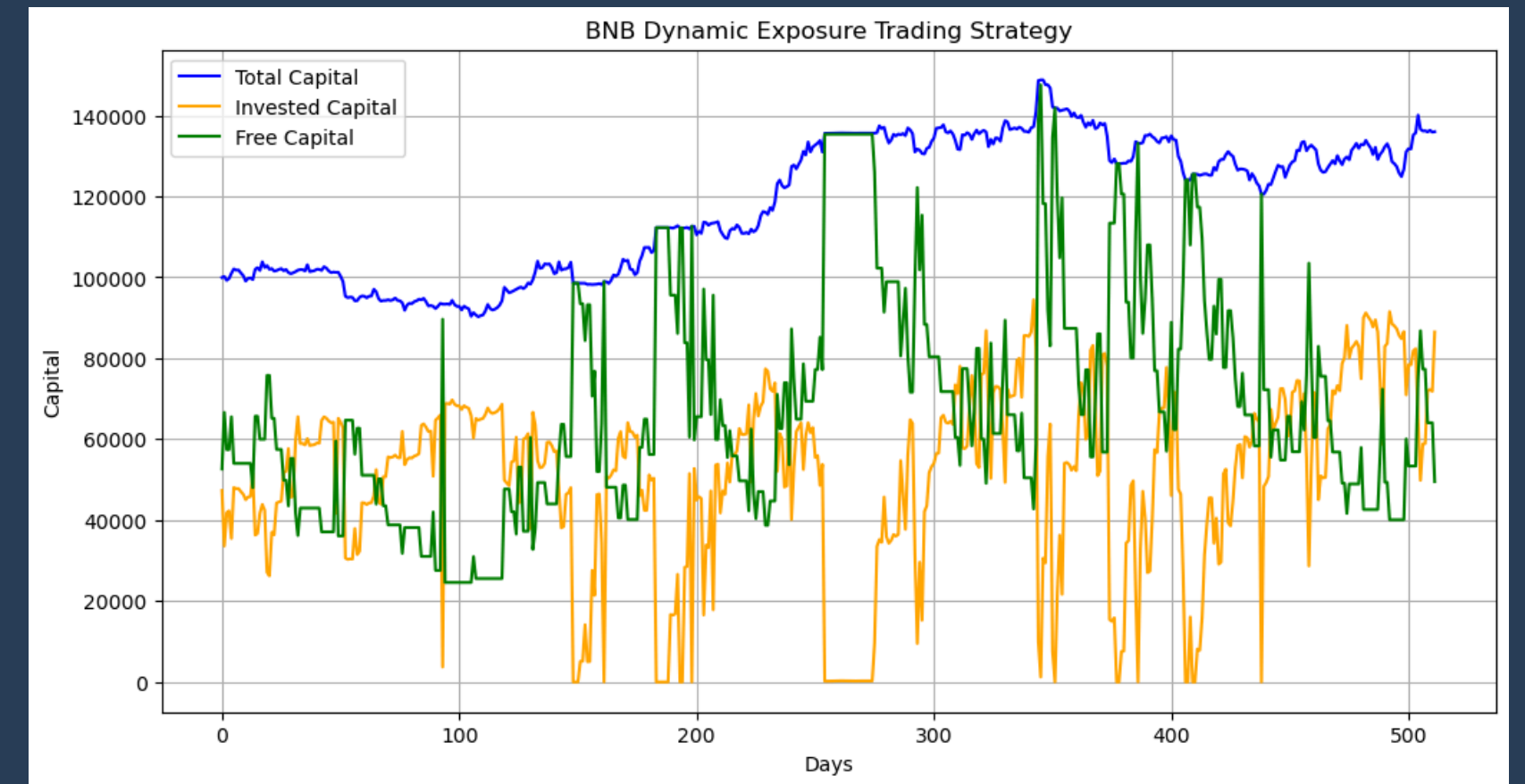
Resultados

- Para comparar os resultados dos modelos, utilizamos a medida de precisão root mean square error (RMSE).
- O melhor modelo para cada criptoativo foi o seguinte:
 - **BTC:** LSTM | RMSE = 0.1665
 - **ETH:** GARCH | RMSE = 0.2290
 - **SOL:** GARCH | RMSE = 0.3031
 - **XRP:** AFTER | RMSE = 1.4200
 - **ADA:** LSTM | RMSE = 0.3684
 - **BNB:** GARCH | RMSE = 0.1836
- Percebemos que GARCH performa consistentemente bem estimando os criptoativos.
- O modelo AFTER foi uma sólida opção, estando entre os 3 melhores para todos os ativos com exceção do BTC. Além disso, ele minimizou a variância dos resíduos de maneira consistente.

Modelo de Trading

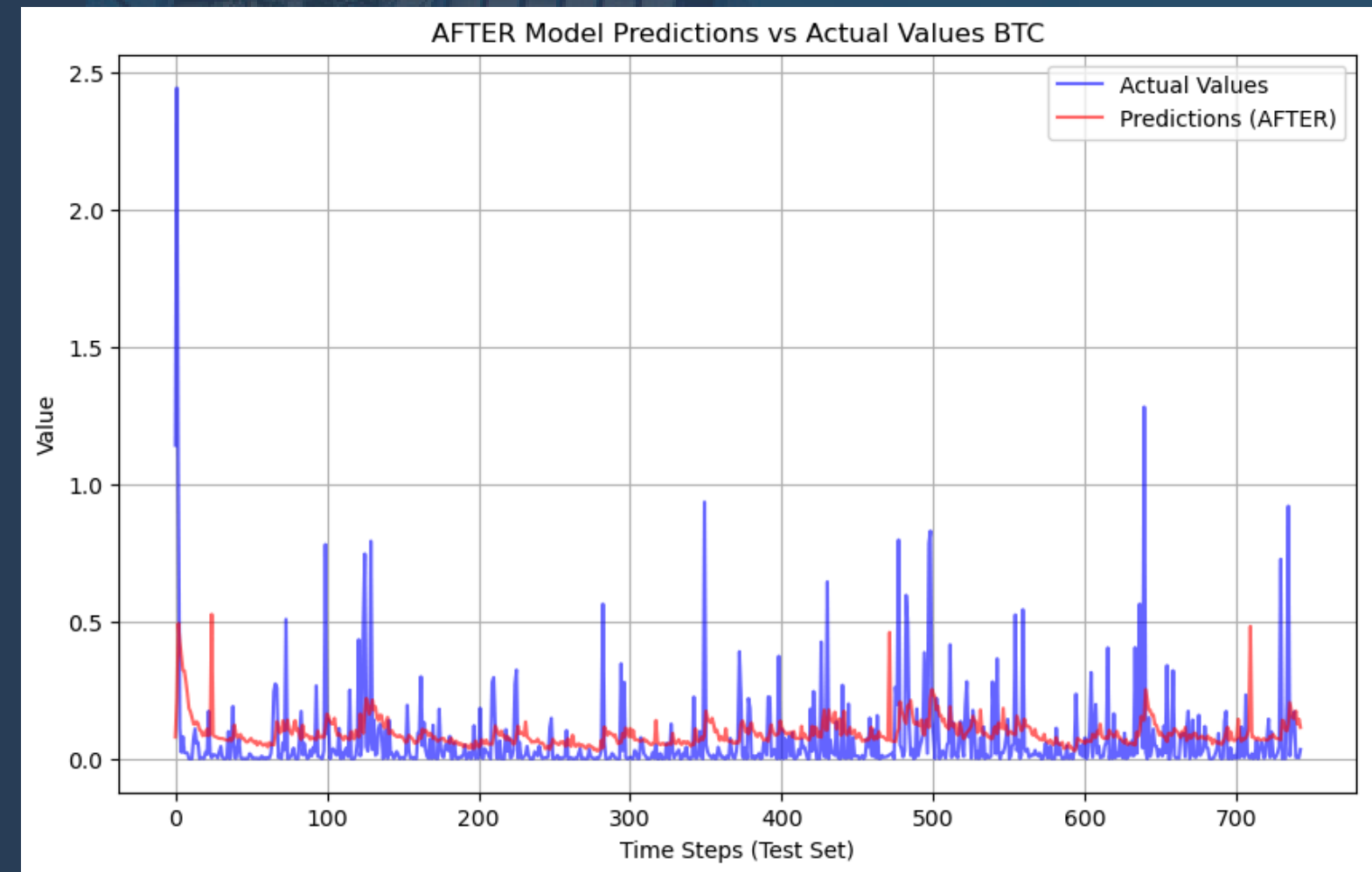
- Em cima do modelo AFTER, fizemos uma estratégia de trading diário para cada criptoativo.
 - A estratégia utilizada foi a **exposição dinâmica**, que se consiste em **aumentar ou diminuir sua exposição ao investimento dado o seu nível de volatilidade**. A ideia é que quanto menor a volatilidade do criptoativo, mais capital será alocado nele e vice-versa.
 - Para lidar com custos de transação, definimos um critério para movimentação com base em um delta entre a nossa **exposição desejada** e a nossa **exposição atual**. Ou seja, para que os custos de transação não corroessem o lucro da nossa estratégia, a nossa posição era modificada apenas se tivéssemos longe da nossa posição desejada, sendo essa distância baseada no delta.
 - Isso permite a **escolha do nível de risco a ser tomado**.
 - O **threshold** do modelo é o nível de volatilidade a partir do qual nós começamos a se expor ao ativo.

$$\text{exposure} = 1 - \frac{\text{volatilidade}}{\text{threshold}}$$



Conclusão

- Por meio da nossa estratégia de trading, podemos ver uma das possíveis aplicações da modelagem de volatilidade para criptoativos. A estratégia de exposição dinâmica permite que o drawdown máximo de um investimento seja limitado.
- Podemos ver através da tabela que a nossa estratégia performa muito bem para criptoativos, pois mesmo sem aplicar o dinheiro em caixa à uma taxa livre de risco, obtivemos um bom índice de sharpe e máximo drawdown.
- Podemos ver também que o GARCH teve uma performance um pouco melhor que outros modelos em algumas moedas. Achamos que isso pode estar ligado a uma falta de recursos computacionais para os modelos de machine learning.
- Isso também teria prejudicado o modelo AFTER (apesar de sua boa performance no geral), pois a combinação de modelos performa melhor quando os modelos são melhores.
- Com maior poder computacional ao dispor, acreditamos que o modelo AFTER performaria ainda melhor.
- Podemos falar que a combinação de modelos é uma metodologia que merece ser explorada mais afundo no caso da previsão de volatilidade de criptomoedas.



Sharpe da estratégia	1.494
Desvio da estratégia	0.020
Maximum drawdown	-0.229
Retorno total	253.169
Capital final	353169.167